

POSSIBILITATI DE EVALUARE A CALITATII PRODUSELOR SOFTWARE

Crețu Emil¹

Gramada Argentina Dragu¹

Abstract . Quality represents the best insurance policy to loyalize customers and to gain new business, the most effective defence against competition and the only way to develop and stabilize revenues. In the first part, the article refers to the importance of quality assessment of all IT products, starting with the necessity to measure the quality of these products, the preoccupations and the particularities of the domain. In the second part we present traditional models in software product quality, particularities of the current models and the metrics, with their characteristics and role in software products quality assessment. The measure is to know.

1 Importanța evaluării calității produselor din domeniul tehnologiei informației.

Un produs IT este, în general, o combinație de hardware, sisteme software, aplicații software (care pot fi special comandate sau sub forma de pachete de programe), software și echipamente de comunicații, livrări de servicii de instruire și de susținere, în conformitate cu politici și proceduri, documentate, corespunzătoare, precum și materiale consumabile. Dintre componentele unui produs IT o atenție deosebită trebuie acordată software-ului, acesta fiind, conform standardului ISO 9000-3 “o creație intelectuală care cuprinde programe, proceduri, reguli și orice documentație asociată referitoare la funcționarea unui sistem de prelucrare a datelor”².

Produsul software reprezintă ansamblul complet format din programe, proceduri și documentație asociată, pentru calculator precum și datele destinate să fie livrate unui utilizator. Deci un produs software include și documentația de instalare și operare, documentația de utilizare și întreținere, proceduri de utilizare și materiale de instruire.

1.1 Necesitatea măsurării calității produselor din domeniul tehnologiei informației

Realizarea de produse software pentru aplicații complexe presupune repartizarea unor programatori cu experiență și utilizarea unor sisteme de calcul performante. În acest scop, trebuie să se specifice câți programatori sunt necesari și cât timp vor lucra. De asemenea, trebuie specificate câte posturi de lucru se vor aloca, ce resurse sunt necesare (compilatoare, periferice, memorie) și pe ce durată. Dezvoltarea de aplicații complexe presupune analiză, proiectare, testare, implementare, toate necesitând forță de muncă și resurse hardware. *Devizul de cheltuieli* care precede elaborarea unui produs software reflectă valoric consumurile de resurse necesare realizării produsului, iar stabilirea costurilor presupune *efectuarea de măsurări*. Cumpărătorul (utilizatorul) alege după diferite criterii un anumit produs. Dominantă este *corelația cost-calitate*. Atât costul, cât și calitatea, trebuie exprimate cantitativ, pentru a evidenția mărimea avantajului pe care o determină alegerea făcută. Efectele noilor tehnologii sunt cele care determină aplicarea lor numai în măsura în care, prin **cuantificare**, conduc la rezultate superioare.³ De exemplu, *programarea structurată*, prin sistematizările determinate, a condus la reducerea complexității, la creșterea productivității și facilitarea mentenanței software-ului. Toate acestea au fost evidentiate cantitativ, prin indicatori. *Programarea orientată* spre creșterea gradului de reutilizabilitate determină economie de muncă, chiar dacă nivelul de complexitate a problemelor ce se rezolvă este mult mai mare. Aceste tehnologii se implementează cu ajutorul unor produse software dedicate. Studiul profitabilității, necesar achiziționării acestor instrumente, trebuie să răspundă la întrebarea în cât timp se recuperează investițiile de achiziționare. Acest răspuns este posibil numai prin efectuarea unor cuantificări ale avantajelor generate de programarea structurată. *Analiza comparată* prin prezentarea în paralel a unor programe destinate rezolvării aceleiași probleme, scoate în evidență ceea ce au ele în comun, precum și ceea ce le diferențiază. Trăsăturile importante care lipsesc vor determina perfecționări ulterioare. Noul produs trebuie caracterizat în mod riguros, atât prin costurile implicate cât și prin nivelul caracteristicilor de calitate obținute. Fără măsurare, orice încercare de a stabili modele care să descrie legăturile structurale între valorile tehnice și fizice, nu se

¹ Titu Maiorescu University, Computer Science Department and Economical Sciences Department

² Centrul Internațional de Comerț UNCTAD/GATT, Organizația Internațională de Standardizare ISO-MANULUL CALITATII-Ghid pt implementarea standardelor internaționale ISO-editura Tehnica Bucuresti 1998

³ *Cuantificarea* constituie operația de introducere a conceptelor cantitative, de trecere în studiul unui fenomen de la aprecieri atributive la concepte cantitative.

poate realiza. Referindu-se la măsurare Lord Kelvin făcea următoarea afirmație: "când poți să măsoari lucrul despre care vorbești și poți să-l exprimi în numere, știi ceva despre acel lucru, dar când nu poți să-l măsoari și să-l exprimi în numere atunci cunoștințele despre el sunt nesatisfăcătoare".⁴ În domeniul IT, există dificultăți fundamentale în măsurarea produselor și mai ales a produselor software. Dacă privim mai bine conceptul de calitate, în special **calitatea proiectului**, descoperim că originalitatea și creativitatea umană sunt strâns legate de aceasta. Aceste aspecte ale calității sunt greu de măsurat, cu atât mai mult cu cât programatorii își văd munca mai degrabă ca pe o operă de artă, decât ca pe un produs comercial.

1.2 Preocupări în domeniul măsurării calității produselor software

În anii '70, în SUA s-au întreprins primele studii și cercetări pentru măsurarea calității în domeniul produselor software. Obiectivele acestor studii au fost:

- definirea cadrului conceptual al calității software-ului, pentru a putea specifica obiectivele calității pentru produsele software;
- evaluarea calității intrinseci a produselor software pe parcursul dezvoltării sale. Să reamintim trei dintre caracteristicile majore ale software-ului și anume:
- software-ul poate fi considerat un produs complex dacă definim complexitatea prin numărul de componente elementare care-l constituie (software de câteva sute de mii de instrucțiuni este acum ceva obișnuit);
- software-ul este un produs imaterial, intangibil;
- software-ul este un produs unic, adică realizarea sa se oprește în momentul în care prototipul este sau este gata pentru vânzare. Software-ul nu are o fază de producție de serie. În cazul livrării către mai mulți clienți, el este doar multiplicat. Acestor caracteristici li s-au adăugat începând cu anii '70 constatările economiștilor, printre care amintim: creșterea exponențială a costurilor, **absența managementului proiectelor**, numeroase eșecuri în dezvoltarea unor produse din domeniul IT, insatisfacția utilizatorilor și mentenanță defectuoasă. Concluziile studiilor au fost grupate pe două direcții de acțiune⁵:
- raționalizarea producției prin : descrierea proceselor de dezvoltare a software-ului, descompunerea proceselor în faze succesive, definirea entităților (activități și resurse) care pot fi planificate, administrate și evaluate;
- definirea și punerea în aplicare a procedurilor și tehnicilor care să permită managementul calității produselor software în timpul dezvoltării acestora.

Pe parcursul studiilor au fost identificate componentele calității software-ului din punctul de vedere al utilizatorului. În identificarea criteriilor a trebuit să se răspundă la întrebarea următoare: pentru a atinge nivelul calității cerut de un factor dat, care sunt criteriile asupra cărora echipa de dezvoltare a software-ului poate acționa? Criteriile au fost apoi legate de un ansamblu de atribute măsurabile, care au fost numite "**metrici**".

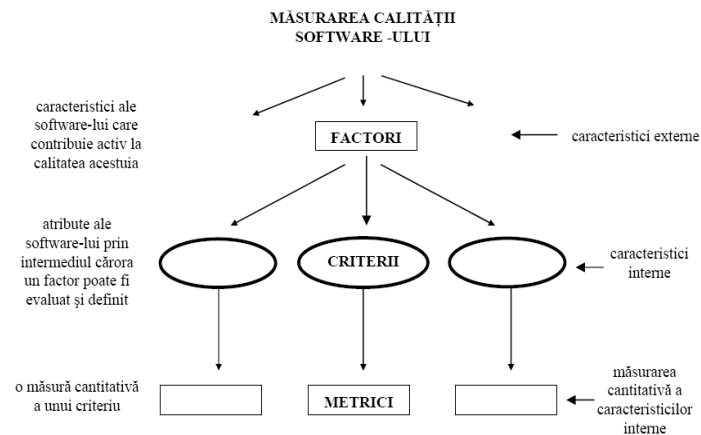


Fig 1. Modelul ierarhic al măsurării calității software-ului⁶

În fig.1 este prezentat modelul ierarhic al măsurării calității software-ului realizat în acest mod. Această abordare a fost sugerată pentru prima dată de B. Boehm și echipa sa, apoi de Mc.Call, Richards și Walters, care au început și

⁴ www.com.ase.ro/curs-finante/capitolul_5.htm In aceeași sursă *J.C. Maxwell considera că "a măsura înseamnă a ști"*.

⁵ www.ase.ro/curs-finante/capitolul_5.htm

⁶ www.biblioteca.ase.ro/downres.php?tc=1478

demersurile privind standardizarea modelului și a terminologiei aferente. În privința standardizării, una din organizațiile de standardizare din SUA, „Institute of Electrical and Electronics Engineers” (IEEE) a realizat în 1988 un proiect de standard care a apărut în 1992 ca standard - IEEE STD 1061:1992 "Standard for Software Quality Metrics Methodology"⁷; în 1991 și organizația de standardizare internațională „International Organization for Standardization” (ISO) împreună cu organismul internațional de standardizare din domeniul electrotehnic „International Electrotechnical Commission” (IEC) au realizat un standard și anume ISO /IEC 9126:1991 "Information technology – Software product evaluation – Quality characteristics and guideline for their use"⁸.

1.3 Particularități ale procesului de măsurare a calității produselor software

Măsurarea calității produselor software a fost mult timp neglijată datorită dificultăților menționate anterior. În prezent marii producători de hardware și software recunosc importanța testării și evaluării nivelurilor calității. Încă din anul 1986 compania Siemens a făcut deosebiri între categoriile de observare, evaluare și estimare (predicție)⁹. Astfel *observarea* este un proces prin care se apreciază valoarea caracteristicilor unui produs în timpul utilizării. De exemplu, timpul de răspuns al programelor este observat. Scopul acestei măsurări este de a evalua conformitatea cu proiectul.

Prin *evaluare*, raportarea unui tip de caracteristică așteptată are loc folosind un produs în diferite faze de dezvoltare cât și într-o aplicare simulată.

Estimarea permite evaluarea valorii anticipate a unei caracteristici cu ajutorul descrierii unui model al produsului sau al contextului aplicației. De exemplu, timpii de răspuns într-un sistem informațional pot fi estimați folosind predicții pe un model simulat. Dezavantajul folosirii observației constă în faptul că dezvoltarea produsului este deja terminată, iar erorile pot deja să existe. Prin estimare pot fi evitate parțial aceste dezavantaje, din moment ce vorbim aici de o observare „ipotetică” asupra tipurilor de caracteristici. Una dintre diferențele dintre predicție (estimare) și observare constă în faptul că, condițiile implementării produsului sau aplicației produsului sunt numai parțial cunoscute. O altă diferență este aceea că în cazul predicției ideile se aplică pe un model de produs abstract. Pentru procesul de dezvoltare al produselor software, se poate aplica postulatul lui Galileo „*numără ce este numărabil, măsoară ce este măsurabil și dacă ceva nu este măsurabil, încearcă să-l faci măsurabil*”.¹⁰

În comparație cu tehnicile clasice de măsurare, tehnicile de măsurare în domeniul dezvoltării produselor software au făcut doar primii pași. Prin măsurare, în domeniul IT, se înțelege obținerea unei mulțimi de valori ale unei metrici a unui proces de dezvoltare, a unui proces de mentenanță sau a unui produs software prin intermediul unor instrumente (de exemplu: prin analize statice, colectare de date de proiectare și prin sisteme de evaluare a proiectului).

Procesul de măsurare implică parcurgerea următoarelor etape:

- determinarea obiectelor măsurării;
- fixarea metricilor măsurării și a scalelor măsurării;
- alocarea metodelor de măsurare și a instrumentelor de măsurare;
- determinarea valorilor;
- interpretarea rezultatelor.

Măsurarea caracteristicilor produselor software presupune elaborarea unei descrieri cantitative adecvate, prin cuantificare, care să faciliteze și să pregătească măsurarea propriu-zisă.¹¹ În urma operațiilor de cuantificare numerică se obțin *metricile*. Măsurarea unui atribut care nu depinde de măsurarea altui atribut se numește *măsurare directă*. Rezultatul măsurării directe caracterizează atributele simple, cu preponderență atributele interne ale produsului. Măsurarea unui atribut care necesită măsurarea unuia sau mai multor atribute ale aceleiași entități sau ale entităților din alte clase se numește *măsurare indirectă*. Rezultatul măsurării indirecte se utilizează preponderent pentru caracterizarea atributelor externe ale produselor, în scopuri de evaluare, control și predicție. În acest context, numeroase lucrări de specialitate descriu experimente ale căror rezultate evidențiază existența unor relații de dependență și de interdependență între atributele interne și atributele externe ale software-ului, fapt foarte important

⁷ [IEEE Recommended Practice For Software Acquisition - IEEE Std 1062 ...](http://ieeexplore.ieee.org/iel4/5977/16011/00741938.pdf) la adresa ieeexplore.ieee.org/iel4/5977/16011/00741938.pdf

⁸ ieeexplore.ieee.org/iel4/5977/16011/00741938.pdf

⁹ www.biblioteca.ase.ro/downres.php?tc=1478

¹⁰ www.scribd.com/doc/57148800/13/Masurarea-calitatii-produselor-software

¹¹ *Cuantificarea* constituie operația de introducere a conceptelor cantitative, de trecere în studiul unui fenomen de la aprecieri atributive la concepte cantitative.

pentru domeniul măsurării și estimării calității software-ului, deoarece este mult mai ușor să avem măsurări directe ale atributelor interne decât ale atributelor externe.

Estimarea este precedată și în același timp, condiționată de operațiile de cuantificare și măsurare. Prin operația de estimare se “aplică” modelul (definit și specificat cantitativ) la datele obținute prin măsurare (prin tehnici și instrumente de analiză statistică și testare dinamică). Este vorba de estimare deoarece nu se cunoaște cât de complet este testat produsul, iar caracteristica de calitate estimată este determinată și de alte proprietăți pe care modelul sau metrica respectivă nu le ia în considerare. Deci, se aproximează valoarea caracteristicii. Analiza și validarea determină statistic dacă există o relație semnificativă între valorile (colectate și măsurate) ale metricilor și valorile (estimate) ale caracteristicilor.

Estimarea poate fi necesară în următoarele situații și scopuri:

- evaluarea calității produsului: înainte de livrare, în raport cu obiectivele referitoare la calitate, prin testarea sistemului; înainte de acceptare, în raport cu cerințele utilizatorului prin testarea de acceptabilitate; compararea produselor software, în raport cu un set de criterii; selectarea unui produs software dintr-o clasă de produse, în raport cu un set de criterii;
- controlul calității produsului software în scopul îmbunătățirii calității prin acțiuni corective/preventive;
- estimarea calității produsului software la momentul t_i+1 , pe baza datelor obținute la momentul t_i .

Modalitățile concrete de lucru, pentru atribuirea valorilor, conduc la constituirea unor tipuri de scale sau niveluri de măsură. În elaborarea regulilor de atribuire a valorilor trebuie acceptate anumite tipuri de scale.

În domeniul măsurării calității produselor software se pot utiliza patru tipuri de scale: nominală, ordinară, interval și proporțională. Primele două tipuri se mai numesc și scale neparametrice, iar ultimele două tipuri formează categoria scalelor parametrice¹². Cunoașterea proprietăților tipurilor de scale prezintă importanță deoarece s-a dovedit faptul că o serie determinată de date permite să se adopte un anumit tip de scală. Totodată se impune să menționăm că tipurile de scale influențează în mod direct operațiile de calcul în cadrul analizei și prelucrării rezultatelor obținute. Fiecare tip de scală are la bază anumite ipoteze în ceea ce privește relația dintre atributele entității și sistemul de măsurare (a se vedea Tabelul 1 Proprietăți ale tipurilor de scale)¹³.

Proprietăți ale tipurilor de scale

Nr.	Tipuri de scale	Relații matematice	Transformări admisibile	Tendința centrală	Măsurarea corelației
1	Nominală	-echivalență	substituție	grupul modal	Coefficient de corelație a contingenței
2	Ordinală	-echivalență -ordine	transformare monotonă	mediana	Coefficient de corelație a rangurilor
3	Interval	-echivalență -ordine -raport între două intervale	transformare liniară	media aritmetică	Coefficient de corelație
4	Proporțională	-echivalență -ordine -raport între două intervale -raport între două valori de pe scală	transformare de similaritate	media geometrică	Coefficient de corelație

Atunci când informațiile referitoare la atributele entităților devin mai bogate, se poate trece de la utilizarea unui tip de scală la alt tip (de la scala nominală la cea proporțională). În elaborarea unei scale este necesar să se aibă în vedere două criterii: scala trebuie să fie accesibilă subiecților care fac aprecierea și trebuie să diferențieze nivelele de intensitate ale atributelor entităților.

a) Scala nominală, cea mai simplă din punct de vedere al capacității de măsurare și cea mai puțin restrictivă din punct de vedere statistico-matematic permite clasificarea entităților în grupe ale căror componente diferă după atributul ce a fost scalat, fără să ducă și la realizarea unei ordonări a entităților în funcție de intensitatea atributelor sau la măsurarea distanțelor care le separă. De exemplu, programele sunt grupate astfel: ”extrem de dificil de înțeles”, ”foarte dificil de înțeles”, ”dificil de înțeles”.

b) Scala ordinală corespunde situațiilor în care atributele studiate se pot ierarhiza în funcție de intensitatea lor, folosindu-se valori ordinale: primul, al doilea, pe această bază putându-se indica rangul fiecărei poziții, fără a se

¹² Balog, A., -*Analiza statistică și evaluarea calității software-ului*, Editura Calipso, București, 1997

¹³ Balog, A., -*Analiza statistică și evaluarea calității software-ului*, Editura Calipso, București, 1997

putea preciza cu cât este mai mică sau mai mare una în raport cu alta. Această scală nu permite însă evaluarea distanțelor dintre variante. De exemplu, putem spune nu numai că toate programele A, B și C sunt dificil de înțeles, ci se poate spune că programul B este mult mai dificil de înțeles decât programul C, iar programul A este mult mai dificil de înțeles decât programul B. Analiza statistică a informațiilor cuantificate cu ajutorul scalei ordinare, deși rămâne tot de natură neparametrică, poate fi mai elaborată decât în cazul scalei nominale.

c) Scala interval (cardinal) corespunde situațiilor în care atributul măsurat poate fi descris cantitativ (cuantificat) prin utilizarea valorilor numerice, fără însă a exista un punct natural de pornire în atribuirea acestora. Se obține astfel un șir de valori egal distanțate pe scală. Datorită faptului că valorile atribuite sunt numere, la acest tip de scală se poate indica cu cât este mai mare o valoare decât alta și în consecință se pot efectua operații de adunare și scădere a intervalelor. De exemplu, se poate spune că programul A este mult mai dificil de înțeles decât programul B cu 10 “unități de dificultate”.

d) Scala proporțională (de raport) corespunde situațiilor în care atributul măsurat poate fi descris cantitativ luându-se în considerare valoarea zero ca punct de pornire. Scala proporțională permite efectuarea tuturor operațiilor admise de celelalte scale, inclusiv multiplicarea sau divizarea unui număr de pe scală. De exemplu, se poate indica faptul că programul A este de două ori mai dificil de înțeles decât programul B. Informațiile cuantificate cu ajutorul scalei proporționale pot fi analizate prin oricare dintre metodele oferite de statistică.

2. Calitatea și modelul ei în domeniul tehnologiei informației

Cercetările specialiștilor au condus la elaborarea unui model conceptual al sistemului de caracteristici, cunoscut sub denumirea de „*modelul calității software-ului*”. Modelul a fost sugerat pentru prima dată de Barry Boehm și colaboratorii săi de la TRW Systems Group (SUA). Ulterior, James McCall, P. Richard și G. Walters de la General Electric Company au contribuit la perfecționarea și extinderea acestuia. Modelul este recomandat de Organizația Internațională de Standardizare (ISO) prin standardul ISO/IEC 9126:1991.

Caracteristicile calității servesc la definirea cerințelor utilizatorului și la stabilirea obiectivelor calității pentru produsul software. Unii specialiști numesc acest nivel, ca fiind nivelul *factorilor*. Pe următorul nivel, se identifică un set de *caracteristici derivate* ale calității care se combină în seturi de *criterii* necesare pentru evaluarea caracteristicilor de pe primul nivel. Caracteristicile derivate au semnificație tehnică și sunt relevante pentru personalul de specialitate (analisti, proiectanți, programatori, etc.). Ele facilitează comunicarea între șeful de proiect și personalul de specialitate în ceea ce privește atingerea obiectivelor calității. Caracteristicile derivate se definesc pentru componentele produsului software și pentru diferitele forme de reprezentare a caracteristicilor derivate “modularitate” se exprimă anume: caracteristici-atribute în modelul Boehm, factori-subfactori în modelul McCall. Trebuie subliniat că modelele diferă atât prin modul de ierarhizare a caracteristicilor și caracteristicilor derivate, cât și prin definițiile utilizate. (specificații, cod sursă, etc.) pe parcursul ciclului de viață al produsului. La ultimul nivel se definesc *metricile*, care permit măsurarea caracteristicilor derivate (a criteriilor) și, în consecință, exprimarea cantitativă și estimarea caracteristicilor (a factorilor) de pe primul nivel. Pentru exemplificare, cantitativ prin metricile următoare: gradul de independență între modulele unui program, nivelul de structurare ierarhică a produsului și gradul de coeziune a componentelor.

2.1 Modele tradiționale ale calității produselor software. În această categorie includem modelele calității care au stat la baza dezvoltărilor actuale incluse în standardele internaționale. Cele mai cunoscute sunt modelele în anii '70 de grupul de cercetare de la TRW Systems Group din SUA, condus de B.W. Boehm și modelul dezvoltat la General Electric Company de J. McCall și colaboratorii săi. Există și alte modele menționate în lucrările de specialitate, însă acestea reprezintă extensii ale modelelor clasice amintite. Prezintăm succint numai semnificația caracteristicilor și atributelor incluse în modele.

a) Modelul calității produselor software propus de Boehm (1978). În figura 2 este prezentat modelul calității propus de Boehm cu evidențierea caracteristicilor și a relațiilor dintre ele.

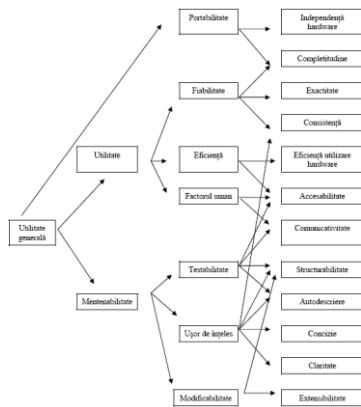


Fig. 2 Modelul calității produselor software propus de Boehm¹⁴

Tabel nr 1 Semnificația caracteristicilor calității produselor software (modelul propus de Boehm)

Nr.	Caracteristică derivată	Semnificația caracteristicii derivate
1	Independența	Gradul în care produsul software poate fi executat pe configurații hardware altele decât cele definite inițial.
2	Completitudine	Gradul în care sunt prezente toate părțile componente ale produsului software și fiecare parte este complet realizată.
3	Acuratețe (exactitate)	Gradul în care ieșirile furnizate de produsul software sunt suficient de precise pentru a satisface cerințele impuse.
4	Consistență	Gradul în care produsul software conține notații, termeni și simboluri unitare.
5	Eficiența utilizare hardware	Gradul în care produsul software utilizează în mod economic hardware-ul.
6	Accesibilitate	Gradul în care produsul software facilitează utilizarea selectivă a componentelor sale.
7	Structurabilitate	Gradul în care produsul software prezintă, într-o manieră organizată, componentele sale.
8	Concizie	Gradul în care produsul software nu conține informații excesive.
9	Claritate (lizibilitate)	Gradul în care funcțiunile produsului software pot fi identificate prin citirea codului sursă.
10	Extensibilitate	Gradul în care produsul software facilitează extinderea sa cu noi funcțiuni de prelucrare.

Tabel nr 2 Semnificația caracteristicilor calității produselor software (modelul propus de Boehm)¹⁵

Nr.	Caracteristică	Semnificația caracteristicii
1	Portabilitate	Gradul în care produsul software poate fi exploatat ușor și în bune condiții pe alte configurații de calcul decât cele presupuse inițial.
2	Fiabilitate	Gradul în care produsul software execută fără erori, funcțiunile impuse.
3	Eficiență	Gradul în care produsul software execută funcțiunile sale fără să irosească resursele.
4	Factorul uman	Gradul în care produsul software îndeplinește scopul fără a risipi timp și energie din partea utilizatorului.
5	Testabilitate	Gradul în care produsul software permite stabilirea unui criteriu de acceptare și suportă evaluarea performanțelor sale.
6	Ușurința înțelegerii	Gradul în care obiectivele și scopul produsului software sunt clar definite.
7	Modificabilitate	Gradul în care produsul software facilitează efectuarea schimbărilor, odată ce a fost determinată natura schimbării.

Tabel 3 Semnificația caracteristicilor derivate ale calității produselor software (Boehm)¹⁶

¹⁴ Gillies, A., *Software Quality – Theory and Management*, International Thomson Computer Press, London, UK, 1997)

¹⁵ Sursa: Balog, A., *Analiza statistică și evaluarea calității software-ului*, Editura Calipso 2000, București, 1997

¹⁶ Sursa: Balog, A., *Analiza statistică și evaluarea calității software-ului*, Editura Calipso 2000, București, 1997

Nr.	Caracteristica derivată	Semnificația caracteristicii derivate
1	Independența	Gradul în care produsul software poate fi executat pe configurații hardware altele decât cele definite inițial.
2	Completitudine	Gradul în care sunt prezente toate părțile componente ale produsului software și fiecare parte este complet realizată.
3	Acuratețe (exactitate)	Gradul în care ieșirile furnizate de produsul software sunt suficiente de precise pentru a satisface cerințele impuse.
4	Consistență	Gradul în care produsul software conține notații, termeni și simboluri unitare.
5	Eficiența utilizare hardware	Gradul în care produsul software utilizează în mod economic hardware-ul.
6	Accesibilitate	Gradul în care produsul software facilitează utilizarea selectivă a componentelor sale.
7	Structurabilitate	Gradul în care produsul software prezintă, într-o manieră organizată, componentele sale.
8	Concizie	Gradul în care produsul software nu conține informații excesive.
9	Claritate (lizibilitate)	Gradul în care funcțiunile produsului software pot fi identificate prin citirea codului sursă.
10	Extensibilitate	Gradul în care produsul software facilitează extinderea sa cu noi funcțiuni de prelucrare.

b) Modelul calității produselor software propus de McCall (1977) Modelul calității produselor software propus de McCall, este prezentat în fig.3, cu evidențierea caracteristicilor calității și a relațiilor dintre ele.

c) Modelul calității produselor software definit de standardul ISO/ IEC 9126

Modelul calității ISO/IEC 9126 recomandă utilizarea următoarelor caracteristici principale: funcționalitate, utilizabilitate, fiabilitate, eficiență, mentenabilitate și portabilitate. Datorită terminologiei utilizate, modelul este denumit „*modelul caracteristici principale-caracteristici derivate- metrici*”.

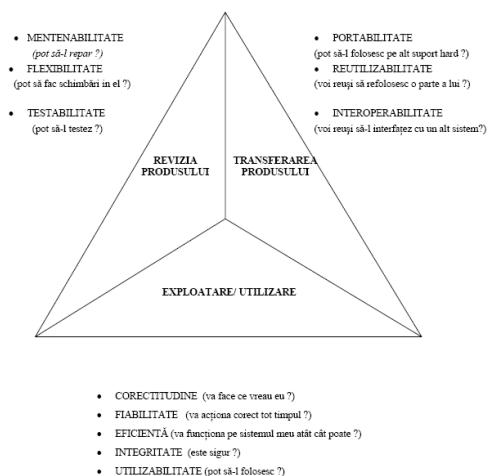


Fig. 3 Modelul calității produselor software propus de McCall¹⁷

Tabelul 4 Caracteristicile și semnificația caracteristicilor derivate din modelul ISO/ IEC 9126

¹⁷ Sursa: Gillies, A., *Software Quality – Theory and Management*, International Thomson Computer Press, London, UK, 1997

Nr.	Caracteristica	Semnificația caracteristicii
1	Corectitudine	Gradul în care produsul software satisface specificațiile și îndeplinește cerințele utilizatorului.
2	Fiabilitate	Gradul în care produsul software execută funcțiunile sale cu precizia solicitată.
3	Eficiența	Volumul resurselor de calcul și a codului sursă utilizat de produsul software pentru a realiza funcțiunile.
4	Integritate	Gradul în care poate fi controlat accesul în produsul software și în date de către persoane neautorizate.
5	Utilizabilitate	Efortul necesar pentru „învățarea produsului software” și pregătirea datelor de intrare.
6	Mentenanabilitate	Efortul necesar pentru localizarea și corectarea erorilor într-un produs software operațional.
7	Testabilitate	Efortul necesar pentru a testa un produs software în scopul asigurării funcționalității.
8	Flexibilitate	Efortul necesar pentru a modifica un produs software operațional.
9	Portabilitate	Efortul necesar pentru a transfera produsul software în alt mediu hardware-software.
10	Reutilizabilitate	Gradul în care produsul software poate fi utilizat în alte aplicații.
11	Interoperabilitate	Efortul necesar pentru a cupla produsul software cu alte produse.

2.2 Aspecte specifice ale modelelor actuale ale calității produselor software.

Un aspect important al modelelor calității produselor software îl reprezintă **divizarea ierarhică a caracteristicilor calității**, metricile fiind la nivelul de bază. În funcție de modelul calității stabilit și de cerințele specifice ale produsului, se determină, pentru fiecare proiect, un număr specific de caracteristici principale, caracteristici derivate și de metrici. Numărul acestora poate varia considerabil. De exemplu, în cazul unui produs cu cerințe deosebite privind mentenanță, trebuie să se acorde atenție suplimentară modularității arhitecturii software, clarității și ușurinței de înțelegere a codului în proiectare și codificare. Aceste însușiri pot fi în relație inversă cu eficiența codului executabil, de exemplu, un program cu o structură modulară clară și un cod clar poate fi mai încet în execuție decât unul nestructurat. Rezultă că pot fi conflicte de țeluri în definirea diferitelor caracteristici derivate. În general, în cazul multor proiecte se specifică numai cerințele funcționale, cerințele de performanță și cerințele interfeței cu utilizatorul. Cerințele privind ușurința de întreținere, portabilitatea sau caracteristici similare, sunt în majoritatea cazurilor ignorate. Motivul este că cei responsabili cu specificația nu au suportul necesar pentru formularea acestor cerințe privind calitatea. Modelele calității și metricile calității oferă posibilitatea de a specifica aceste cerințe și ajută la îmbunătățirea calității procesului. Problemele tipice ale procesului sunt de exemplu, o planificare și un control nesatisfăcătoare ale proiectului, nedeterminarea costului de dezvoltare la începutul proiectului și întârzierea în livrarea produsului. Alegând metricile potrivite pentru factori specifici ai procesului, cum ar fi planificarea proiectului și leadership, sau proceduri standardizate în cadrul proiectului, problemele referitoare la calitatea produsului sau procesului pot fi mai ușor rezolvate. Modelele calității ne pot ajuta, de asemenea, să înțelegem și să demonstrăm, mai bine, relația dintre caracteristicile calității produselor software. Structura sistemului de caracteristici evidențiază anumite relații de subordonare și de dependență ce sunt reflectate în mod diferit de modelele calității. Astfel, se pot identifica următoarele tipuri de relații: ierarhice, de influență de tip conflictual și de influență de tip complementar.

a) Relațiile ierarhice, de subordonare, care sunt definite prin structura sistemului de caracteristici și ele diferă de la un model la altul. Aceste relații fac caracteristicile respective să fie *“neconectate”*. Două caracteristici ale calității sunt neconectate dacă nu există o interacțiune vizibilă între ele, de exemplu dacă nu există o interacțiune vizibilă între caracteristicile derivate. De exemplu, portabilitatea și corectitudinea sunt considerate ca fiind neconectate.

b) Relațiile de influență de tip conflictual, semnifică faptul că pe măsură ce crește nivelul unei caracteristici, este de așteptat ca nivelele celorlalte caracteristici să scadă. Aceste relații fac caracteristicile respective să fie *“competitoare”*. Două caracteristici sunt competitoare (concurrente) dacă îmbunătățirea unei caracteristici duce la diminuarea celeilalte. De exemplu, creșterea fiabilității duce, de regulă, la diminuarea performanței.

c) Relațiile de influență de tip complementar, semnifică faptul că dacă nivelul unei caracteristici crește, atunci este de așteptat ca și nivelul celorlalte caracteristici să crească. Aceste relații fac caracteristicile respective să fie *“revitalizante”*. Două caracteristici se revitalizează dacă îmbunătățirea uneia afectează îmbunătățirea celeilalte. De exemplu, corectitudinea și fiabilitatea sunt caracteristici care se revitalizează. Cunoașterea interdependențelor dintre caracteristici este necesară în toate fazele de specificare, determinare și evaluare a calității produselor software și în special, în faza de stabilire a obiectivelor calității. Modelele calității sunt de mare ajutor pentru îmbunătățirea influenței personalului independent, cu atribuții definite referitoare la asigurarea calității produselor software. Printr-o rețea bine definită de metrici în toate fazele și prin ajustarea după particularitățile proiectului, personalul independent poate să descopere erorile din procesul de dezvoltare sau în fazele intermediare, sau chiar la produsul final. Așa cum am precizat deja, calitatea este relativă și se bazează mereu pe cerințe predeterminate. Aceste cerințe predeterminate sunt desigur foarte strâns legate de persoana care le-a definit. Persoane din medii diferite având de-a face cu dezvoltarea produsului software, au idei diferite despre calitatea acestuia. Marea majoritate a specialiștilor

din domeniul IT definesc patru categorii de persoane, implicate în domeniul IT, care au opinii diferite privind calitatea produselor software: utilizatorii, managerii, proiectanții și programatorii.

Cerințele utilizatorilor de produse software privind calitatea acestora afectează, de obicei, interfețele produsului. Managerii sunt aceia care furnizează produse software în zona lor de influență, de exemplu ei fac produsele disponibile pentru utilizator,¹⁸ organizează utilizarea și determină ciclul de viață. Cerințele acestei categorii sunt, de regulă, orientate către aplicabilitate și perspectivele produsului software.

Proiectanții determină structura tehnică a produsului ca o combinație de componente cu anumite funcții. Țelurile privind calitatea, ale acestei categorii, afectează, de regulă, structurabilitatea arhitecturii și satisfacerea cerințelor prin dezvoltarea continuă a produsului.

Programatorii creează componentele produsului software în conformitate cu funcțiile prescrise, sub forma de programe și module. Obiectivele lor privind calitatea se referă de obicei, la structura detaliată de programare, la stilul de programare și la algoritmi utilizați. În ceea ce privește aplicarea modelelor calității, se constată că majoritatea organizațiilor întâmpină dificultăți în specificarea și evaluarea calității datorită deficiențelor acestor modele. Luând în considerare rezultatele publicate de specialiști din diferite firme pe plan mondial, se pot menționa următoarele aspecte critice privind modelele calității produselor software:

- unele caracteristici ale calității se aplică la tipuri specializate de produse software, altele la majoritatea produselor, iar altele se referă la procesul de realizare, nu la produsul software;
- natura caracteristicilor derivate este foarte diversă: unele sunt descrieri mai detaliate ale caracteristicilor principale, altele reprezintă proprietăți necesare pentru a asigura că în produsul final este prezentă caracteristica calității dorită, iar alte caracteristici sunt atribute (proprietăți) ale procesului de realizare;
- metricile nu sunt măsurători obiective și nu reprezintă întotdeauna aspectul cantitativ, deoarece ele apar și sub formă de liste de verificări, proceduri de control etc.
- nu există relații explicite între caracteristici derivate-metrici și etapele din ciclul de viață al produselor software;
- caracteristicile calității nu sunt definite în termeni măsurabili, astfel încât este dificilă validarea relațiilor dintre caracteristici și metrici;
- caracteristicile principale și caracteristicile derivate calității sunt definite în mod inconsistent: unele se definesc din punctul de vedere al celor care dezvoltă produsele software, iar altele din punctul de vedere al utilizatorului acestora. Indiferent de modelul calității agreat la un moment dat este de dorit să se abordeze problema evaluării calității ca un proces sistematic, luând în considerare următoarele reguli:
 - identificarea punctelor specifice ale aplicației (clasificarea produselor software);
 - determinarea caracteristicilor importante ale calității (maxim trei);
 - definirea metricilor pentru caracteristici individuale, care pot fi ușor identificate;
 - organizarea metricilor în faze, astfel încât să poată fi măsurate și determinate;
 - definirea metodelor de măsurare și a listelor de verificare pentru evaluarea criteriilor necuantificabile;
 - programarea măsurărilor;
 - selectarea instrumentelor simple, care să ușureze măsurarea și evaluarea;
 - stabilirea limitelor și valorilor așteptate pentru metrici;
 - determinarea valorilor metricilor;
 - analiza valorilor obținute ale metricilor în raport cu valorile prescrise / așteptate;
 - inițierea de acțiuni corective dacă există o abatere de la valorile limită, și verificarea aplicării lor.

3 Metricile și rolul lor în domeniul evaluării calității produselor software

3.1 Definirea metricilor și a caracteristicilor acestora. În lucrările de specialitate există opinii diferite privind semnificația conceptului “metrici” și raportul între conceptele “metrici”, “măsuri” și “indicatori”, în evaluarea calității software-ului. După cum se menționează în lucrarea *Software Metrics: A Rigorous Approach*¹⁹, termenul “metrici software” a fost introdus prima dată de Rubey și Hartwick, având semnificația de măsuri cantitative ale atributelor (proprietăților) software. Conceptul a fost preluat și utilizat în majoritatea lucrărilor de specialitate și standardele din domeniul calității software. Utilizarea curentă a conceptului “metrici” este legată de cercetările vizând cuantificarea, măsurarea și evaluarea calității software-ului.

¹⁸ Sursa: Walmüller, E., *Software Quality Assurance – A Practical approach*, Prentice Hall Int., UK, 1994)

¹⁹ **Bowman, J.B., Newman, A. William** - *Software Metrics as a Programming Training Tool* - Journal of Systems Software nr.13, 1990- pag. 139-147

a) **Metrica** (conform standardului IEEE –STD-1061:1992 - Standard for a Software quality metrics – Methodology²⁰) reprezintă nivelul standard pentru măsura cantitativă a unei caracteristici principale sau derivate; nivelul standard poate fi o valoare absolută, o valoare relativă sau un interval de valori. Pentru a facilita înțelegerea și aplicarea diverselor recomandări metodologice pentru măsurarea, analiza și evaluarea calității se face distincție între metrica asociată unei caracteristici derivate și metrica directă a unei caracteristici principale.

b) **Măsura** reprezintă expresia cantitativă a unei caracteristici a calității și are următoarele trăsături:

- definește relația dintre caracteristica calității ce trebuie măsurată și proprietățile software-ului care determină sau influențează caracteristica respectivă;
- asigură determinarea obiectivă a diferitelor nivele ale caracteristicii de calitate și asigură încadrarea nivelului într-o scală de măsurare. Potrivit standardului ISO-CEI 9126:1991 – *Information technology – Software product evaluation – Quality characteristics and guidelines for their use*²¹, *metrica* este o scală cantitativă și metoda care poate fi utilizată pentru a determina valoarea pe care o ia un atribut (proprietate) a unui produs software specific. Din ambele definiții (care nu diferă mult, ci mai degrabă se completează) rezultă că metricile au rolul exclusiv de a măsura nivelul unei proprietăți a software-ului cu luarea în considerație a dependențelor dintre caracteristicile calității).

c) Pentru exprimarea cantitativă a proprietăților unui produs software în timpul procesului de realizare se utilizează frecvent termenul de „*indicatori*”, preferându-se utilizarea termenului „*metrici*” pentru produsul software finalizat sau aflat în utilizare curentă. Metricile software se pot clasifica în două tipuri și anume: predictive sau descriptive.²²

O *metrică predictivă* se folosește pentru a face predicții despre software, în etapele ciclului de viață al produselor software. De exemplu, structurabilitatea este folosită pentru a prevedea mentenabilitatea produsului software în utilizare. O *metrică descriptivă* descrie stadiul software-ului în momentul măsurării. De exemplu, o metrică a fiabilității se poate baza pe numărul de căderi ale sistemului în timpul unei perioade date. Cei doi pionieri ai modelelor ierarhice, McCall și Boehm, au adoptat o abordare diferită a metricilor. McCall a folosit rezultatul unor ecuații, pe când Boehm a folosit metode bazate pe liste de verificare care cereau răspunsuri cu da/nu. În 1987 Watts recomanda ca evaluarea calității metricilor să se facă prin intermediul caracteristicilor mentenabilitate, fiabilitate, utilizabilitate, extensibilitate și integritate. Multe metrici se bazează încă pe subiectivitate. IBM, de exemplu, măsoară capacitatea de folosire a produselor software prin supravegherea utilizatorilor lor²³. Multe măsuri ale utilizabilității se bazează pe judecata utilizatorului și este dificil să se vadă cum o măsură efectivă a utilizabilității poate fi eliberată de subiectivism. Criteriul care este cel mai greu de demonstrat este validitatea, neexistând un standard în domeniu. Rezumând, putem spune că pentru măsurarea calității avem nevoie de evaluări simple care să fie expresive și relevante.

3.2 Metricile și evaluarea calității produselor software. Încă din anul Rubey și Hartwick au publicat o lucrare despre „Măsurarea cantitativă a calității unui program”. Conform lui Conte, Dunsmore și Shen metricile produsului software cuprind dimensiunea produsului (ex. numărul de linii ale codului sau numărul simbolurilor terminale într-un program), complexitatea structurii (de exemplu: fluxul de control, adâncimea de imbricarea, adâncimea de recursivitate), complexitatea structurii datelor (de exemplu: numărul de variabile folosite sau fișierele datelor) și zona de aplicare a produsului (de exemplu: salarii/ număr de personal) sau combinații ale acestora. În timp, măsurările produsului sau procesului au crescut ca importanță fiind utilizate în legătură cu modele pentru explicarea procesului de dezvoltare și mentenanță a software-ului. Aceste modele servesc, de obicei, pentru estimarea productivității sau a costului proceselor. Watts a luat în considerare 40 de metrici când a definit cele șapte caracteristici ale unei metrici corespunzătoare. Din distribuția acestora, pentru măsurarea caracteristicilor calității urmărite, rezultă că metricile nu sunt uniform distribuite. Trei pătrimi din cele de metrici se referă la două caracteristici ale calității: fiabilitatea și mentenabilitatea. Pentru patru caracteristici ale calității nu este disponibilă nici o metrică, iar trei caracteristici ale calității au asociate numai o singură metrică. Această răspândire neuniformă este amplificată de faptul că multe dintre metrici se bazează pe aceleași proprietăți fundamentale măsurabile. De exemplu, complexitatea este folosită atât la fiabilitate cât și la mentenabilitate.

²⁰ Bush, E. Martin, Fenton, E. Norman - *Software Measurement: A Conceptual Framework* -Journal of System Software nr.12, 1990- pag. 223-231

²¹ <http://www.software-metrics.ase.ro/articole/METRICI%20%20SOFTWARE.htm>

²² <http://www.software-metrics.ase.ro/articole/METRICI%20%20SOFTWARE.htm>

²³ <http://www.software-metrics.ase.ro/articole/METRICI%20%20SOFTWARE.htm>

Tabelul 5 . Metricile disponibile pentru măsurarea fiecărei caracteristici a calității (după Watts, 1987)

Caracteristica a calității	Număr de metrici disponibile	Caracteristica a calității	Număr de metrici disponibile
Mentenabilitate	18	Portabilitate	1
Fiabilitate	12	Eficiență	0
Utilizabilitate	4	Adaptabilitate	0
Corectitudine	3	Interoperabilitate	0
Integritate	1	Reutilizabilitate	0
Extensibilitate	1		

După Wallmüller²⁴ există aproximativ 100 de metrici în literatură care pot fi luate în considerare pentru evaluarea complexității unui program software. Fiecare dintre aceste metrici oferă o imagine diferită asupra complexității produselor software. Aceste metrici au fost împărțite după următoarele elemente:

posibilitatea de a fi analizat, complexitatea nivelului unui modul, predicția/ frecvența erorilor, posibilitatea de a fi modificat, modularitate, independența sistemului (hardware sau de operare), posibilitatea de testare, înțelegerea textului program.

S-a urmărit ca fiecare metrică să fie alocată unei caracteristici a calității. S-au avut în vedere următoarele caracteristici ale calității: utilizabilitate, corectitudine, fiabilitate, eficiență, securitate, mentenabilitate, adaptabilitate, extensibilitate, posibilitatea de conectare, reciclabilitate, și transferabilitate. S-a constatat că marea majoritate a metricilor care există măsoară mentenabilitatea și fiabilitate. Itzfeld²⁵ a examinat folosirea metricilor în practica IT în Germania. Concluzia studiului prezentat în acest capitol este că mentenabilitatea are mare importanță în comercializarea produselor IT, depășind în importanță, corectitudinea și eficiența. Menționez **metricile utilizate în practica IT**, pentru evaluarea anumitor caracteristici ale calității și anume:

- *eficiența*: timpii de tranzacționare, timpii de implementare, numărul de operații de intrare și ieșire;
- *corectitudinea*: metrici pentru complexitate McCabe, metrici de acoperire a testelor, numărul de erori pe unitatea de timp, metrici pentru complexitate Biowowsky, metrici Moses, statistici ale mediei timpului de bună funcționare;
- *fiabilitate*: metrici de acoperire a testelor, timpii de oprire sau viteza, viteza erorilor remanente, statistici ale mediei timpului de bună funcționare;
- *mentenabilitate*: metrici pentru complexitate McCabe, numărul și dimensiunea modulelor, dimensiunea procedurilor, adâncimea de îmbricare, numărul de erori pe unitatea de timp legate numărul de linii ale codului alterate, numărul de alterări pe unitatea de timp, nivelul de documentare pentru program. Având în vedere rezultatele celor trei studii analizate putem aprecia că în domeniul IT, până în prezent, nu există un sistem unanim recunoscut pentru măsurări și metrici.

Prezentăm o aplicație vba, contribuție proprie pentru managementul erorilor și defectelor în întreținerea unei aplicații informatice.²⁶

SecuritateHard						
	C	D	E	F	G	H
5	Luna	Uptime sisteme%	Nr. Angajati pe securitate	Probleme descoperite	Rezolvate (minute)	Total rezolvare (min.)
6	ianuarie	98,72		3	190	570
7	februarie	98,56		1	191	191
8	martie	97,74		3	298	894
9	aprilie	98,56		4	161	644
10	mai	97,74		3		810
11	iunie	97,74		9		1062
12	iulie	98,15		9		999
13	august	97,74		4	252	1008
14	septembrie	98,15		3	73	219
15	octombrie	98,15		2	208	416
16	noiembrie	98,36		2	367	734
17	decembrie	98,15		6	138	628
18						
19			Completeaza nr. Angajatilor	Afiseaza grafic		

Autocompleteaza datele cu valori aleatoare

Reseteaza toate valorile

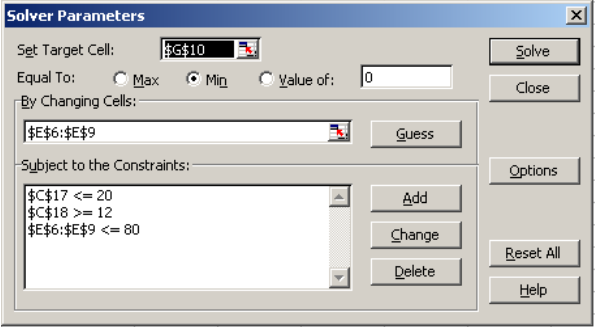
Sub Button3_Click() 'functie de randomizare a valorilor Dim r As Range ' variabila pentru setarea	Sub Button5_Click() 'functie de stergere a valorilor Dim r As Range For d = 6 To totalang + 17
---	---

²⁴Wallmüller- Software Quality Assurance – A Practical approach, Prentice Hall Int., UK, 1994

²⁵Wallmüller- Software Quality Assurance – A Practical approach, Prentice Hall Int., UK, 1994

²⁶Gramada Argentina Dragu- Aplicatii si teste Word, Excel, Vba, Frontpage-Editura Renaissance-2006

<pre> celulelor din foaia de lucru For d = 6 To totalang + 17 'se parcurg coloanele Set r = Range("f" + CStr(d)) ' se seteaza celula care se va schimba r.Value = Rand(1, 9) 'se genereaza o valoare aleatoare in intervalul [1,9] Next d 'analog pentru urmatoarea coloana For d = 6 To totalang + 17 Set r = Range("g" + CStr(d)) r.Value = Rand(45, 400) Next d End Sub </pre>	<pre> Set r = Range("f" + CStr(d)) r.Value = "" Next d For d = 6 To totalang + 17 Set r = Range("g" + CStr(d)) r.Value = "" Next d For d = 6 To totalang + 17 Set r = Range("g" + CStr(d)) r.Value = "" Next d For c = 6 To 17 Step 1 Set r = Range("e" + CStr(c)) r.Value = "" Next c End Sub Public Function Rand(ByVal jos As Long, ByVal sus As Long) As Integer 'Functie de randomizare pe un interval Rand = Int((sus - jos + 1) * Rnd) + jos End Function </pre>
--	--

<pre> Sub Button7_Click() 'Functie de calculare al nr de angajati pe luna Dim r As Range Dim c As Integer Dim d As Integer Dim luna(12) As Integer Dim totalang As Integer Dim gasit As Integer Dim e As Integer totalang = InputBox("Introduceti nr total de angajati") For c = 1 To 12 'contor pentru nr. de luni For d = 26 To totalang + 25 Set r = Range("b" + CStr(d)) 'citirea valorii din celula pentru a sti din ce luna incepe angajatul If (r.Value = c) Then For e = c To 12 luna(e) = luna(e) + 1 Next e End If Next d Next c d = 1 For c = 6 To 17 Set r = Range("e" + CStr(c)) 'setarea valorii in celula r.Value = luna(d) Let d = d + 1 Next c End Sub </pre>	
--	---

Codul vba al modulului de intretinere al aplicatie

Cell	Name	Original Value	Final Value
\$G\$6	MB SOCKET A KM266PRO GIGABYTE (7VM333M-RZ) Total valoare -stoc produs	17.413.000,00	20.000.000,00

Cell	Name	Original Value	Final Value
\$D\$6	MB SOCKET A KM266PRO GIGABYTE (7VM333M-RZ) Cost per bucata (NU TVA)	1.583.000	1.818.182

Cell	Name	Cell Value	Formula	Status	Slack
\$E\$6	MB SOCKET A KM266PRO GIGABYTE (7VM333M-RZ) Cantitate in stoc	11	=\$E\$6<=1	Not Binding	10

Raportul de raspuns arată optimizarea valorii finale in sensul scaderii cheltuielilor zilnice ocazionate de intretinerea liniei tehnologice de asamblare-realizare componente IT .

Concluzie

Posibilitatile de evaluare a produselor software , adica a ansamblelor formate din programe, proceduri și documentații asociate, pentru calculator precum și datele destinate să fie livrate unui utilizator sunt in continua devoltare si aliniere la standardele internationale.

Bibliografie

1. Balog, A., *Analiza statistică și evaluarea calității software-ului*, Editura Calipso 2000, București, 1997
2. Bowman, J.B., Newman, A. William - *Software Metrics as a Programming Training Tool* -Journal of Systems Software nr.13, 1990- pag. 139-147
3. Bush, E. Martin, Fenton, E. Norman - *Software Measurement: A Conceptual Framework* -Journal of System Software nr.12, 1990- pag. 223-231
4. Centrul International de Comert UNCTAD/GATT, Organizatia Internationala de
5. Gramada Argentina Dragu-*Aplicatii si teste Word, Excel, Vba* , *Frontpage*-Editura Renaissance-2006
6. Gillies, A., *Software Quality – Theory and Management*, International Thomson Computer Press, London, UK, 1997
7. Iuliana Ciochina si altii -*Calitatea Serviciilor in Contextul Procesului de Globalizare* -la adresa <http://mmq.ase.ro/LucrCD/sectiunea%20I/Calitatea%20serviciilor%20Ciochina,%20Iordache,%20Decus%20eara-1.htm>
8. www.biblioteca.ase.ro/downres.php?tc=1478
9. www.software-metrics.ase.ro/articole/METRICI%20%20SOFTWARE.htm
10. [IEEE Recommended Practice For Software Acquisition - IEEE Std 1062 ...](http://ieeexplore.ieee.org/iel4/5977/16011/00741938.pdf)la adresa ieeexplore.ieee.org/iel4/5977/16011/00741938.pdf
11. www.ase.ro/curs-finante/capitolul_5.htm
12. www.mendeley.com/research/metrics-tool-multilanguage-software/