

Turn-Key Software Programs

Prof.dr. Ion IVAN, prep. Cătălin BOJA, stud. Cosmin TOMOZEI
Catedra de Informatică Economică, A.S.E. București

There is described the concept of program products generations in the context of current programming techniques development. There are detailed the contents of turn-key software products. There are presented ways and methods to develop software that would be a turn-key product. For existing software programs are detailed the requirements imposed by the reengineering techniques in order to obtain fully functional, turn-key applications.

Keywords: software, quality characteristics, reengineering, web applications, turn-key.

Generații de produse program

Dezvoltării computerului îi corespunde o dezvoltare a software-ului. Generațiile de hardware au permis dezvoltarea unor generații de software. Ritmurile de dezvoltare ale celor două, hardware, respectiv, software au fost diferite. Întotdeauna, tehnicile și metodele de programare au devansat resursele hardware, iar acestea din urmă au temperat complexitatea limbajelor de programare, uneori frânând-o.

Programarea clasică, fără reguli precise a condus la realizarea de produse program performante, ale căror interfețe au avut rolul de a limita accesul persoanelor fără pregătire în informatică la lansarea în execuție, la introducerea de date și la definirea de opțiuni. Programele stocate pe cartele perforate, pe bandă magnetică sau pe discuri fixe au necesitat prezența autorului lor în cazul fiecărei rulări.

Tehnica programării structurate a corespuns nevoii de a crește productivitatea programatorilor prin introducerea unor reguli. Au fost create premisele dezvoltării simultane a modulelor având același nivel de performanță. Prin creșterea omogenității programelor s-a obținut un control bine definit prin utilizarea structurilor de control.

Ultimele generații de limbaje de programare sunt stocate în biblioteci de funcții puternice care permit accesarea tuturor resurselor hardware, dintre care cele de grafica și de alocare dinamica a memoriei sunt esențiale. Abordarea obiectuală, bază a maximizării nivelului de reutilizare a construcțiilor deja verificate ca fiind valabile, a reprezentat ele-

mentul forte în dezvoltarea de interfețe, implementarea produselor de validare și în dezvoltarea de mecanisme pentru stocarea și regăsirea de informații, deosebit de puternice.

O dată create condițiile, s-a produs și o creștere accelerată a utilizatorilor de software. Orientarea produselor program spre utilizatori neinformaticieni impune o nouă abordare a codului de dezvoltare.

Un rol special l-au jucat pe de o parte scăderea dramatică a costurilor și dimensiunilor calculatoarelor și pe de altă parte, creșterea performanțelor. S-a ajuns ca în numeroase familii cu venituri în jurul mediei pe economie să existe câte un calculator cu acces la resursele Internet.

În acest context, producția de software a trebuit revoluționată în direcțiile următoare:

- realizarea de aplicații informatice de largă utilizare, care prin numărul de utilizatori să permită recuperarea cheltuielilor generate de ciclul de dezvoltare;
- asigurarea calității produselor software astfel încât acestea să conducă la obținerea de rezultate complete și corecte atunci când datele de intrare îndeplinesc condițiile cerute;
- funcționarea programelor prin operații simple de încărcare, instalare, fără a necesita existența unui programator sau a unui depanator la dispoziția fiecărui utilizator;
- obținerea produsului software de pe piața de produse finite la prețuri accesibile, însoțite de o documentație care prezintă preponderent funcționalitatea și mult mai puțin lista de erori, cauze și modele de eliminare.

Evoluțiile din IT au impus o adaptare a producției de software spre tipologia industrială

concretizată prin:

- procese bine definite, în care etapele se definesc prin intrări, ieșiri, operații, termene și restricții de succesiune;
- planificarea duratelor, consumurilor și calității fiecărui produs intermediar dar și a produsului finit în totalitatea lui;
- controlul riscurilor privind obținerea diferențelor peste anumite limite între procentul definit inițial și produsul finit care face obiectul implementării la beneficiari;
- procese complexe de managementul proiectelor care planifică și monitorizează toate aspectele legate de ciclul de viață al produsului;
- controlul strict al calității aplicației software bazat pe procese riguroase de testare și analiză a cerințelor utilizatorilor.

În aceste condiții s-a impus definirea conceptului de software la cheie, care din punct de vedere al utilizatorilor este o cutie neagră, care funcționează independent de cei care l-au produs și livrat.

Caracteristici de calitate software

Nivelul de calitate al a produselor software se determină pe baza nivelelor măsurate ale caracteristicilor de calitate. Pe lângă sistemele de caracteristici software, dintre care cel mai cunoscut este ISO 9126, aplicațiile informatice care permit realizarea produselor la cheie, impun acordarea unui grad de importanță mai ridicat unui set de caracteristici descris în figura 1.

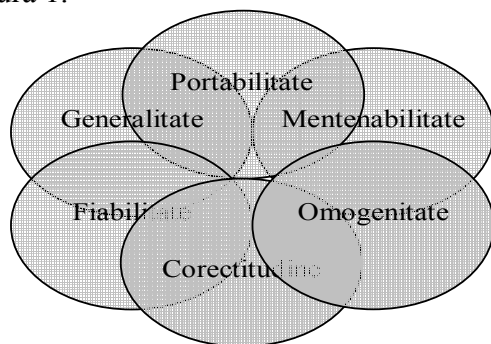


Fig. 1. Caracteristici software

Generalitatea este o caracteristică importantă cu care orice produs software la cheie trebuie să fie înzestrat. De la utilizator la utilizator există diferențe privind datele de intrare, volum, formă de prezentare și domeniu de variație. Rezultatele finale din punct de vedere al

corectitudinii nu diferă de la utilizator la utilizator. Diferă forma de prezentare și existența unor indicatori pe care programul oricum îi calculează.

Generalizarea unui produs software vizează:

- opțiuni privind modul de stocare a datelor de intrare și prelucrarea acestora;
- închiderea unor proceduri care largesc aria de prelucrare incluzând tratarea cazurilor particulare, tratarea unor noi aspecte legate de dimensiunea problemei, de modificări ale structurii datelor de definire precum și aspecte privind înlocuirea unor constante, cu parametri, respectiv, eliminarea ipotezelor de liniaritate și includerea unor noi tipuri de inecuații;
- elemente prin care se definește nivelul de precizie adoptat ca fiind acceptabil
- luarea în considerare a datelor cantitative dar și a atributelor, astfel încât să fie preluate serii de date rezultate din măsurători, cât și serii de date obținute din aprecieri;
- specificarea trecerii de la k dimensiuni la $k+1$ trepte de date pentru care sunt indicate modalități de introducere a informațiilor așa cum le are utilizatorul;
- definirea unui cadru de prezentare a informației care să nu fie dependent de natura și numărul acestora.

Fiabilitatea care vizează raportul dintre rulările de succes față de total rulări. Un produs software care este livrat la cheie trebuie să fie suficient de bine testat, încât să fie eliminate toate situațiile în care se produc întreruperi accidentale ale execuției. Un produs fiabil este acela care conduce în peste 78% din cazuri la rezultate complete și corecte.

Tehnicile de programare moderne includ numeroase modalități de creștere a nivelului fiabilității software.

Definirea de clase adecvate, în care alocările de zone de memorie sunt supuse testelor menționându-se concordanța între ceea ce s-a definit și ceea ce se inițializează, are ca efect obținerea de programe fiabile.

Testarea nivelului de fiabilitate al aplicației se realizează prin utilizarea de seturi de date de intrare cât mai variate, atât ca natură cât și ca număr. Datorită gradului ridicat de generalitate al aplicației, trebuie analizată gradul în

care natura datelor de intrare influențează nivelul de erori de execuție. Se identifică situațiile particulare care conduc la scăderea nivelului fiabilității.

Corectitudinea este o caracteristică de calitate esențială pentru a obține procese de producție ale căror rezultate produse program la cheie.

Corectitudinea este evidențiată printr-un proces deosebit de complex de testare software. La elaborarea produselor program se definesc specificații în mod riguros, ceea ce conduce la simplificarea operațiilor de elaborare de cod. Textele sursă prezentate pe baza unor specificații bine gândite conduce după testare la forme finale de produse software cu riscuri controlate de obținere a rezultatelor eronate. Această caracteristică este direct influențată de gradul de generalitate al diferiților algoritmi și metode de prelucrare a datelor. Nivelul general al corectitudinii aplicației este compus din nivelele intermediare ce privesc procesele de preluare a datelor de intrare, de prelucrare și de obținere a rezultatelor finale. Din punct de vedere al datelor, se obține un grad ridicat de corectitudine dacă procesele interne aplicației nu afectează valoarea sau semnificația acestora.

Mentenabilitatea este capacitatea unui produs software de a fi actualizat cu eforturi cât mai scăzute. Actualizarea produsului software constă în:

- modificarea structurilor de tip articol prin includerea de noi câmpuri în cazul în care algoritmi de prelucrare vor conține în formulele de calcul noi variabile;
- introducerea de noi secvențe de prelucrare pentru a obține conținutul impus de cerințele formulate pentru obținerea de noi indicatori, de noi structuri de rapoarte;
- modificarea structurii rapoartelor prin adăugarea de coloane sau de linii, pentru a prelua rezultatele obținute de secvențele de calcul adăugate în produsul software.

Mentenanta presupune dezactivări permanente sau temporare a unor câmpuri din baza de date, a unor secvențe de prelucrare și a unor secvențe de definire a rapoartelor, respectiv, de inițializare a câmpurilor acestora.

Un produs software este construit în ideea de

a fi mentenabil dacă:

- modulele au un grad de specializare deosebit de ridicat;
- se izolează modulele care sunt afectate curent de procesul de mentenanță, de celelalte module;
- se externalizează părți de prelucrare prin introducerea de parametri a căror inițializare generează obținerea tuturor varietăților de structuri de expresii analitice.

Portabilitatea reprezintă un criteriu important pentru nivelul general de calitate al unui produs software. Caracteristica vizează capacitatea aplicației de a fi implementată pe diferite platforme hardware și software fără a necesita modificări substanțiale în cod și fără a fi afectată execuția.

Nivelul portabilității depinde de gradul de generalitate al șablonului pe baza căruia sunt realizate versiunile particulare. Cu cât generalitatea este mai mare, aplicația are implementate :

- rutine pentru identificarea altor aplicații software cu care se interacționează: sistem de operare, browsere în cazul aplicațiilor web, interfețe de comunicare instalate; aceste elemente sunt introduse în codul sursă în modulele de bază și sunt inițializate fie prin fișiere de configurare fie prin secvențe executate în faza de preprocesare;
- componente variate care să-i permită executarea corectă a prelucrărilor în funcție de setările utilizatorului;
- interfețe care să-i permită utilizarea componentelor existente pe platforma hardware și software;
- rutine de încărcare automată, de pe rețea sau de pe disk, a componentelor necesare.

Omogenitatea modului de interacțiune a utilizatorului cu aplicația urmărește implementarea cu preponderență a acelorași metode de selecție, vizualizare și introducere a datelor în toate componentele aplicației. De exemplu, alegerea criteriului de selecție a informației se face numai prin implementarea de combobox-uri, sau introducerea datelor se realizează utilizând câmpuri de tip text editabile. Un nivel ridicat al omogenității influențează pozitiv gradul de utilizabilitate al aplicației, deoarece utilizatorul folosește același

mod de lucru în toate componentele aplicației. De asemenea, el acordă o atenție mai mare calității informației, decât să se familiarizeze cu componente noi. Pentru atingerea acestui obiectiv, producătorul trebuie să ia în calcul și un grad ridicat pentru omogenitatea componentelor implementate.

Lansarea aplicației în producție presupune indicarea clară a specificațiilor de instalare și rulare astfel încât să elimine situațiile în care producătorul este obligat să suporte costurile de modificare a produsului astfel încât să corespundă cerințelor utilizatorului.

Și alte caracteristici precum integrabilitate sau caracterul prietenos sunt deosebit de importante. Producătorii de software la cheie dezvoltă proceduri speciale pentru a atinge nivelurile planificate ale acestora.

Proprietățile produselor software la cheie

În primul rând produsul software la cheie este o construcție de sine-stătătoare. În cazul în care are o structură modulară, se lucrează sub forma unui set de CD-uri sau de dischete și care are un regim special de încărcare în vederea execuției. Dacă produsul software este un executabil se indică mecanismul de utilizare a resurselor hardware și software pentru a dezvolta procese de execuție fără incidente. Este preferabil ca produsul software la cheie să fie instalat fără a fi necesare resurse speciale în afara unei configurații, frecvent întâlnită la posesorii de computere.

În al doilea rând, produsul software la cheie trebuie să funcționeze independent cel care l-a creat. Fie pe o etichetă, fie sub forma unei broșuri, elaboratorul de software explică pașii pe care utilizatorul trebuie să-i întreprindă în vederea obținerii rezultatelor pe care le așteaptă. Produsul software este însoțit de o documentație simplă, a cărei parcurgere asigură înțelegerea modului în care utilizatorul gestionează lansarea în execuție, introducerea de date și obținerea unei varietăți de rezultate finale.

În al treilea rând, produsul software la cheie trebuie să conțină mesaje proprii pentru toate situațiile de neconcordanțe între ceea ce execută programul și ceea ce dorește utilizatorul. În acest fel sunt create premisele pentru atenționarea utilizatorilor în vederea introducerii

de date corecte. Sunt reluate unele secvențe de inițializare și se oferă indicații și exemple de cum trebuie să arate datele de intrare pentru a fi acceptate.

Interfețele adaptabile au menirea de a prezenta gradual tipologii de mesaje. Pentru începători sunt date mesaje foarte detaliate, iar pentru avansați, în vederea creșterii vitezei de interacțiune, se dau mesaje scurte sau se grupează datele de intrare în blocuri, iar blocurile li se asociază un singur mesaj.

În al patrulea rând, produsul software la cheie conține numeroase elemente care se întâlnesc și la alte produse cu frecvență de utilizare foarte mare, pentru a asigura continuitatea în exploatare celor familiarizați cu anumite opțiuni cu o anumită semnificație a cuvintelor cheie asociate comenzilor.

În al cincilea rând, forma de prezentare a produsului software impune restricții severe în ceea ce privește depanarea de către utilizatori. Dimensiunea produsului, referirile din bibliotecile de proceduri, utilizarea de medii al căror produs este text generat, creează condiții de a-l izola pe utilizator de procesul de efectuare a corecțiilor și de procesul de mentenanță.

În al șaselea rând, produsul software la cheie este un produs care a traversat etapele de proces cu caracter individual în care, la fiecare post de lucru se produc testarea calității, controlul, predarea, respectiv, preluarea fiecărui stadiu.

În al șaptelea rând produsul software la cheie este rezultatul activității unei echipe complexe care prin activitatea desfășurată măsoară calitatea, efectuează corecții pentru diminuarea noncalității. Sunt aplicate regulile managementului calității software. Testarea software are menirea de a construi toate seturile de date necesare implementării metricilor software.

Aplicații WEB la cheie

În contextul accesului la resursele Internet cu costuri suportabile, este important ca produsele software să capete o mai largă răspândire. Acest deziderat se obține prin dezvoltarea de aplicații Web din care se lansează programe și se interoghează baze de date. Aplicațiile Web la cheie trebuie proiectate pentru

a răspunde cerințelor grupului țintă căruia i se adresează. Grupul țintă este format din persoane cu vârste, calificări și nivel de cunoștințe diferite. Când se proiectează o aplicație este important ca ea să satisfacă cerințele grupului țintă.

Dacă am defini un indicator de apreciere al gradului de satisfacție a utilizatorilor ca raport între numărul de accesări finalizate cu succes și numărul total al accesărilor, se va considera ca o aplicație Web este la cheie dacă peste 92% dintre utilizatori sunt satisfăcuți. Aplicația trebuie proiectată de la început în această direcție. Ea trebuie să îndeplinească următoarele cerințe:

- mod omogen de alegere a opțiunilor de traversare a structurii asociate fluxurilor ce definesc legăturile;
- existența concordanței între ceea ce dorește utilizatorul și ceea ce oferă aplicația Web; se efectuează studii statistice pentru a evidenția frecvențele de apariție a cuvintelor cheie și cele care au frecvențele cele mai mari se depun în pagina de start a aplicației.

Aplicațiile Web la cheie presupun respectarea unor reguli precise privind următoarele aspecte :

- corectitudinea datelor afișate pentru a declanșa acțiuni ale celor ce accesează resursele și pentru a finaliza aceste acțiuni;
- obligativitatea de a defini durata de viață a unei informații, pentru a include în aplicație strict numai ceea ce este în actualitate, eliminând informațiile expirate;
- crearea de fluxuri care actualizează automat baza de date încât aceasta să reflecte în timp real evoluția fenomenelor;
- stocarea informațiilor și resursele oferite pentru a da posibilitatea utilizatorilor să-și soluționeze problemele proprii, inclusiv să manipuleze fragmente de fișiere.

Aplicațiile Web la cheie se proiectează astfel încât prin introducerea unor date sau prin selectarea de opțiuni să se producă o diminuare a listelor pe care utilizatorul le prelucrează în continuare.

Problematika optimizării aplicațiilor Web vizează minimizarea numărului de acțiuni de acces întrerupte din cauza modului în care utilizatorul își definește datele de intrare, prin

faptul ca a efectuat introduceri eronate și nu există modalități de corectare a lor.

În proiectarea de aplicații web există reguli precise, există o experiență acumulată, de aceea este foarte important, pentru a asigura succesul unei noi aplicații web să se respecte regulile de bază și chiar să fie preluate unele soluții valoroase în ceea ce privește definirea datelor de intrare. Cu cât se reduce ponderea informațiilor culese de la tastatură, cu atât se obțin șanse de acceptare a aplicației web de către un număr mai mare de utilizatori.

Dacă sunt definite modalități de stocare a informațiilor de identificare, prin introducerea lor o singură dată, folosind un formular, sunt create premisele reale pentru a facilita realizarea de fluxuri complete de atribuire de resurse și de creare a obligațiilor ce decurg din alocările efectuate sub controlul celor care au accesat resurse și s-au autentificat;

Dezvoltarea aplicațiilor web la cheie presupune un nou mod de gândire adaptat la exigențele unor utilizatori neomogeni.

O parte însemnată din aplicațiile Web la cheie au rolul de magazine virtuale, realizând operațiuni de comerț electronic. Aplicația va conține în mod obligatoriu o bază de date cu produsele pe care le oferă consumatorilor sau serviciile prestate precum și coșul de cumpărături cu ajutorul căruia vizitatorul selectează produsele dorite. Pentru livrarea *la cheie* a unei aplicații de comerț electronic într-un timp acceptabil din punct de vedere al beneficiarului și pentru a se evita repetarea unor erori anterioare, se constituie biblioteci de funcții/proceduri și structuri de date reutilizabile.

Reutilizabilitatea este bazată în primul rând pe generalitate. În acest mod, pentru realizarea unei aplicații de comerț electronic nouă, se utilizează biblioteci deja existente la care se adaugă noi funcționalități, în funcție de specificul aplicației curente.

Nivelul înalt de generalitate al fondului de programe existent va oferi dezvoltătorului de software posibilitatea realizării unui nou proiect cu un efort minim în procesul de actualizare. Valorificarea fondului de date și programe necesită o verificare/expertiză

prealabilă a calității acestuia. Lipsa unei verificări corespunzătoare a calității poate produce daune importante beneficiarului, dar mai ales celui care a conceput și creat produsul software. Vânzarea unui produs/serviciu atrage după sine răspunderea juridică a celui ce l-a realizat/prestat.

Înlăturarea unor astfel de neajunsuri se realizează prin auditul informatic. Acordul auditorului denotă un nivel al calității produsului software conform cu standardele acceptate prin reglementările în vigoare.

Metoda cea mai frecvent utilizată de experți pentru a valida calitatea produsului este cea a utilizării datelor de test, urmărindu-se în mod deosebit fiabilitatea produsului și corectitudinea rezultatelor obținute. Auditorul utilizează propriile date de test pentru a verifica modul în care acestea sunt prelucrate de către aplicație. Seturile de date se elaborează în funcție de specificul aplicației și de gradul de complexitate estimat.

În cazul în care vor fi observate neconcordanțe între cerințele și rezultatele proiectului se întreprind modificări, corecții pentru înlăturarea acestora. Constatarea bunei funcționări de către experți va da unda verde comercializării produsului.

Reingineria aplicațiilor de comerț electronic

Procesul de reinginerie software urmărește utilizarea de produse software existente și care au fost testate de practică ca bază de start în dezvoltarea de noi aplicații software. Acestea includ module și componente preluate de la produse anterioare și care sunt supuse unui număr mic de modificări pentru a fi adaptate noului context.

Aplicațiile de comerț electronic includ modele economice și financiare ce definesc structura procesului și totodată diagrama componentelor produsului. Asigurarea unui proces de analiză și proiectare care să identifice factorii importanți și variațiile acestora conduce la realizarea unei aplicații de comerț electronic cu grad ridicat de mentenabilitate și modularitate. Dacă se ia în considerare și caracterul de stabilitate al procesului economic pe perioade lungi de timp, aplicația inițială reprezintă o sursă importantă pentru activită-

țile de reinginerie software.

Dacă se dorește realizarea unui magazin virtual având ca scop comercializarea unor articole de îmbrăcăminte, iar proiectantul aplicației posedă deja o bibliotecă de aplicații de comerț electronic având ca obiect comercializarea altor categorii de produse, acestea se vor dovedi utile în procesul de reinginerie.

Acesta înglobează totalitatea tehnicilor, metodelor, procedurilor și funcțiilor necesare obținerii unei noi aplicații în vederea realizării unui obiectiv diferit de cele anterioare.

Procesul de reinginerie este determinat de existența unei aplicații care este supusă unui lanț de transformări, fiecare dintre ele determinând un stadiu de dezvoltare al proiectului. S-a demonstrat că obținerea unui produs prin reinginerie este mult mai eficientă din punct de vedere al utilizării resurselor decât reluarea întregului proces de realizare.

Lanțul de transformări la care este supusă o aplicație proiectată pentru vânzarea biletelor în stațiuni turistice pentru a se obține magazin virtual de cămăși poate fi următorul:

t_1 – înlocuirea colecției de existente cu imaginile noilor produse oferite consumatorilor;

t_2 – schimbarea conținuturilor etichetelor și mesajelor adresate utilizatorilor;

t_3 – alegerea unui nou titlu pentru fiecare modul al proiectului;

t_4 – operarea de modificări în baza de date;

Colecția de fotografii reprezentând peisaje din stațiunile turistice va fi înlocuită de fotografii ce reprezintă produsele oferite spre vânzare, cămăși. Pe structura existentă, fiecare mesaj, etichetă a interfeței web vor fi reanalizate, păstrate dacă este cazul, modificate sau șterse.

Modificarea bazei de date se realizează atât la nivelul denumirii coloanelor din tabele și la nivelul legăturilor. Dacă tabela care se referă la biletul oferit clientului avea denumirea *bilet*, această denumire ar fi trebuit modificată. Printr-un grad înalt de generalitate însă, dacă s-ar fi numit *produs*, redenumirea acesteia nu ar mai fi fost necesară. Se întreprind de asemenea adăugări, modificări, ștergeri de coloane.

Realizarea codului din pagina HTML trebuie

să aibă caracter cât mai general, fiind orientată pe generarea dinamică a informațiilor din pagină în funcție de datele afișate. Figura 2 descrie o secvență dintr-o pagină web ce conține instrucțiuni scrise în ASP și care permite

afișarea datelor dintr-o tabelă a unei baze de date sub forma unui tabel. Afișarea datelor se face automat, fiind inițializat titlul paginii și numele câmpurilor din tabel.

```
<body bgcolor="#FFFFFF">
<h2 align="center" class="font-large"> Detele cu privire la "<% = Request.QueryString ("Categorie") %>" </h2>
<% -- realizare conexiune baza de date -- %>
<table width="100%" height="100" border="0" cellpadding="0" cellspacing="0">
<tr> <td width="77" align="left" valign="top">&nbsp;</td> <td align="center">
<%contor=0%>
<table width="90%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td align="center"> <b><font class="font-small"><% = fldF.Name %></font></b></td>
<%Next%>
</tr>
<%While Not Record.EOF%>
<tr>
<%For Each fldF in Record.Fields
'Response.Write "<option value=""& fldF.Name &"">"&fldF.Name&"</option>"%>
<td align="center"> <font class="font-small">
<% Response.Write Record.Fields (fldF.Name) %></font>
</td>
<%Next%>
</tr>
<% contor = contor+1
Record.MoveNext
Wend %>
</table></td><td width="77"> </td></tr></table>
</body>
```

Fig. 2. Afișare dinamică într-o pagină WEB a datelor dintr-o bază de date.

Se are în vedere conservarea pe cât posibil a sistemului de chei, care asigura legăturile dintre tabelele bazei de date. Prin urmare re-denunirile, ștergerile de attribute cheie trebuie sa se facă cu o deosebita precauție.

Una dintre cele mai importante componente ale aplicației de comerț electronic este cea care vizează sistemul de încasări și plăți. Dacă acest subansamblu al aplicației a fost audiat în prealabil, iar experții care au testat sistemul au manifestat fără rezerve acordul în ceea ce privește protecția datelor, securitatea informațiilor despre contul clientului, număr de card și realizarea corectă a plăților, atunci sistemul de plată electronică trebuie menținut în totalitate, fără a se mai efectua nici un fel de modificare.

Concluzii

Aplicațiile software la cheie se proiectează astfel încât să evite:

- introducerea de la tastatură a acelorași date în mod repetat;
- utilizarea cheilor de identificare a altelor

decât codul numeric personal sau numele persoanei care accesează aplicația;

- absența unor validări ale câmpurilor , CNP, adresa, e-mail, prețuri, cantități;
- absența ierarhizării unor criterii de selecție care să elimine din aproape în aproape elemente ale colectivităților;
- introducerea unui număr exagerat de mare de date;
- absența omogenității modulului de selectare a opțiunilor;
- lipsa de a prelua informații o dată introduse, care trebuie stocate într-o bază de date, dar în absența salvării acestora se pierd;
- absența corelațiilor între câmpuri , care evită situații de incompatibilitate , generatoare de alocări de resurse neavenite;
- operarea de variabile neinițializate;
- absența controlului alocării de memorie printr-o limită definită;
- lipsa concordanței dintre opțiunile definite si cele implementate efectiv;
- introducerea în uz curent a produselor ne-

testate prin întreaga structură asociată eşantionului reprezentativ, al grupului țintă;

Bibliografie

[COOK05] Jason Cook – *Site Optimization Tutorial*, articol în format electronic, <http://webmonkey.wired.com/webmonkey/design/tutorials/tutorial2.html>

[ISAA97] Scott Isaacs - *Inside Dynamic HTML*, Microsoft Press, 1997

[IVAN05] Ion IVAN, Alecu Felician – *Structuri HTML*, Editura ASE, București, 2005, ISBN 973-594-586-X

[KING05] Andrew King – *Extreme HTML Optimization*, articol în format electronic, <http://www.webreference.com/authoring/languages/html/optimize/index.html>

[NAST02] Floarea NASTASE, Pavel NASTASE – *Tehnologia aplicațiilor WEB XML – DOM – ASP*, Editura Economica, București, 2002, ISBN 973-590-644-9

[NAST98] Pavel NASTASE, Floarea NASTASE – *INTERNET world wide web JavaScript – HTML – Java*, Editura Economica, București, 1998

[STAU96] Todd Stauffer – *HTML by Example*, 1996, carte în format electronic