

CERTIFICAREA CALITĂȚII APLICAȚIILOR DISTRIBUITE

Ion Ivan

ionivan@ase.ro

Daniel Milodin

daniel.milodin@ase.ro

Florentin Șchiopu

florentin.schiopu@gmail.com

Academia de Studii Economice București

Abstract: Se definește conceptul de certificare pentru aplicații informatice distribuite. Se prezintă aspecte legate de certificarea software. Se dezvoltă noțiuni legate de certificarea calității datelor și a rezultatelor. Se analizează caracteristicile de calitate pentru datele de intrare, rezultate și software, se construiesc indicatori angajați pentru fundamentarea procesului de certificare. Se prezintă modul în care este certificată aplicația de alocare temporară de resurse.

Cuvinte cheie: aplicații distribuite, ortogonalitate, calitate produse program, certificare.

Abstract: It is defined the certification concept for distributed informatics applications. There are presented aspects regarding the software certification. There are developed concepts regarding the data quality and results certification. There are analyzed the quality characteristics for data entry, results and software, there are built indicators certification process substantiation. There is presented the certification way for application used to assign temporary resources.

Key Words: distributed applications, orthogonality, programming products quality, certification.

1. Concepte de bază

O aplicație informatică este construită pentru a adăuga un obiectiv bine definit.

Crearea de aplicații distribuite trebuie să urmărească scopuri precise în ceea ce privește accesarea facilităților oferite de produse software prin intermediul internetului. Se prezintă o serie de obiective ce necesită crearea de aplicații distribuite:

- vânzarea on-line de produse electronice;
- rezervare on-line de camere la hotel;
- testare on-line a cunoștințelor de conducere auto;
- instruire on-line la geografie;
- învățare on-line a limbii engleze;
- contractări de produse de valoare mare și obținere de împrumuturi;
- elaborare de planuri pentru construcții;
- obținerea de documente oficiale de la autorități;
- realizarea de muzee virtuale;
- dezvoltarea de programe TV personalizate;
- consultanță medicală on-line;
- tranzacționări la bursa de valori.

Pentru fiecare aplicație de stabilește grupul țintă prin:

- estimarea numărului de utilizatori ai aplicației;
- structurarea colectivității după criterii precum: vârstă, instruire, experiență, deprinderi;
- frecvența de utilizare;
- structurile datelor de intrare, de care dispun utilizatorii;
- frecvența cu care membrii grupului țintă folosesc resursele aplicației distribuite;
- tipurile de probleme pe care membrii grupului țintă doresc să le rezolve.

Se stabilesc caracteristicile de calitate $Q_1 Q_2 \dots Q_m$ ale aplicației viitoare, prin corespondență cu aplicații similare existente în țară sau în străinătate, și se planifică nivelurile de calitate $q_1^{pl} q_2^{pl} \dots q_m^{pl}$ ale acestor caracteristici, unde m reprezintă numărul caracteristicilor considerate semnificative de către toți utilizatorii existenți, pentru produsele - program aparținând unei clase.

Nivelul caracteristicilor produsului care se va realiza trebuie să îndeplinească condiția

$$q_i^{pl} \geq \max \{ q_{1i}^1; q_{2i}^1 \dots q_{ri}^1 \}; i = 1, 2 \dots m$$

unde r este un număr de produse similare de care are cunoștință cel ce dorește să dezvolte aplicația.

Toate informațiile privind:

- obiectivul pentru care a fost construită aplicația distribuită;
- funcțiile pe care trebuie să le realizeze produsul;
- resursele folosite pentru dezvoltarea etapelor din ciclul de realizare;
- nivelurile planificate ale constantelor de calitate;
- direcțiile necesare traversării etapelor;
- costurile implicate;
- datele de test utilizate pentru a valida produsul,

sunt definite riguros în specificațiile aplicației.

Există specificații, resurse materiale, financiare și umane necesare pentru realizarea aplicației distribuite și implementarea acestora.

Se derulează procesul de realizare a aplicației distribuite.

Auditarea este un proces de comparare a performanței produsului real cu performanțele planificate, existente în specificații.

Rezultatul auditului indică dacă:

- produsul corespunde specificațiilor din punct de vedere calitativ și funcțional;
- produsul nu corespunde și se precizează pentru fiecare diferență dintre produs și specificații care sunt valorile acestor diferențe.

Auditorul trebuie să argumenteze foarte precis decizia de a caracteriza faptul că aplicația distribuită nu satisface cerințele impuse prin specificații.

Componente de produse software sunt auditate, aceasta însemnând că dispun, teoretic și practic, de toate condițiile pentru a realiza aplicații distribuite de calitate. Aceasta nu înseamnă că, în mod obligatoriu, se realizează aplicații de calitate, existând fluctuații în ceea ce privește îndeplinirea tuturor specificațiilor.

Fluctuațiile de calitate sunt în legătură directă cu:

- migrația forței de muncă din companiile IT;
- limitările privind calificarea personalului;
- restricțiile de achiziție de software pentru dezvoltare;
- lipsa unor metodologii și proceduri de măsurare a rezultatelor.

Fluctuațiile devin vizibile când se lansează în execuție curentă produsul. Produsul înregistrează defecte de funcționare precum:

- întreruperi de execuție fără mesaje proprii;
- obținerea de rezultate incomplete;
- obținerea de rezultate incorecte;
- comportament instabil din cauza conținutului zonelor de memorie rămase de la programele executate anterior.

Aplicațiile distribuite trebuie să fie certificate.

Certificarea este un proces prin care o componentă independentă:

- analizează produsul informatic;
- testează comportamentul;
- compară cerințele cu nivelul efectiv;
- evaluează întregul produs;
- stabilește în ce măsură aplicația distribuită îndeplinește cerințele impuse prin specificații.

Certificarea se bazează pe specificațiile aplicației distribuite, dar ține seama de nivelul de performanță a produselor similare de pe piață, produse cunoscute de specialiștii în certificare.

2. Certificarea software

Se realizează de echipe de specialiști.

Necesită o serie de condiții, precum:

- existența textelor sursă;
- documentația produsului să fie disponibilă;
- existența tuturor elementelor necesare lansării în execuție a produsului - program.

Se elaborează un plan de certificare care presupune:

- cunoașterea exactă a centrelor definite de beneficiar;
- cunoașterea specificațiilor;
- însușirea procedurilor după care au fost realizate componentele software.

Se produce o eșalonare a activităților de certificare. Acestea includ:

- măsurarea după metode proprii a performanțelor;
- măsurarea după metodele producătorului;
- stabilirea diferențelor între nivelurile obținute și cele precizate de realizatorii aplicației;
- aprofundarea produsului software;
- întocmirea raportului de certificare din care rezultă fie că produsul software corespunde, fie că nu corespunde, situație în care sunt prezentate recomandări și sunt efectuate modificări.

Certificarea este procesul prin care se realizează un transfer de credibilitate de la organizația care certifică, spre organizația care a realizat produsul.

Certificarea se realizează de către organizații credibile. Credibilitatea este necesară întrucât masa omogenă de utilizatori trebuie să știe că există o formă externă, independentă și imparțială, care verifică concordanța dintre specificații și produsul - software realizat și garantează corespondența dintre specificații și rezultate, asigurând în acest fel că aplicația distribuită este înzestrată cu toate componentele care pentru date de intrare complete și corecte conduce la obținerea de rezultate complete și corecte.

Produsul certificat, fără a se face modificări, răspunde exigențelor grupului țintă.

Procesul de certificare trebuie precedat de instruirea producătorilor asupra exigențelor care apar în procesul de certificare.

În acest fel, crește calitatea produsului și procesul de certificare, prin transparența lui, evidențiază calitățile produsului și mai puțin defectele acestuia.

Există standarde de derulare a procesului de certificare software pe care companiile de consultanță și certificare le aplică. Aceasta înseamnă că dezvoltatorul de software are acces liber la documentația de certificare și creează un nucleu de specialiști care pregătesc produsele software pentru certificare.

Cheltuielile de certificare depind de:

- complexitatea produsului software;
- calitatea documentației oferite;
- nivelul de certificare dorit a se obține;
- nivelul de performanță a produselor de pe piață;
- accesibilitatea la informații privind comportamentul în condiții comparabile dintre produsul dat spre certificare și produsele existente.

În final, dacă produsul software îndeplinește condițiile impuse de procesul de certificare, echipa va elabora un raport favorabil [7].

Rezultă că produsul certificat satisface cerințele utilizatorilor și aceștia îl vor folosi cu încredere dacă și numai dacă sunt respectate condiții de bună funcționare.

Certificarea software implică o serie de aspecte ce țin de normele de ortogonalitate:

- software-ul creat este original, iar aspectele preluate din alte produse - program au acordul titularilor de drepturi;
- datele preluate și prelucrate de aplicația distribuită au redundanță scăzută pentru a nu afecta performanțele produsului program;
- structura bazelor de date are un nivel de ortogonalitate ridicat, câmpurile tabelelor sunt construite fără a se repeta și fără a necesita stocarea datelor în mod repetat;
- programele create pentru gestionarea datelor nu conțin secvențe care se repetă pentru a nu solicita resursele de calcul, aplicațiile eficiente construind biblioteci de programe care sunt apelate în funcție de necesități;

- crearea de software calitativ prin înlăturarea secvențelor redundabile din cadrul programelor; se asigură în acest sens timpi optimi de prelucrare a datelor.

În cadrul aplicațiilor distribuite, ortogonalitatea certifică faptul că software-ul creat este original, că are caracter inovator atât în ceea ce privește soluțiile abordate, cât și cu privire la modalitatea de implementare a acestora. Aplicațiile distribuite trebuie orientate către domenii noi de activitate, care solicită software inovator pentru soluționarea unei problematice distincte și neabordate.

3. Certificarea datelor de intrare

Datele de intrare trebuie să fie complete și corecte [8].

Certificarea produselor software se realizează prin:

- verificarea concordanței dintre nivelurile înregistrate și cele reale;
- existența unor chei de verificare (în elemente din colectivitate și în articole diferite);
- identificarea erorilor din date și a datelor eronate;
- determinarea articolelor duplicate;
- efectuarea de validări asupra datelor preluate de la utilizatori;
- verificarea corectitudinii datelor prelucrate și identificarea situațiilor în care sunt modificate datele inițiale prin:
 - pierderea unei cifre din numere: $1730 \rightarrow 173$;
 - interschimbarea de cifre: $1359 \rightarrow 1539$;
 - pierderea unui număr dintr-un șir:
 în loc de
 170 522 433 615
 avem
 170 433 615
 - introducerea unei cifre în loc de alta: în loc de 7 și se tastează 1.

Erorile tehnice care sunt cunoscute trebuie studiate la momentul producerii acestora pentru a se evita astfel cauzele generatoare.

Certificarea înseamnă:

- proces de analiză a datelor;
- confruntarea datelor din firmă cu cele din documente;
- compararea rezultatelor teoretice cu cele practice, obținute pe baza datelor din cadrul organizației.

Toate acestea conduc, în final, la faptul că între datele de intrare ale problemei de rezolvat și ceea ce se află în firmă și în bazele de date existente trebuie să fie o concordanță perfectă.

Datele de intrare certificate și programele certificate conduc la rezultate de programare valide. Pentru creșterea calității rezultatelor se impune ca și acestea să fie certificate [16].

Mecanismul intermediar dintre datele de intrare și rezultate este aplicația. Construirea unei aplicații calitative și certificate trebuie să țină cont de scopul pentru a fost proiectată și de mijloacele prin care se atinge acest scop.

La proiectarea aplicației, se definesc mecanisme de generare a numelui de fișiere, de colecții, de seturi de date.

În cazul aplicațiilor distribuite, care operează cu concatenări virtuale de baze de date, se definesc sisteme de codificare la nivelul utilizatorilor, iar unde se produce efectiv concatenarea se verifică dacă șirurile de denumiri asigură acoperirea întregului sistem de baze de date. În acest fel, se elimină riscul prelucrărilor incomplete, care impun realocări de resurse speciale pentru elementele neincluse, în cazul în care se impune reluarea procesului de ierarhizare pentru colectivitatea completă, dacă ierarhizarea inițială s-a efectuat numai pentru o parte a colectivității și a fost urmată de alocarea resurselor în mod eronat.

Certificarea datelor de intrare necesită existența unor aplicații orientate spre procesarea datelor în cadrul unui context și identificarea legăturilor și dependențelor dintre date pentru a asigura determinarea datelor care nu respectă condițiile de apartenență la clase de date.

Caracterul redundant al datelor este evidențiat prin implementarea de indicatori software, care certifică

nivelul de repetabilitate al datelor. Impunerea unui prag de redundanță contribuie la îmbunătățirea performanțelor aplicațiilor distribuite prin scăderea timpilor de identificare și procesare a datelor. De asemenea, prin eliminarea redundanței, se asigură actualizarea tuturor datelor în procesul de prelucrare a acestora și evitarea situațiilor în care este tratată doar prima înregistrare din șirul de date care se repetă.

4. Certificarea rezultatelor

Plecând de la date de introdus certificate și de la program certificat, utilizatorii sunt monitorizați pentru a se asigura obținerea rezultatelor dorite. Certificarea rezultatului evidențiază interacțiunea om-calculator.

Certificarea rezultatelor privește modul în care acestea stau la baza fundamentării deciziei.

Certificarea rezultatelor permite utilizarea cu încredere a acestora în construirea de strategii.

Aplicația informatică oferă o structură de rezultate caracterizate prin:

- tipuri de rezultate grupate după gradul de agregare; sunt rezultate sub formă tabelară, care conțin date despre toate elementele unei colectivități; există rezultate care includ selecții după diverse criterii ale elementelor colectivității; există rezultate care includ valori ale indicatorilor agregați;
- domenii de apartenență a valorilor, astfel, pentru unii indicatori valorile trebuie să fie cuprinse între $[0;1]$, pentru alți indicatori valoarea agregată V trebuie să îndeplinească condiția:

$$V < \max_{1 \leq i \leq k} \{x_i\} * 10^h,$$

astfel încât $10^{h-1} < k < 10^h$

Atribute ale calității produselor program:

Testabilitatea este atributul de calitate, care permite utilizatorilor să pună în evidență cât mai multe variante de probleme care sunt rezolvate cu ajutorul produsului - program respectiv. Testabilitatea este evidențiată prin utilizarea exemplului de control existent în documentația care însoțește produsul - program, dar mai ales prin utilizarea părții din documentație în care sunt descrise facilitățile de întreținere a produsului. Testabilitatea presupune întreruperea dirijată a execuției în cazul în care datele problemei activează funcții de prelucrare neadecvate sau sunt depășite valori limită ale parametrilor ce definesc configurația produsului - program. Evaluarea nivelului de testabilitate se realizează fie pe baza structurii produsului - program, fie analizând comportamentul produsului la testările efectuate de beneficiari diferiți. Un produs - program având o robustețe ridicată prezintă, de asemenea, un nivel ridicat al testabilității. Utilizatorii pun în evidență combinații de prelucrări care nu au fost în atenția proiectanților, dar identifică și situații în care produsul nu realizează corect sau complet prelucrări ale datelor descrise conform documentației. Printr-un dialog continuu între utilizatorii și realizatorii produsului - program, se efectuează modificările ce conduc la creșterea nivelului de testabilitate a produselor program.

Completitudinea arată măsura în care toate părțile sunt potențial activabile și fiecare dintre părți realizează funcțiile de prelucrare, date în specificațiile de programare. Stabilirea completitudinii se efectuează prin analiza textului sursă în paralel cu documentația și cu rezultatele obținute prin rularea unui set de date inițiale. Absența completitudinii este semnalată de utilizatori pe măsura imposibilității de a rezolva probleme pentru care documentația nu menționează restricții. Un program pentru care s-a demonstrat corectitudinea este în același timp și un program complet. Gradul de completitudine se exprimă procentual. Completitudinea influențează nivelurile fiabilității și portabilității produselor - program. În cazul realizării pe etape a sistemelor informatice, produsele - program se construiesc sub forma versiunilor. Trecerea la o nouă versiune înseamnă adăugarea unor funcții de prelucrare, existente în documentația inițială, în dreptul cărora se face mențiunea că nu sunt încă implementate. Spre deosebire de adaptabilitate, funcțiile care fac obiectul completării nu prezintă noutate. Ele sunt prevăzute la proiectarea produsului - program, dar realizarea lor este decalată în raport cu momentul intrării în execuție a produsului - program.

Interoperabilitatea unui produs - program arată capacitatea acestuia de a se cupla cu alte produse - program. Interoperabilitatea determină efortul pe care trebuie să-l depună programatorii pentru a face ca un produs - program să devină prin cuplare parte a altui produs - program sau produsul - program să includă ca parte un alt produs - program. Acest atribut permite reutilizarea integrală a unui produs - program pentru realizarea altei aplicații complexe. Nivelul interoperabilității este influențat de gradul de modularitate cu care este conceput produsul - program. Cuplarea produselor - program se efectuează direct sau folosind interfețe. În ambele cazuri, transmiterea parametrilor respectă cerințele privind standardele parametrilor. În cadrul aplicațiilor distribuite, modulele componente stau la baza prelucrării parțiale a datelor [11]. Asigurarea

interoperabilității modulelor conduce la asigurarea unui caracter unitar al aplicației.

Generalitatea este atributul prin care se pune în evidență aria de cuprindere a funcțiilor de prelucrare, variantele problemei ce sunt rezolvate, cazurile particulare și dimensiunile maxime ce sunt luate în considerare. Generalitatea aplicației este înfățișată utilizatorilor prin multitudinea facilităților pe care le conține explicit documentația care însoțește produsul - program. Creșterea gradului de generalitate a produsului - program influențează negativ eficacitatea și fiabilitatea acestuia. Produsele - program având un grad de generalitate ridicat trebuie să aibă și un nivel ridicat al stabilității. Creșterea gradului de generalitate este însoțită de creșterea gradului de complexitate pentru produsul - program considerat.

Consistența unui produs - program arată măsura în care componentele sale realizează funcții de prelucrare necontradictorii și se bazează pe uniformizare în folosirea simbolurilor și a regulilor de construire a identificatorilor, etichetelor și în general a secvențelor omogene. Consistența internă a produsului - program este pusă în evidență prin analiza textului sursă, iar evaluarea ei se poate face automat. Consistența externă este dată de comportamentul necontradictoriu al produsului. Evaluarea consistenței externe se efectuează după terminarea testării sau după ce s-a efectuat verificarea corectitudinii programului. Programul pentru care a fost stabilită caracteristica de corectitudine este în același timp și un program consistent. Consistența determină influența asupra nivelurilor de fiabilitate și mentenabilitate. Creșterea consistenței se realizează prin introducerea de standarde în construirea modulelor și prin utilizarea de tehnici de programare evaluate.

Complexitatea unui program se determină prin asocierea unui graf, în care nodurile corespund instrucțiunilor, iar arcele indică ordinea de execuție a acestora. Acest graf determină structura programului. Complexitatea este un atribut ce permite atât stabilirea diferențelor dintre structurile programelor, cât și ierarhizarea programelor după structura fiecărui program. Complexitatea programelor poate fi determinată dacă se analizează frecvențele de apariție a instrucțiunilor în programe. Evaluarea complexității produsului - program se realizează fie pornind de la textul sursă al programului, fie pornind de la graficul asociat lui. Estimarea complexității produsului - program permite planificarea necesarului de resurse pentru realizarea componentelor sale, stimulându-se realizarea unor produse - program complexe, echilibrate din punct de vedere al dimensiunilor și al funcțiilor de prelucrare. Nivelul complexității influențează negativ fiabilitatea, mentenabilitatea și stabilitatea produselor - program.

Flexibilitatea unei aplicații arată în ce măsură aceasta implică un număr cât mai redus de restricții la utilizator. Restricțiile se referă la modul de descriere a datelor pentru problemele care se rezolvă și la modul în care apar rezultatele finale. Flexibilitatea este dependentă de volumul opțiunilor care stau la dispoziția utilizatorilor, de multitudinea de combinații care se pot genera prin utilizarea parametrilor. Flexibilitatea este pusă în legătură și cu posibilitatea de obținere a rezultatelor complete, chiar dacă nu sunt corecte, atunci când datele inițiale conțin erori. Programele cu un nivel al fiabilității ridicat sunt în același timp și programe flexibile. Flexibilitatea influențează nivelul complexității programelor. Programele cu un nivel de modularitate ridicat au nivelul flexibilității de asemenea ridicat. Flexibilitatea este privită, din punct de vedere al programatorilor, ca fiind posibilitatea programelor de a prelua cerințe noi de prelucrare ce nu au fost prevăzute. Programele flexibile necesită pentru realizarea adaptărilor eforturi minime. Noile funcții de prelucrare corespund unor valori noi pentru anumiți parametri sau dezvoltări ce aparțin aceluiași clase de comenzi pentru care există deja elaborate componente program.

Structurabilitatea indică modalitatea de grupare a părțile omogene sau care realizează o funcție de prelucrare distinctă în secvențe, subprograme, secțiuni, module. Structurabilitatea aplicațiilor distribuite este un atribut prin care entităților din produsul respectiv li se asigură un caracter distinct. Numărul de niveluri pe care se efectuează descompunerea trebuie să fie optim. Structurabilitatea permite identificarea pașilor algoritmului, precum și identificarea structurilor fundamentale din cadrul programelor, dar și între programe. Programul va apărea numai ca o asamblare de structuri fundamentale. Produsul - program va apărea tot ca o asamblare de componente distincte - module. Structurabilitatea presupune stabilirea poziției fiecărei componente care alcătuiește produsul - program și a legăturilor ei cu celelalte componente. Cuantificarea structurabilității are la bază numărul nivelurilor pe care se dispun componentele și numărul legăturilor dintre componente. Structurabilitatea influențează modularitatea produselor - program.

Modularitatea definește utilizarea modulelor în cadrul aplicațiilor. Modulul reprezintă acea componentă care realizează o funcție de prelucrare completă. Atributul de modularitate al aplicației este dat de capacitatea acesteia de a fi descompusă în module și de a stabili legăturile dintre acestea. Structurabilitatea este atributul care dă ordinea în cadrul modulului. Modularitatea este atributul care descrie dispunerea modulelor în cadrul produsului. O aplicație are un nivel de modularitate ridicat în cazul în care identificarea funcțiilor de prelucrare după reguli standard necesită un efort minim. Nivelul modularității este stabilit încă din faza de proiectare a aplicațiilor. Evaluarea acestui nivel are la bază textul sursă al aplicației. Modularitatea influențează toate

caracteristicile de calitate și este influențată de complexitatea aplicațiilor.

5. Concluzii

Certificarea este necesară pentru toate aplicațiile care presupun întreținere on-line. Ele asigură și garantează calitatea prelucrărilor.

Numai aplicațiile informatice foarte sigure trebuie să facă obiectul certificării.

Orice modificare în cadrul aplicației, în date și în modalitate de implementare a cerințelor, atrage după sine reluarea procesului de certificare.

Domeniul calității produselor - program este abordat cu preponderență în mediile de cercetare și de către firmele mari, producătoare de software. Cercetările se află într-un stadiu avansat și s-au obținut rezultate deosebite, dar uneori așteptările nu au fost confirmate decât într-o foarte mică măsură. Dacă inițial marea masă a producătorilor și utilizatorilor de produse - program nu era convinsă de necesitatea și utilitatea specificării și evaluării calității produselor - program, ulterior, s-a conștientizat importanța acestui domeniu.

Pe plan intern, s-au desfășurat cercetări în acest domeniu începând cu anul 1981 și au fost subordonate preocupărilor de introducere a metodelor și procedurilor pentru asigurarea și controlul calității produselor - program. Principalele preocupări din domeniul calității produselor - program au fost orientate în următoarele direcții:

- definirea metodologiei pentru specificarea și determinarea calității produselor - program pe întreg ciclul de viață;
- elaborarea metodelor cantitative și a instrumentelor de măsurare și evaluare a calității produselor - program;
- elaborarea nomenclatorului și a clasificării indicatorilor calitativi pentru produsele - program;
- definirea și implementarea metodelor statistico-matematice în validarea indicatorilor de calitate;
- experimentarea indicatorilor de calitate în cadrul proiectelor concrete, cu accent deosebit pe caracteristicile de fiabilitate și mentenabilitate;
- cercetarea posibilităților de adaptare și implementare a seriei de standarde ISO 9000 la aplicațiile distribuite orientate pe alocarea de resurse pe bază de abonament atât pe parcursul procesului de realizare, cât și în faza de recepție, testele de acceptare.

Prelucrările moderne presupun existența de baze de date, definite de fiecare utilizator, și se execută pe un server central al aplicației distribuite cu acces pe bază de abonament.

În acest fel, se asigură că toate rezultatele prelucrărilor oferite de utilizatori ca date de intrare pentru un proces de agregare sunt obținute folosind același algoritm, ceea ce le asigură comparabilitatea și omogenitatea.

Analiza laturilor calitative ale activității de programare se reflectă prin realizarea de produse - program având performanțe ridicate. Posibilitatea de urmărire a modului cum se realizează din punct de vedere calitativ produsele - program depinde de culegerea automată a datelor privind produsele respective, privind comportamentul acestora de-a lungul tuturor etapelor ce alcătuiesc ciclul de viață și evaluarea, de asemenea, automată a indicatorilor pentru nivelurile caracteristicilor și atributelor de calitate.

Introducerea în cadrul procesului de certificare a aspectelor ce țin de ortogonalitatea aplicației contribuie la creșterea calității acesteia prin asigurarea caracterului de originalitate, prin dezvoltarea de software dedicat problemelor ce trebuie soluționate, și nu preluat din cadrul altor aplicații și ulterior adaptat, prin construirea de biblioteci dedicate aplicației distribuite și prin îmbunătățirea proceselor de validare a datelor în diverse etape de procesare: la preluare, la actualizare, la procesare și la afișare.

Bibliografie

1. **BALOG AL.:** Evaluarea proceselor software, Editura I.C.I., București, 1999.
2. **BRAKHAHN, W., U. VOGT:** ISO 9000 pentru servicii, Editura Tehnică, București 1998.
3. **BURAGA, S. C.:** Tehnologii Web, Editura Marix Rom, București 2000
4. **COSTACHE D., G. AVRAM, M. MOISE, E. SONEA ș.a.:** Manualul calității AISTEDA, Editura AISTEDA, București, 1999.
5. **FRENȚIU, M., B. PÂRV:** Elaborarea programelor. Metode și tehnici moderne, Editura Promedia, Cluj, 1994.

6. **GHILIC-MICU, B.:** Aspecte ale impactului trecerii la societatea informațională. În: Informatica economică, editată de Catedra de Informatică Economică și asociația INFOREC, cu sprijinul Ministerului Educației și Cercetării, volumul V, nr. 3 (19)/2001, pp. 10 – 15.
7. **IVAN, I., P. POCATILU:** Testarea software orientat obiect, Editura Infosec, București 1999.
8. **IVAN, I., G. NOȘCA, S. TCACIUC, O. PÂRLOG, R. CĂCIULĂ:** Calitatea datelor, Editura Infosec, București 1999.
9. **IVAN, I., L. TEODORESCU:** Managementul calității software, Editura Infosec, București 2001.
10. **IVAN, I., P. POCATILU, M. AMITROAIE:** Metrici ale societății informaționale, Informatica economică, editată de Catedra de Informatică Economică și asociația INFOREC, cu sprijinul Ministerului Educației și Cercetării volumul V, Nr. 4 (20)/2001, pp. 33 – 40.
11. **IVAN, I., D. MILODIN, L. SĂCUIU:** Calitatea datelor de intrare în aplicații distribuite. În: Revista Română de Informatică și Automatică, vol. 17, nr. 4, 2007, pp. 35 – 48.
12. **MEYER, B.:** Object Oriented Software Construction, Editura Prentice Hall, Englewood Cliffs, 1988.
13. **MCCALL, J.:** Factors in Software Quality, Editura General Electric Co., 1997.
14. **MOTOIU, R.:** Ghid pentru managementul calității și calitatea managementului anilor 2000, Editura FiaTest, București, 2000.
15. **SOARE, E., AL. D. COLCERU:** Certificare calității, Editura Tribuna Economică, București 1996.
16. **VASILESCU, G., A. BALOG:** Sistemul caracteristicilor de calitate ale produselor program - referat, Academia de Studii Economice, Facultatea de Cibernetică, Statistică și Informatică Economică, București, noiembrie, 1992.