

## Evaluarea performantei sistemelor informatice

Prof.dr. Ion IVAN, prep. Doru CAZAN, prep. Paul POCATILU  
Catedra de Informatica Economica, A.S.E. Bucuresti

*Evaluarea performantei sistemelor informatice este necesara compararii solutiilor de proiectare si realizare, folosindu-se diferite tehnici si instrumente. Trecerea de la o generatie la alta de software s-a concretizat prin obtinerea unor structuri de sisteme informatice cu grad de complexitate din ce în ce mai ridicat. Utilizarea tehnicilor, metodelor si instrumentelor moderne de proiectare si realizare a sistemelor informatice a condus la scurtarea duratelor de elaborare si, concomitent, la cresterea nivelului lor de calitate.*

**Cuvinte cheie:** sistem informatic, evaluare calitate, performanta, metrici SI.

### Factori de influenta

**F** Evaluarea performantelor sistemului informatic este un domeniu de cercetare deosebit de important. Calitatea a devenit o problema majora pentru managementul actual al firmei. În conditiile unui management strategic al calitatii, dezvoltat în toate etapele ciclului de viata a unui sistem informatic, se impune definirea unor modele de evaluare a performantei pentru a stabili masura în care raportul cost/calitate permite atât cresterea numarului de utilizatori cât si garantarea recuperarii investitiilor la producator si la beneficiarii sistemului informatic. În [McCa96], [HELL97], [Hals77] sunt fundamentate concepte de baza care permit realizarea unei analize coerente a tuturor aspectelor legate de evaluare.

Termenul *sistem* este de origine greaca si defineste un grup de entitati înrudite care functioneaza împreuna, formând un întreg. Si sistemele informatice sunt definite ca un set de componente intercorelate (module) care colecteaza, proceseaza, stocheaza si distribuie informatie pentru a sustine procesul de management [www01]. Conform [IEEE90], complexitatea este gradul în care un sistem sau o componenta a acestuia are o implementare dificil de înțeles si verificat. Dupa [EVAN87], complexitatea este gradul de intercorelare a unui sistem sau a unei componente a sistemului, determinata de factori ca numarul si complexitatea interfetelor, numarul si complexitatea structurilor alternative, gradul de imbricare

al structurilor de program, tipurile de date si structurile de date utilizate.

Factorii de influenta sunt diferiti si au intensitati variabile în timp pentru fiecare din etapele ciclului de viata. Ciclul de viata a sistemului informatic este alcatuit conform [www02] din urmatoarele etape:

- Investigatii preliminare
- Analiza sistemului
- Proiectarea sistemului
- Dezvoltarea
- Implementarea
- Mentenanta sistemului

În continuare sunt prezentate o serie de modele de evaluare a performantei sistemelor informatice în care se includ ca factori de influenta: numarul si complexitatea modulelor; interdependente programe/module; numarul fisierelor cu date si gradul de interconectare (referiri încrucisate) a câmpurilor (operanzi); complexitatea bazei de date; numarul punctelor de interactiune incluse în interfetele utilizatorului; numarul parametrilor care determina activarile de module.

Ierarhizarea acestor factori si formele analitice ale intensitatii interactiunilor, conduc la obtinerea diferitelor clase de modele de evaluare.

### Model de complexitate a sistemului informatic McCabe

În [www03], [McCa96] este prezentat modelul McCabe pentru evaluarea complexitatii programelor. În ipoteza omogenitatii perfecte a instructiunilor se construiesc grafuri asociate secventelor de program

pentru care se masoara complexitatea, fiecare instructiune  $I_1, I_2, \dots, I_n$  fiind reprezentata de un nod, ordinea de executie a acestora fiind evidentiata cu ajutorul arcelor. De exemplu, pentru secventa de program S1:

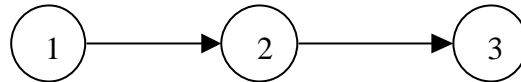


Fig. 1. Graful asociat secventei de program S1

Complexitatea McCabe este data de relatia  $C = m - n + 2$ , unde  $m$  este numarul arcelor din graf, iar  $n$  este numarul nodurilor grafului.

În secventa de program S1, complexitatea McCabe are valoarea  $C = 2 - 3 + 2 = 1$ .

Pentru o secventa S2:

$A = B + 1;$  //I1

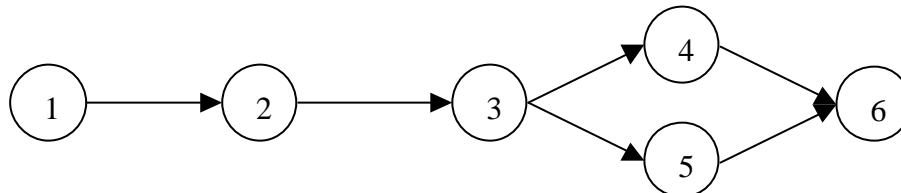


Fig. 2. Graful asociat secventei de program S2

Complexitatea McCabe a acestuia are valoarea  $C = 6 - 6 + 2 = 2$ . Se observa ca valoarea complexitatii McCabe depinde de numarul de structuri alternative si de numarul de cicluri cuprinse în secventa analizata. În [McCa96] se arata ca valoarea complexitatii McCabe este egala cu numarul maxim de drumuri liniare independente din graf.

Studiile au aratat ca exista o proportionalitate directa între complexitatea McCabe si frecventa erorilor [www04]. O valoare mai mica de 10 a complexitatii McCabe reflecta structuri de program de complexitate scazuta, usor de înțeles si de modificat, riscul de aparitie a erorilor fiind redus, în timp ce valori ale complexitatii McCabe mai mari de 50 caracterizeaza secvente de program greu de înțeles, având un risc foarte mare de aparitie a erorilor.

Complexitatea McCabe este extinsa în [MCCA89] pentru a permite masurarea complexitatii sistemelor informatice.

Valoarea complexitatii McCabe este calculata în faze avansate ale procesului de dez-

$A = B + 1;$  //I1

$B = 2 - C;$  //I2

$D = A + B;$  //I3

se asociaza graful din figura 1.

$B = 2 - C;$  //I2

if ( $A > B$ ) //I3

$D = A - B;$  //I4

else

$D = A + B;$  //I5

$D = D - C;$  //I6

graful asociat este dat în figura 2.

voltare de software, neputând fi un instrument eficient în activitatea de management a proiectelor. E folosit ca instrument de predictie a costurilor si eforturilor viitoare consumate în procesul de mentenanta.

### Modelul de performanta COCOMO

COCOMO (COConstructive COst MOdel) este un model care permite managerilor sa estimeze costurile, efortul si timpul necesare pentru un proiect de dezvoltare de software. Modelul este prezentat în [BOEH81] pe trei niveluri (de baza, intermediar si detaliat), fiecare având câte trei moduri de aplicare (organic, semi-detasat si inclus):

- Modul organic este utilizat în cazul proiectelor relativ mici, cu caracteristici foarte bine cunoscute, echipa având experienta în lucrul cu proiecte de acelasi tip;
- Modul semi-detasat reprezinta un stadiu intermediar între modul organic si cel inclus, fiind utilizat în cazul proiectelor mai putin familiare fata de cele pentru care se aplica modul organic;

- Modul inclus este utilizat pentru proiectele ale caror caracteristici nu sunt definite riguros, problemele de rezolvat fiind deosebit de dificile.

Complexitatea COCOMO este data de relatia  $E = aS^b m$ , unde:  $E$  este efortul calculat;  $s$  este valoarea metricii LOC (Lines Of Code);  $m$  este un multiplicator determinat pe baza atributelor sistemului care influenteaza costul final;  $a$  si  $b$  sunt constante determinate de modul si nivelurile modelului.

Punctul slab al modelului consta în faptul ca multiplicatorii sunt calculati pornind de la date empirice. Pentru modul inclus, aplicat problemelor dificile, datele pe baza carora se calculeaza multiplicatorii nu au întotdeauna consistenta necesara. De asemenea, în procesul de dezvoltare a sistemelor informatice, instrumentele, tehnicile si natura sistemului evolueaza, datele folosite pentru determinarea multiplicatorilor pierzând din veridicitate. Datele trebuie validate si reevaluate continuu actualizând multiplicatorii implicati în calcule.

### Modelul de performanta Halstead

Complexitatea de tip Halstead a fost dezvoltata pentru a fi masurata pornind de la textul sursa al modulului program, punând accentul pe complexitatea calculelor. Se determina cinci indicatori de complexitate pe baza numarul de operatori si operanzi:  $n_1$  - numarul operatorilor distincti;  $n_2$  - numarul operanzilor distincti;  $N_1$  - numarul total al operatorilor;  $N_2$  - numarul total al operanzilor:

- Lungimea programului:  $N = N_1 + N_2$
- Vocabularul programului:  $n = n_1 + n_2$
- Volumul:  $V = N \times \ln(n)$
- Dificultatea programului:  $D = (n_1/2) \times (N_2/n_2)$
- Efortul:  $E = D \times V$

Indicatorii sunt usor de calculat odata ce sunt definite reguli pentru identificarea operatorilor si operanzilor. Indicatorii se calculeaza pentru sistemele operationale, masoara eforturile de dezvoltare dupa ce codul a fost scris si se folosesc ca indicatori de complexitate ai procesului de

mentenanta. Daca se urmareste evolutia acestor indicatori pe parcursul procesului de dezvoltare se determina pe baza trendului un risc potential de crestere a complexitatii. Criticile aduse modelului de masurare a complexitatii afirma ca el masoara mai curând complexitatea lexicala a modulelor program, decât complexitatea structurala a acestora. Folosind, însa, un model complementar de masurare a performantelor (de exemplu, modelul McCabe), se poate construi un sistem informatic puternic pentru evaluarea complexitatii modulelor de program.

### Modele statistice pentru evaluarea performantei sistemului informatic

Comportamentul sistemelor informatice presupune înregistrarea în timp a consumurilor de resurse. Se definesc proceduri privind modul de culegere si stocare a datelor, se stabilesc algoritmi de prelucrare, precum si criteriile dupa care se interpreteaza rezultatele obtinute.

În cazul în care se lucreaza cu baze de date de dimensiuni mari, prezinta interes stabilirea duratei medii a unei tranzactii. Este important sa se stabileasca ce este o tranzactie, momentul care delimiteaza începutul tranzactiei, respectiv momentul de terminare a acesteia.

Prin volum de tranzactii se înțelege, de regula, numarul tranzactiilor. Tranzactia este o operatie de prelucrare care presupune specificarea unei chei de regasire, cautarea în baza de date a articolului si/sau a articolelor de legatura cu aceasta cheie, prelucrarea de date si afisarea unui mesaj sau a unui rezultat.

Se construiesc serii de date ce contin coduri, specificatii ale bazelor de date si duratele tranzactiilor. Se calculeaza durate medii si dispersii si se procedeaza la analiza statistica a stabilitatii proceselor de cautare si a omogenitatii comportamentului întregului sistem informatic cu toate operatiile care se produc asupra bazelor de date. Se construiesc perechi de forma:  $(c_i, d_i)$ ,  $(c_i, e_i)$ ,  $(c_i, f_i)$ ,  $(c_i, g_i)$ , unde:  $c_i$  este codul tranzactiei  $i$  sau a operatiei  $i$ ;  $d_i$  este durata

tranzactiei  $i$ ;  $e_i$  este durata operatiei de modificare a unui câmp;  $f_i$  este durata de stergere unui articol;  $g_i$  este durata de adaugare a unui articol.

În cursul unei perioade date se înregistrează  $N$  operatii ( $i=1, 2, \dots, N$ ). Omogenitatea prelucrarilor vizează mai multe perioade, durate medii ale operatiilor și efectuarea de teste statistice pentru verificarea egalității de medii și/sau dispersii.

### Concluzii

Performanța sistemelor informatice este dată de structura și este evaluată prin modele de tip Halstead, McCabe. În cazul în care proiectul are o complexitate foarte ridicată ce impune planificare și gestiune de resurse importante se utilizează modelul COCOMO.

Performanța este rezultatul unei activități manageriale eficiente în toate etapele ciclului de dezvoltare. Problema evaluării performanței sistemelor informatice revine la a defini un sistem de indicatori care măsoară compoziția structurală a sistemelor, comportamentul în exploatare a acestora și corelează consumurile în expresie valorică cu efectele pe care le determină la agenții economici beneficiari, respectiv costurile și profitul.

Este importantă proiectarea de software care realizează înregistrarea automată a comportamentului sistemului informatic și declanșează periodic evaluarea indicatorilor statistici ai performanței. De asemenea, produsul software efectuează analiză de structură a programelor și bazelor de date ce alcătuiesc sistemul informatic pentru a cuantifica interacțiunile operanți – operatori și pentru a evalua gradul de redundanță încorporat în sistemul informatic.

Evaluarea calității sistemelor informatice și-a pus bazele de la începutul anilor 1990 [DALH97]. Au fost dezvoltate modalități de abordare, dar nici una nu propune soluții general acceptate și adecvate atât mediului științific, cât și scopurilor practice. Cele mai multe abordări sunt orientate pe aspecte tehnice și de control ale calității,

rezultatele obținute fiind nesatisfăcătoare. Se impune în acest context dezvoltarea unui sistem de indicatori și a unui nou model de evaluare a calității sistemelor informatice.

### Bibliografie

[DALH97] Dahlberg Tomi, Jarvinen Janne - Challenges to IS quality, Information and Software Technology, 39 no. 12/1997.

[EVAN87] Evans, Michael W. & Marciniak, John - Software Quality Assurance and Management, New York, NY: John Wiley & Sons, Inc., 1987.

[HALS77] Halstead, Maurice - H. Elements of Software Science, Operating, and Programming Systems Series, Volume 7, New York, NY: Elsevier, 1977.

[HELL97] L.A. von Hellens - Information systems quality versus software quality. A discussion from a managerial, an organisational and an engineering viewpoint, Information and Software Technology, 39 no. 12/1997.

[IEEE90] Institute of Electrical and Electronics Engineers - IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, N.Y. 1990.

[MCCA89] McCabe, Thomas J. & Butler, Charles W. - Design Complexity Measurement and Testing, Communications of the ACM 32, 12 December 1989.

[MCCA96] Arthur H. Watson, Thomas J. McCabe - Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric, Computer Systems Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-0001, September 1996

[www01]<http://www.gsia.cmu.edu/bb26/70-451/95/lectures/lecture2/lecture2.html>

[www02][http://www.cs.adfa.oz.au/teaching/studinfo/cs1/Lecture\\_notes/programming/lecture09/index.html#feasibility](http://www.cs.adfa.oz.au/teaching/studinfo/cs1/Lecture_notes/programming/lecture09/index.html#feasibility)

[www03]<http://www.cstp.umkc.edu/personal/jfairfax/homework6.html>

[www04]<http://www.sei.cmu.edu/str/descriptions/cyclomatic.html>