

Managementul calitatii software

Prof.dr. Ion IVAN, Catedra de Informatica Economica, A.S.E. Bucuresti
Dr.ec. Laurentiu TEODORESCU, SOFTPLUS

Cultura calitatii în societatea contemporana a determinat aparitia unor modele de management corespunzatoare. Managementul calitatii este parte componenta a stiintei managementului dar si a managementului operational al firmelor si organizatiilor.

Introducerea în practica a tehnicilor managementului calitatii produce modificari în cadrul organizatiilor, atât în ceea ce priveste organizarea procesuala si structurala (compartimentele de asigurare a calitatii ce au aparut pe lângă controlul tehnic de calitate, cercurile calitatii, servicii de audit si controlling, echipe transfunctionale) cât si în ceea ce priveste mentalitatile managerilor si executantilor, pozitia clientilor, pozitia proceselor unul fata de altul, etc. Calitatea nu mai trebuie doar verificata, ci produsa.

Cuvinte cheie: calitate, organizare, management.

1. Introducere

Conceptia moderna despre calitate priveste acest concept în dinamica sa si leaga calitatea produsului sau serviciului de calitatea concepiei (proiectului) si calitatea fabricatiei. Conform standardelor ISO, calitatea fabricatiei reprezinta gradul de conformitate a produsului cu documentatia tehnica. iar calitatea proiectului exprima masura în care proiectul produsului asigura satisfacerea cerintelor beneficiarilor si posibilitatea de folosire, la fabricatia produsului a unor procedee tehnologice rationale si fezabile din punct de vedere economic (figura 1).

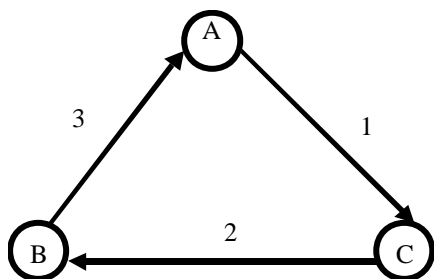


Figura 1

unde:

A = cerintele beneficiarului;

B = caracteristicile calitatii prevazute în documentatia tehnica;

C = caracteristicile produsului;

1 = calitatea concepiei;

2 = calitatea fabricatiei;

3 = calitatea produsului (sau serviciului).

La aceste notiuni se adauga si conceptul de calitate livrata. Aceasta se refera la nivelul real al calitatii produsului sau serviciului, în momentul achizitionarii de catre beneficiar.

2. Calitatea produselor software – particularitati

Daca societatea contemporana este o societate a calitatii, atunci ea este, în aceeași masura, o societate a informatiei si a tehnologiei informatiei.

În industria de software, întâlnim aceeași concentrare pe produs a concepiei, executiei, asigurarii si verificarii calitatii, ca în faza preindustrială, dar, pe de alta parte, datorita complexitatii produsului rezultat, exista grupul de executanti, împartit în colective sau indivizi specializati, care actioneaza pe parcursul unor înlantuirii de faze (ciclu de viata). Modelul de productie în industria software este un model de “productie a proiectelor”. Particularitatea consta în faptul ca activitatile desfasurate pot fi specifice unei anumite faze a ciclului de viata, sau pot fi independente de fazele ciclului de viata [ISO2].

Importanta calitatii produselor software rezida în cel puțin trei aspecte: erorile din programele de aplicatie pot fi fatale în anumite domenii unde vietile oamenilor depind de acestea; aceste erori pot provoca pierderi financiare, materiale si tot felul de alte tipuri de insatisfactii sau pierderi; daca în domeniul produselor hardware costurile au o tendinta generala de scadere, în domeniul dezvoltarii de software, desi productivitatea a crescut substantial, nu se înregistreaza si o scadere a costurilor care sa duca la aceeasi tendinta.

Acest ultim aspect se datoreaza particularitatilor prin care calitatea se manifesta în domeniul produselor software, asa cum sunt ele relevate în [BARO1]:

- comportamentul instructiunilor nu se deterioreaza în timp;
- erorile sunt provocate de folosirea sau combinarea incorecta a componentelor elementare, si nu de aceste componente în sine;
- interactiunile dintre componentele unui program sunt, mai complexe, mai ales daca acestea ruleaza în cadrul unor aplicatii complexe;
- erorile exista deja în program, ele sunt eliminate cu timpul, prin depanare, deci programul se îmbunatateste prin trecerea timpului;
- eliminarea unei erori nu da siguranta ca s-a diminuat numarul total de erori cu o unitate;
- non-calitatea programelor poate fi atribuita în întregime greselilor umane, de proiectare, conceptie, programare, documentare.

Un manager preocupat de calitatea produselor software trebuie sa posede, conform [WEI2], anumite abilitati speciale si chiar calitati native, cum ar fi: sa observe ce se întâmpla si sa înțeleaga semnificatia propriilor observatii; sa se poata comporta si sa poata actiona congruent în situatii interpersonale dificile, chiar si când este derutat, suparat sau speriat; sa poata înțelege situatiile complexe. Aceasta ultima

abilitate permite sa se poata planifica un proiect si apoi sa se observe si sa se actioneze astfel încât proiectul sa decurga conform planului sau sa poata fi modificat conform cerintelor si schimbarilor aparute pe parcurs.

Calitatea produselor software este definita în [BARO3] ca masura în care acestea satisfac cerintele utilizatorilor prin caracteristici tehnice, economice si psiho-sociale. Aceasta definitie se bazeaza pe doua concepte care, puse în legatura, dau masura calitatii unui produs software, si anume cerintele utilizatorilor si caracteristicile produsului.

3. Principii de proiectare a proceselor de dezvoltare software

Masura fundamentala a oricarui proces cu feedback controlat este posibilitatea de a compara ceea ce a fost planificat a se realiza, cu ceea ce s-a realizat în fapt. Proiectarea proceselor de dezvoltare software are la baza doua activitati:

- A. *analiza riguroasa a cerintelor si specificatiilor*, fara de care nu se poate face o evaluare asupra dimensiunilor, efortului de realizare si a obiectivelor privind calitatea;
- B. *planificarea masurabilitatii* proceselor si produselor software intermediare si finale, fara de care nu se poate asigura nici gestiunea si nici managementul operational al calitatii.

Desi evolutiile lumii reale determina modificarea continua a cerintelor chiar si în timpul procesului de realizare a proiectului, ceea ce conduce, la un dialog permanent între producatorul de software si beneficiar, procesul iterativ de aproximare a cerintelor trebuie sa îngusteze aceasta tendinta si sa o îndrepte catre ceva ce se poate finaliza si pune în opera.

Asa cum se arata în [ZUL1], clientii cumpara sau accepta produsele software atât timp cât: 1. îi ajuta în rezolvarea problemelor; 2. îi ajuta în sesizarea oportunitatilor de îmbunatatire a propriei activitati; 3. le ofera o imagine buna pe piata, în fata

clientilor lor cei mai semnificativi; 4. le usureaza munca si contribuie la îmbunatatirea confortului acesteia. Aceste elemente constituie motive puternice pentru focalizarea atentiei echipei de dezvoltare software în directia sesizarii corecte a problemelor si oportunitatilor clientului. În cadrul managementului calitatii totale, în vederea livrării catre clienti a unor produse de valoare, s-a dezvoltat în Japonia, metodologia desfasurării (explicitării) functiei calitatii - Quality Function Deployment - QFD - ca o structura sofisticata, cu mai multe subsisteme. Instrumentele si tehnicile acestei metodologii sunt aplicate în prezent în proiectele de dezvoltare de hardware, servicii si software.

Planificarea masurabilitatii si a ceea ce se va masura pentru managementul calitatii este un element esential al proiectarii gestiunii calitatii, în cadrul oricarui proiect de dezvoltare de software. Aceasta presupune desfasurarea unui set minimal de activitati care sa puna bazele unui management de calitate si ale producerii de software de calitate. Premisele realizării unui proiect masurabil sunt, dupa opinia exprimata în [WEI1]:

- compunerea proiectului din procese masurabile;
- un sistem de creare si întreținere a unei vederi globale publice (rapoarte, scheme, reprezentari grafice) asupra progresului realizat în domeniul calitatii proiectului;
- un sistem de cerinte bine întocmite, care documenteaza tot ceea ce furnizorul si beneficiarul înțeleg - de comun acord - prin calitatea produsului software;
- un sistem de revizii si inspectii care sa masoare orice evolutie a calitatii proiectului.

Realizarea oricarui proiect presupune mutatarea unui sistem dintr-o stare A (initiala), într-o stare B (starea finala dorita). O asemenea transformare nu poate fi facuta fara perceperea exacta a starii A, cunoasterea cerintelor pentru starea finala B si cunoasterea realitatilor stariilor A si B. Gerald

Weinberg recomanda în [WEI1] urmatorii pasi pentru proiectarea generala a unui sistem de procese care sa duca la realizarea cu succes a unui proiect masurabil de dezvoltare de software si sa permita totodata managementul calitatii:

a) *Pregatirea pentru o succesiune de procese iterative.* Odata cu definirea completa, fara ambiguitati si corecta a proiectului, orice manager trebuie sa fie pregatit (cel puțin psihologic) pentru viitoarele rafinari si redefiniri;

b) *Identificarea clientilor si valorizarea optiunilor acestora.* Începând proiectul cu clientii si valorizându-le corespunzator optiunile si cerintele, aceasta presupune deja plasarea problemei calitatii pe “creata valului”;

c) *Selectarea obiectivelor posibile de atins.* Formularea obiectivelor trebuie sa fie clara si, exprimata în termeni nenegativi. De exemplu, în locul formulei: “Proiectul nu va putea sa depaseasca bugetul alocat”, se poate opta pentru formula: “Proiectul va avea un buget si un set de cerinte stabilite, de comun acord, de managerii organizatiei si de managerii proiectului, înainte de începerea acestuia. Proiectul se va executa în conditiile îndeplinirii setului de cerinte si în limita bugetului aprobat”.

d) *Stabilirea obiectivelor într-o maniera masurabila.* Stabilirea în acest mod a obiectivelor presupune existenta unui raspuns clar la orice întrebare de genul: “Cum se poate dovedi ca obiectivul X a fost atins?”. De aceea, în formularea obiectivelor trebuie evitate fraze cu un înțeles vag precum: “Se va actiona în directia cresterii calitatii si productivitatii”.

e) *Stabilirea obiectivelor trebuie facuta cu constrângeri minime asupra proiectului.* Se recomanda evitarea formularii unor obiective de genul: “Testarea produsului software se va face pe 100 statii de lucru IBM, care vor fi instalate pâna la data de 20 martie”. O alternativa care duce la minimizarea presiunilor asupra proiectului poate fi: “Produsul software va fi testat pe

80 până la 100 stații de lucru. Toate stațiile de lucru vor fi identice, iar configurația aleasă trebuie să fie cea recomandată de furnizorii următoarelor produse software”.

f) *Verificarea posibilelor obstacole.* Trebuie să se alcătuiască o listă care să cuprindă

cât mai multe posibile obstacole ce pot sta în calea realizării obiectivelor. Ulterior, fiecare posibil obstacol va fi analizat și se va stabili dacă poate fi într-adevăr un obstacol, precum și posibilele căi de depășire a acestuia (figura 2.).

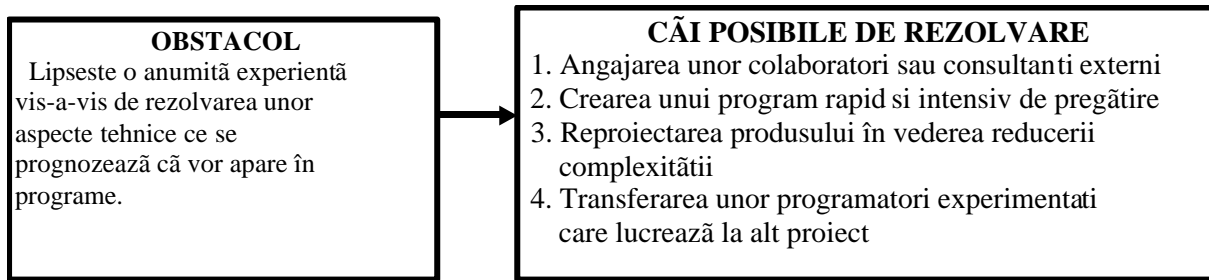


Figura. 2.

g) *Verificarea resurselor.* Acesta este un pas peste care mulți manageri de proiect trec cu ușurință, având în vedere faptul că resursele umane, financiare și materiale, sunt cuprinse în planurile proiectului. De exemplu, dacă clientul face următoarea afirmație: “Doresc o interfață utilizator foarte prietenoasă, care să faciliteze operarea”, dar la întrebarea producătorului “Cât timp puteți aloca pentru a defini exact cerințele dumneavoastră în ceea ce privește această interfață prietenoasă”, clientul răspunde: “Sunt prea ocupat pentru a face acest lucru. Voi faceți interfața și apoi voi fi mai explicit când voi vedea rezultatul”, acest răspuns releva cel puțin două aspecte: pe de o parte interesul clientului în realizarea interfeței respective nu este prea “arzător”, iar, pe de altă parte, clientul nu poate fi luat în considerare ca o resursă pentru proiect (ceea ce este mai grav).

h) *Planificarea inversă.* Pentru a putea planifica invers, este necesar ca, mai întâi, să existe o imagine clară a stării B (starea finală dorită). Pornind înapoi către starea inițială A, se identifică o stare intermediară C, din care starea B este cel mai probabil de atins, și așa mai departe: $A \rightarrow E \rightarrow D$

$\rightarrow C \rightarrow B$. Desigur că nici un plan real nu poate fi liniar.

Planificarea inversă poate fi eficientă în condițiile în care:

- lungimea grafului este suficient de mică, astfel încât elementele de incertitudine să fie diminuate;
- planificarea trebuie să fie incrementală, în pași realisti, astfel încât, dacă sistemul nu a atins starea intermediară dorită, să poată fi corectat în pasul următor.

4. Conceptul de software corespunzător pentru utilizare

Fiecare client are preferințe individuale, care pot fi satisfăcute prin caracteristici de calitate diferite. Această relație se reflectă puternic în industria de software. Astfel, alături de unii factori ai caracteristicii de flexibilitate, se afirmă tot mai mult posibilitatea de personalizare a produselor software, ca o caracteristică de calitate tot mai apreciată.

Pentru satisfacerea cerințelor, este important ca relația calitate-cumpărător să fie puternic reflectată nu numai în definirea calității, dar și în managementul și gestiunea acesteia, deoarece cumpărătorul hotărăște, în final, ce este calitatea. Astfel,

specificatiile, ca reflectari ale cerintelor identificate si definite ale beneficiarilor, nu reprezinta criterii de calitate absolute, ci numai mijloace necesare pentru satisfacerea asteptarilor.

În cazul managementului total al calitatii, relatia client-furnizor este generalizata prin internalizarea ei. Trecerea de la un proces la altul, de la o etapa la alta, este abordata conform principiului "urmatorul proces este clientul". Cele trei componente ale managementului calitatii totale sunt îmbunatatirea continua; satisfacerea utilizatorilor si avansul organizatiei. Tehnicile manageriale pentru realizarea lor sunt controlul statistic al proceselor (Statistical Process Control-SPC), desfasurarea functiilor calitatii (Quality Function Deployment-QFD) si explicitarea politicii (Policy Deployment- PD).

5. Concluzii

În pofida timpului scurt scurs de la nasterea industriei de software, se apreciaza ca fiecare sablon al acestei culturi organizationale evolutive, descrise prin nivelele de maturitate ale modelului Maturitate/Capabilitate dezvoltat la Software Engineering Institute - S.U.A., a avut propria contributie la dezvoltarea industriei software - de la a face calculatoarele mai putin îns-paimântatoare pentru publicul larg, pâna la a oferi modalitati de dezvoltare si tinere sub control a unor proiecte de dimensiuni mari.

Asa cum se prezinta în [BAL1], începând cu anul 1998, evaluarea si certificarea produselor software va beneficia de doua noi serii de standarde, - ISO 9126 si ISO 14598:

ISO/IEC 9126 - Caracteristici si metrici ale calitatii software	
ISO 9126-1	Caracteristici si subcaracteristici de calitate
ISO 9126-2	Metrici externe
ISO 9126-3	Metrici interne
ISO/IEC 14598 - Evaluarea produselor software	
ISO 14598-1	Generalitati
ISO 14598-2	Planificare si management
ISO 14598-3	Proces pentru dezvoltatori
ISO 14598-4	Proces pentru achizitori
ISO 14598-5	Proces pentru evaluatori
ISO 14598-6	Documentatia pentru modulele de evaluare

Dincolo de modelele oferite de literatura de specialitate, de diferite institute si organizatii (ISO, SEI, IEC), rezulta ca managementul calitatii este principalul factor care contribuie la statuarea meta-modelului calitatii software, aplicabil nu atât la nivelul organizatiilor producatoare de software, cât la nivelul întregii industrie. Acest meta-model al calitatii software nu reprezinta doar îmbunatatirile incrementale ce apar de la luna la luna sau de la an la an în domeniul analizat, ci consta în acumularea solida a micilor pasi ce construiesc drumul îmbunatatirii calitatii, de-a lungul istoriei acestei industrie.

Indiferent de definirea si operationalizarea managementului calitatii, asa cum se arata în [BAL1], "productia de software si furnizarea de servicii informatice asociate depind în mare masura de managementul efectiv al efortului intelectual", si "nici o natiune nu-si poate permite sa aiba o industrie de software care sa produca altceva decât software de calitate".

Bibliografie

- [BAL1] - Balog Alexandru, Olaru Marieta, s.a. - *Managementul calitatii si protectia consumatorilor*, vol.3, lito ASE, Bucuresti, 1997

2. [BARO1] - Baron Tudor - coordonator - *Calitate si fiabilitate - manual practic*, Editura Tehnica, Bucuresti, 1988
3. [BARO2] - Baron Tudor, Al.Balog, I.Ivan, C.Baron - *Informatica Economica: Fiabilitatea produselor program*, lito ASE, Bucuresti, 1985
4. [BARO3] - Baron Tudor, I.Ivan, Al.Balog - *The Characteristic System of Software Products Quality*, Economic Computation and Economic Cybernetics Studies and Research, ASE Bucharest, No.1, 1982, p.29-44
5. [ISO1] - *ISO/IEC 9126 International Standard - Information Technology - Software product evaluation - Quality characteristics and guidelines for their use*, 1991, Geneve, Switzerland
6. [ISO2] - *ISO 9000 Part 3 - Quality management and quality assurance standards - Guidelines for the application of ISO 9001 to the development, supply and maintenance of software*, 1991, Geneve, Switzerland
7. [ISO3] - *ISO 8402 International Standard - Quality, Vocabulary*, 1986, Geneve, Switzerland
8. [WEI1] - Weinberg Gerald M. - *Quality Software Management - First Order Measurement*, Dorset House Publishing, New York, USA 1993
9. [WEI2] - Weinberg Gerald M. - *Quality Software Management - System Thinking*, Dorset House Publishing, New York, USA 1993
10. [ZUL1] - Zultner Richard E. - *TQM for Technical Teams*, Communications of the ACM, October 1993, Vol.30, No.10, p.79-90