Testarea operationalitatii sistemelor software complexe

Cristian DIACONU
InterDigital SRL, Bucuresti, cristian.diaconu@interdigital.ro

In complex software systems, the testing problem must be really taken into consideration. Quality testing is an area for developing software leading to important steps for obtaining a successful product. Tacking into account that the operationality is the main quality characteristic for complex systems, operationality testing is an important basis for finding product's errors. In this paper we present a conceptual model for operationality testing, as well as characteristics and modalities for performing operationality testing.
***Keywords**: Operationality, Testing of (Functionality, Usability, Reliability, Efficiency and Security), Complex systems, Banking system, Verification and validation*

1 Fundamentele testarii operationalitatii

Operationalitatea se defineste ca o caracteristica de calitate a produselor software care depinde de urmatoarele atribute de calitate: functionabilitatea, utilizabilitate, fiabilitatea, eficienta, siguranta. Aceasta definitie este larg dezbatuta în lucrarea [DIAC03]. Conform acestei definitii testarea operationalitatii presupune verificarea atributelor de calitate enumerate mai sus.

Pentru a spune ca un sistem este operational trebuie analizat si verificat daca corespunde atributelor sale de calitate.

Procesul de analiza si control al unui sistem software este denumit *proces de Verificare si Validare (V&V)*. În cadrul acestui proces se utilizeaza doua tehnici de analiza si control:

- *inspectarea software* analizeaza si controleaza sistemul reprezentat prin: documente de cerinte, diagrame de proiectare si cod sursa (tehnica statica de verificare si validare).
- *testarea software* implica executia programului cu date de test si examinarea rezultatelor precum si a comportamentului operational si al cerintelor (tehnica dinamica de verificare si validare).
- Ne vom restrânge domeniul articolului în general la tehnica dinamica de V&V, adica la testarea software si în particular la testare operationalitatii.

2. Testarea operationalitatii

Testarea operationalitatii presupune testarea atributelor sale de calitate, astfel încât sa putem spune la sfârsitul testului, ca sistemul este operational.

2.1 Testarea functionalitatii. Verificarea functionalitatii unui produs se poate face atât prin metode statice cât si dinamice. Cele statice se refera la verificarea documentelor de cerinte, de analiza si respectiv documentele tehnice. Dupa verificarea tuturor documentelor se poate spune ca ceea ce a intrat ca *input* în etapa de dezvoltare este corect, si se trece efectiv la metoda dinamica de testare a functionalitatii.

Scopul final al testarii functionalitatii este de a confirma încrederea ca sistemul software este exact ceea ce s-a cerut (*fit for purpose*). Aceasta presupune ca sistemul sa fie complet fara defecte, cu alte cuvinte, sistemul trebuie sa fie bun de utilizat.

În timpul verificarii si validarii, defectele descoperite trebuie rezolvate, ceea ce conduce la modificarea programului iar sistemul trebuie reverificat si revalidat. Toate acestea se fac într-un ciclu repetitiv numit *regression test.*

Testarea functionalitatii este cunoscuta si sub denumirea de *black-box testing* deoarece se testeaza comportamentul cu datele de intrare si rezultatul lor la iesire. Pe lânga incorectitudinea sistemului sau lipsa unor functionalitati, în timpul testarii operationalitatii se pot descoperi si erori de interfata, erori de date sau de performanta, care vor fi contorizate si plasate la testele corespunzatoare atributelor operationalitatii.

Datele de test sunt foarte importante în testarea functionalitatii unui produs si depind foarte mult de complexitatea produselor supuse testarii. În documentele care se realizeaza ca rezul-

tat al testarii, apar diferiti termeni ca: defect, eroare, anomalie, problema, insucces si faimosul cuvânt „bug" care este în general considerat atunci când: software-ul nu face ceea ce în specificatia produsului este spus ca face; face ceva ce în specificatia produsului se spune sa nu faca; face ceva ce în specificatia produsului nu este mentionat; nu face ceea ce în specificatia produsului nu este mentionat dar trebuie facut; este dificil de înteles, greu de utilizat sau lent.

Erorile descoperite pe parcursul testarii pot fi cauza diferitelor faze ale dezvoltarii unui produs si pot fi descoperite în:

- specificatie - erori în documentele de specificatie primite.
- proiectare - erori generale sau particulare în solutiile de proiectare adoptate;
- dezvoltare - erori în scrierea efectiva a codului;
- altele (configurare, etc.)

O mare parte a erorilor sunt în specificatie, deoarece aceasta poate fi insuficienta (nu s-au atins anumite punte), se schimba continuu sau se datoreaza unei comunicari insuficiente cu echipa de dezvoltare. Dupa erorile de specificatie pot urma, din punct de vedere al numarului, atât scrierea de cod cât si de proiectare, aceasta depinzând de complexitatea proiectului, de tipul de proiect si de echipa cu care se realizeaza produsul.

Validarea este terminata când software-ul functioneaza într-o maniera rezonabila acceptata de client. Acceptarile rezonabile sunt definite în documentul ce cuprinde *specificatia cerintelor*, în care se descriu toate atributele cerute de client. Specificatiile contin o sectiune numita *criterii de validare* care reprezinta baza testului de validitate.

2.2 Testarea utilizabilitatii

Multe companii de software cheltuiesc foarte mult timp si bani pe gasirea celei mai bune cai de proiectare a interfetei cu utilizatorul. Pentru aceasta se pun sub observatie diferiti utilizatori, pentru a detecta ce taste folosesc, ce greseli fac si cum folosesc mouse-ul. Alegerea si implementarea solutiei optime depinde de multi factori, printre care si subiectivismul. Un standard ales de producator poate fi considerat bun din punctul lui de vedere, dar din punctul de vedere al clientului poate fi mai putin bun.

Pentru realizarea unei bune interfete cu utilizatorul s-au desprins sase caracteristici importante:

- *Urmarirea utilizarii unor standarde*. Cea mai importanta caracteristica a interfetei cu utilizatorul este urmarirea unui standard existent. De obicei, standardele sunt dependente de sistemele de operare. Daca o platforma nu are un standard sau în software-ul dezvoltat se doreste adaptarea unui standard nou, atunci echipa de proiectare va crea un standard utilizabil.
- *Intuitiv*. Alegerea unui standard care sa fie cât mai usor de utilizat.
- *Consistenta*. Folosirea consecventa a standardului ales în toate aplicatiile sau modulele dezvoltate (taste rapide, meniu, terminologie, numire, consistenta în mesajele de erori, plasament al butoanelor etc.)
- *Flexibilitate*. Sa permita setarea diferitelor caracteristici ale aplicatiei dorite de clienti (culori, valori implicite, diferite cai de introducere de date etc.)
- *Confortabil.*
- *Concret*.
- *Utilitate*.
- *Accesibilitate*.

Ca parte a testarii utilizabilitatii este si testarea documentatiei, care de multe ori este considerata ca fiind o componenta non-software. Dar o documentatie buna duce automat la cresterea utilizabilitatii produsului si implicit la cresterea operationalitatii.

2.3 Testarea fiabilitatii. Daca în urma testarii utilizabilitatii s-a verificat ca sistemul are toate functiile necesare si acestea functioneaza, trebuie ca la sfârsitul testarii fiabilitatii sa putem spune ca sistemul raspunde corect si în timp util la cerinte sale. Pentru acesta, datele cu care se realizeaza testul trebuie sa fie complete, astfel încât sistemul sa treaca prin toate punctele cel putin o data. Prin aceasta, se demonstreaza sau nu, ca sistemul raspunde corect la toate tipurile de date.

Pentru a realiza cel de al doilea obiectiv si anume sa raspunda în timp, atât în conditii reale de utilizare, cât si în conditii extreme, trebuie sa se realizeze un test de stres.

Testarea în conditii aride este foarte importanta pentru a nu duce la pierderea sau distrugerea datelor. Acest test se face atât la nivelul modulelor cât si la nivelul global al întregii aplicatii, însa, pentru verificarea cerintelor de performanta, aceasta testare se face dupa ce integrarea este completa. Testarea de performanta implica atât elemente software cât si elemente hardware.

2.4 Testarea eficientei Un sistem software eficient presupune verificarea urmatorilor factori:

- resursele necesare software si hardware pentru a realiza functiile cerute de sistem;
- costul efectiv de implementare în comparatie cu avantajele oferite de sistemul dezvoltat;
- interfatarea sistemului cu alte sisteme si costul necesar realizarii acestor interfete ;
- costul de întretinere si portabilitate;
- costul pentru distribuirea sistemului si respectiv învatarea sistemului.

Costurile trebuie tinute într-o limita acceptabila în raport cu bugetul alocat realizarii, implementarii si utilizarii proiectului. Costul utilizarii produsului este strâns legat de profitul direct sau indirect pe care îl aduce utilizarea acestuia.

2.5 Testarea sigurantei Siguranta unui sistem presupune atât captarea, transferul si retinerea datelor în conditii de siguranta cât si securitatea sistemului în ansamblu. Securitatea sistemului este data în special de domeniul de activitate al produsului. Astfel, securitatea poate fi de la „foarte mica" pâna la „foarte mare". Domeniul bancar este un domeniu unde securitatea trebuie sa fie spre nivelul maxim. Trebuie tinut cont ca sporirea securitatii este proportionala cu costurile necesare implementarii acestei securitati.

Daca crestem costurile de securitate pentru a fi convinsi ca sistemul final o sa fie sigur, s-ar putea ca sistemul sa nu mai îndeplineasca alt atribut al operationalitatii (cel al eficientei). Nivelul la care un sistem este sigur poate fi stabilit si din alt punct de vedere: un sistem este sigur atunci când efortul pentru spargerea sistemului este mai mare decât rezultatele obtinute prin spargere.

O particularitate importanta pentru aceasta testare o au *sistemele critice*, ale caror erori duc la o pierdere economica semnificativa, o deteriorare fizica sau afecteaza sanatatea umana. Testarea sigurantei la aceste sisteme este foarte importanta din doua motive:

- accidentele sunt evenimente rare în sistemele critice si ele practic sunt imposibil de simulat în timpul testarii sistemului;
- exista cerinte care sunt nesigure si acestea sunt imposibil de demonstrat prin testare sau alte operatii de validare.

3. Concluzii

Pentru ca un sistem sa fie operational trebuie sa fie îndepliniti toti cei cinci factori de calitate al operationalitatii. Testarea operationalitatii presupune testarea acestor factori în ansamblu. Nu putem spune ca un sistem este operational daca în urma testarii unul din factorii sai nu este îndeplinit. În [PRSS01] sunt prezentate un numar de reguli care pot fi considerate ca obiective alte testarii:

- testarea este un proces de executie a unui program cu intentia de a gasi o eroare;
- un caz de test bun este unul care are probabilitate mare de a gasi o eroare nedescoperita înca;
- un test terminat cu succes este un test care descopera o eroare nedescoperita anterior.

Fiecare proiect trebuie sa contina un plan amplu de testare care sa cuprinda toata functionalitatea aplicatiei si sa asigure functionarea corecta a întregului produs. Strategiile de testare se concentreaza pe functionalitate a si utilizabilitatea produsului. Se pune problema testarii explicite a operationalitatii doar în sistemele informatice complexe, sistemul informatic bancar, care cuprinde sute de tipuri de tranzactii, executii online sau offline, cât si un nivel de securitate foarte mare.

Bibliografie

[DICR03] - C. Diaconu, *Evaluarea operationalitatii sistemelor software complexe*, Bucuresti, în curs de editare.

[SCHM00] - M. Schmidt, *Implementing the IEEE Software engineering standard*", SAMS Macmillan Computer Publishing, 2000

[SOMM01] - I. Sommerville, *Software Engineering 6th Edition*, Addison-Wesley, 2001

[PESS01] - R. Pressman, Software engineering a practitioner's approach, McGRAW-Hill, 2001