

Modele de calitate software

Student: Cucu Bogdan-Cristian
Grupa:442A

Cuprins

1. Introducere
2. Atribute de calitate pentru produsele software
3. Modele de calitate
 - 3.1 Modelul McCall
 - 3.2 Modelul Boehm
 - 3.3 Modelul FURPS
 - 3.4 Modelul Dromey
 - 3.5 ISO 9126
4. Concluzii
5. Bibliografie

1. Introducere calitate produs software

In contextul ingineriei software, calitatea unui produs software se refera la doua notiuni distincte:

- Calitatea functionala a software-ului, care se reflecta in cat de bine produsul software respectiv corespunde proiectarii, bazat pe diferite cerinte si specificatii;
- Calitatea structurala a software-ului, care se refera la necesitati cum ar fi robustetea sau mentenabilitatea, la gradul de corectitudine al software-ului dat.

Calitatea structurala este evaluata prin analiza structurii interne a software-ului, codul sau sursa. Calitatea functionala este sustinuta si masurata prin "testarea software".

Standardul IEEE exprima conceptul de calitate intr-un sens larg in functie de asteptarile consumatorului final.

Un produs software de calitate este un produs software care indeplineste un numar de factori de calitate. Acestea sunt descrise in cerinte si specificatii sau pot fi factori de calitate pe care producatorul ii impune, desi nu sunt inclusi in cerintele clientului.

Calitatea este data de totalitatea functionalitatilor si caracteristicilor unui produs care satisface nevoile date, deci este conform cu specificatiile. Reprezinta totodata gradul in care reuseste sa indeplineasca cerintele clientului.

Evaluarea calitatii unui produs software are la baza:

- Un model de calitate
- Caracteristici de calitate
- Metrici de calitate.

In aceasta lucrare vor fi prezentate modele de calitate definite prin standarde.

2. Atribute de calitate pentru produsele software

2.1 Functionalitatea

Functionalitatea constituie un set de atribute care se bazeaza pe existenta unui set de functionalitati si proprietatile lor prevazute.

Aceasta inseamna ca software-ul trebuie sa dispuna de functii si servicii asa cum este prevazut in specificatii.

Acest atribut are mai multe sub-caracteristici:

- Gradul de precizie: exprima nivelul de corectitudine
- Interoperabilitatea: indica daca formatul de date cu care lucreaza programul tinta se conformeaza standardelor internationale
- Respectarea cerintelor: exprima cat de bine produsul software dat respecta cerintele impuse de dezvoltator. Nu se pot masura in timpul procesului de dezvoltare.
- Securitatea: se refera la modul in care software-ul are capacitatea de a se proteja impotriva accesului neautorizat
- Conformitatea: indica daca software-ul respecta standardele internationale, este certificat, etc.

2.2 Fiabilitatea

In general, fiabilitatea se poate defini ca probabilitatea ca un sistem sau software sa cedeze intr-un anumit interval de timp. Altfel spus, fiabilitatea exprima capacitatea unui produs de a mentine un anumit nivel de toleranta, atunci cand este utilizat in anumite conditii.

Cuprinde mai multe sub-caracteristici:

- Maturitatea: in contextul software, se refera la diferitele versiuni comerciale ale unui software si ciclul de aparitie a unei noi versiuni;
- Recuperabilitatea: masoara abilitatea software-ului de a isi reveni dintr-o eroare neasteptata si de a restaura datele salvate inainte de aparitia erorii;
- Toleranta la defectare: indica daca produsul software dat poate sa pastreze un anumit nivel de toleranta la defectari

2.3 Accesibilitatea

Este abilitatea unui produs software de a fi inteles, invatat, folosit, configurat si rulat in anumite conditii. Cuprinde mai multe subcaracteristici:

- Usurinta de intelegere: capacitatea software-ului de a ii facilita utilizatorului intelegerea modului de utilizare a produsului pentru indeplinirea unor anumite sarcini;
- Usurinta de invatare: se refera la usurina cu care un dezvoltator poate sa se acomodeze cu produsul software dat. Spre exemplu, acesta trebuie a dispuna de o documentatie si de un meniu de tip 'help'.
- Atractivitatea: se refera la existente unei interfete atractive
- Conformitatea: indica daca software-ul se conformeaza standardelor internationale.
-

2.4 Eficienta

Exprima capacitatea unui software de a furniza un anumit nivel de eficienta in utilizarea resurselor sistemului de calcul pe care ruleaza. Fara a afecta anumite specificatii, anumite module interne ale software-ului pot fi optimizate pentru un plus de performanta.

2.5 Mentenabilitatea

Describe abilitatea software-ului de a fi modificat.

Subcaracteristici sunt:

- Stabilitatea: abilitatea software-ului de a gestioneza situatiile neprevazute .
- Analizabilitatea: auto-analiza software-ului
- Testabilitatea: module de auto-verificare interna a software-ului

2.6 Portabilitatea

Software-ul poate rula pe medii diferite, eventual cu modificari minore. Astfel, softul respectiv necesita sa fie adaptabil si sa se poata instala pe platforme diferite.

3. Modele de calitate

Modelul de calitate consta dintr-un set de caracteristici si sub-caracteristici, precum si relatiile dintre ele, care furnizeaza baza pentru specificarea cerintelor si evaluarea calitatii componentei sau sistemului software.

Exista mai multe modele de calitate propuse pana acum.

Cel mai cunoscut este modelul McCall, propus de catre McCall si Joseph in 1978. Ei au impartit attributele in trei categorii:

- Operatii asupra produsului soft;
- Revizuri ale softului;
- Tranzitia produsului soft.

Ei au propus un set de 11 caracteristici pentru aceste categorii.

Al doilea model propus a fost modelul Boehm, model ce include si caracteristicile hardware, documentatia si alte aspecte.

Alt model propus este FURPS, model ce descompune calitatea in doua categorii:

- Cerinte functionale: definesc intrarile si iesirile dorite;
- Cerinte non-functionale: definesc factori precum: utilizabilitatea, fiabilitatea

Modelul Dromey a adaugat inca doua caracteristici: re-utilizabilitatea si maturitatea.

ISO propune prin standardul 9126 o definitie generala a calitatii in contextul software prin 6 mari caracteristici.

3.1 Modelul McCall

Modelul McCall este unul dintre cele mai cunoscute modele de calitate software. A fost dezvoltat de catre armata americana si promovat de catre DoD(Department of Defence) al SUA. Are trei perspective majore pentru definirea si identificarea calitatii unui produs software:

- Revizia produsului
- Tranzitia produsului
- Operatii asupra produsului

Revizia produsului se refera la mentenabilitate(efortul necesar pentru a localiza si corecta o eroare in program in cadrul platformei pe care ruleaza), flexibilitatea(usurinta in a realiza schimbari in platforma sub care ruleaza) si

testabilitatea(usurinta cu care se testeaza programul pentru a se asigura ca nu contine erori).

Tranzitia se refera la portabilitate (efortul necesar de modifica programul astfel incat acesta sa ruleze si pe alte platforme), reutilizabilitatea(utilizarea softului intr-un context diferit) si inter-operabilitate(cuplarea doua sisteme diferite).

Calitatea operatiilor asupra produsului depinde de corectitudinea acestuia, programul trebuie sa indeplineasca cerintele de proiectare si fiabilitate, eficienta, ca timp de executie si spatiu de stocare, precum si utilizarea resurselor hardware(UCP, memorie), constrangeri de integritate(protejarea inpotriva accesului neautorizat) si utilizabilitatea, adica usurinta utilizarii.

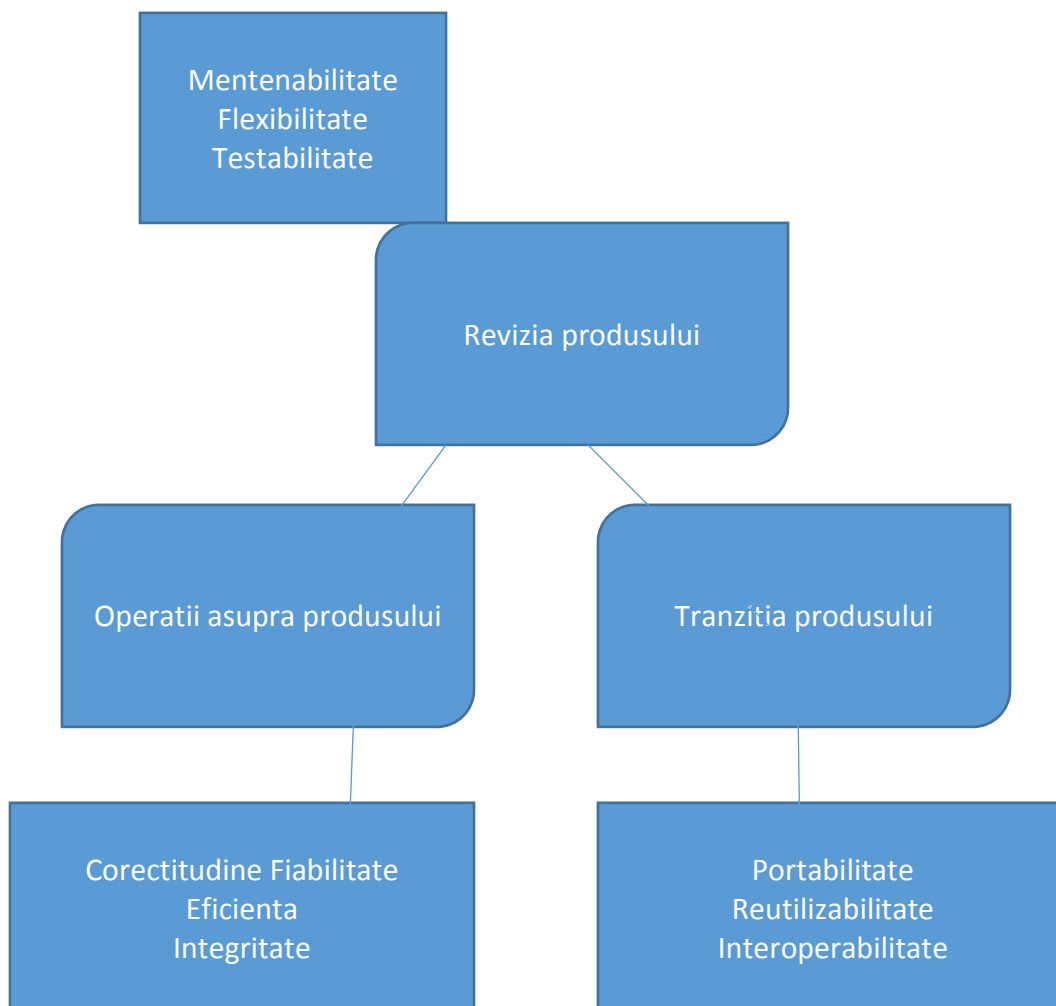


Figura de mai sus poarta numele de: "Triunghiul de calitate al lui McCall"

Factorii de calitate prevazuti de model sunt:

- Corectitudine;
- Fiabilitate;
- Eficienta;
- Integritate;
- Utilizabilitate;

Aceasta abordare intentioneaza sa dezvolte un set de metrici si expresii pentru fiecare factor, astfel:

$$F_{cal}=a_1m_1+a_2m_2+\dots+a_nm_n$$

Unde:

F_{cal} este factorul de calitate

$a_i, i=1,n$ sunt coeficientii de regresie

$m_i, i=1,n$ sunt metricile corespunzatoare factorului de calitate

3.2 Modelul Boehm

Acest model incearca sa defineasca calitatea software folosind un set de attribute si metrici. Caracteristicile de nivel inalt adreseaza trei probleme pe care cumparatorul produsului software si le pune:

- Utilitatea: accesibil, fiabil, eficient
- Mentenabilitatea: usurinta in utilizare, daca se poate modifica si re-testa
- Portabilitatea

Caracteristicile de nivel mediu reprezinta cei 7 *factori de calitate* care se refera la calitatile pe care un produs software trebuie sa le ofere:

- Portabilitatea: codul este conceput spre a putea fi rulat pe platforme hardware diferite, eventual , cu mici modificari;
- Fiabilitatea
- Eficienta: in termeni ce tin de utilizarea resurselor;
- Utilizabilitatea
- Testabilitatea
- Usurinta de utilizare
- Flexibilitatea

Caracteristicile de nivel inferior sunt caracteristicile primitive ale ierarhiei Boehm.

Diferenta intre modelul McCall si modelul Boehm consta in aceea ca primul se axeaza pe masurarea cat mai precisa a caracteristicilor, in timp ce modelul Boehm este bazat pe o gama mai mare de caracteristici si se axeaza pe criteriul ce priveste mentenabilitatea produsului software.

Urmatorul tabel prezinta comparativ cele doua modele:

Factor	McCall	Boehm
Corectitudine	da	
Fiabilitate	da	da
Integritate	da	da
Utilizabilitate	da	da
Eficienta	da	da
Mentenabilitate	da	da
Testabilitate	da	
Interoperabilitate	da	
Flexibilitate	da	da
Reutilizabilitate	da	da
Portabilitate	da	da
Claritate		da
Modificabilitate		da
Documentatie		da
Resilienta		da
Accesibilitate		da
Validitate		da
Functionalitate		
Generalitate		da
Economie		da

Cu alte cuvinte, modelul Boehm se concentreaza pe raportul cost-eficienta.

3.3 Modelul FURPS

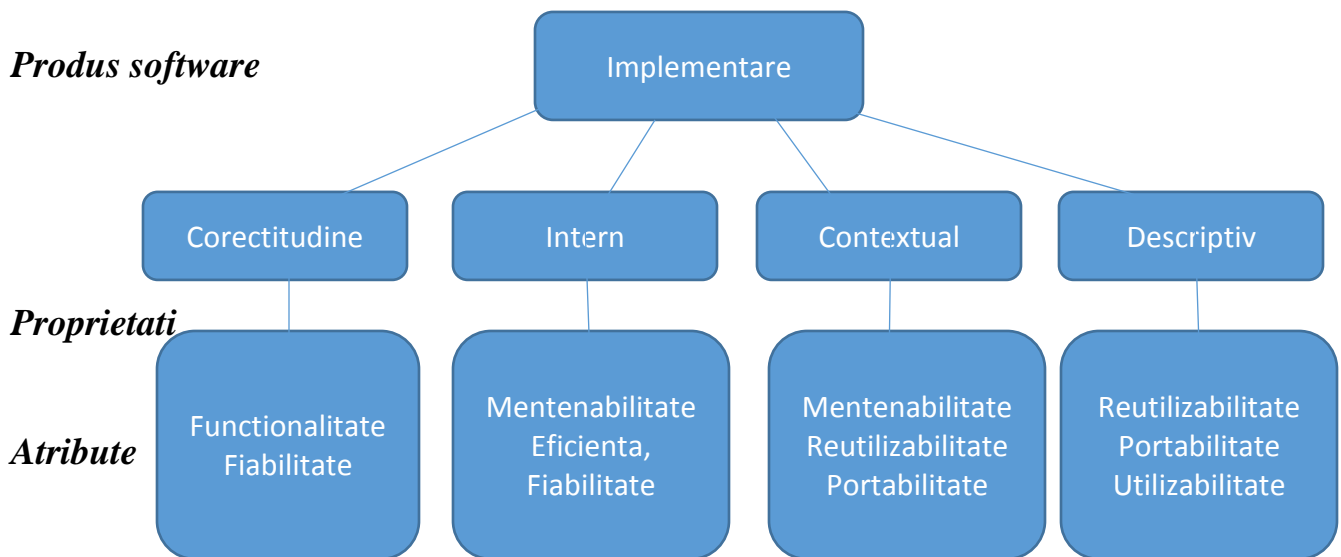
Modelul FURPS este un model mai putin cunoscut, inasa cu o structura asemanatoare primelor doua prezentate. Acum este parte a IBM Rational Software. Acronimul FURPS provine de la:

- Functionalitate: seturi de caracteristici, capabilitati si securitate;
- Utilizabilitate: include factori ce tin de estetica, consistenta interfetei cu utilizatorul , meniu de help, asistent de configurare(wizard), documentatia aferenta aplicatiei/produsului software si alte materiale ajutatoare;
- Fiabilitatea(Reliability): frecventa si severitatea defectarilor, posibilitatea de revenire din eroare, predictibilitatea;
- Performanta: impune conditii de functionare precum viteza, eficienta, rate binare de transfer, timpul de raspuns, nivelul de utilizare a resurselor;
- Suportabilitatea: include testabilitatea, extensibilitatea, adaptabilitatea, mentenabilitatea, gradul de configurare.

Criteriile enumerate de FURPS se impart in doua categorii: functionale (F) si non-functionale(URPS). Ambele categorii de criterii pot fi utilizate atat in faza de formulare a cerintelor, cat si pentru evaluarea calitatii produsului.

3.4 *Modelul Dromey*

Un model mai recent, asemanator cu cele deja prezentate este modelul Dromey, model ce vizeaza relatia dintre attributele de calitate si sub-attributele corespunzatoare acestora.



3.5 ISO 9126

Acest model impune un set caracteristici si recomandari pentru evaluarea calitatii programelor software.

Caracteristicile vizate sunt urmatoarele:

- Portabilitatea
- Functionalitatea;
- Fiabilitatea;
- Accesibilitatea;
- Eficienta
- Mentenabilitatea.

Este un standard bazat pe modelele McCall si Boehm, care insa impune si criteriul functionalitatii.

4. Concluzii

In prezenta lucrare am prezentat cateva din modelele de evaluare a calitatii unui produs software. Modele diferite iau in considerare factori diferiti corelati cu necesitati functionale si non-functionale, insa unul din cei mai importanti factori este dat de fiabilitatea software-ului respectiv.

Pentru realizarea unui program de calitate este nevoie ca etape precum: planificarea, analiza necesitatilor, a cerintelor si implementarea(codarea si depanarea) sa respecte anumite proceduri.

5. *Bibliografie*

[1][https://www.bth.se/com/besq.nsf/%28WebFiles%29/CF1C3230DB425EDCC125706900317C44/\\$FILE/chapter_1.pdf](https://www.bth.se/com/besq.nsf/%28WebFiles%29/CF1C3230DB425EDCC125706900317C44/$FILE/chapter_1.pdf)

[2]<http://www.pearsonhighered.com/samplechapter/0201729156.pdf>

[3]Stephen H. Kan. *Metrics and Models in Software Quality Engineering*. Addison-Wesley, Boston, MA, ed. II, 2002.

[4]ISO, International Organization for Standardization, "ISO 9126-1:2001, Software engineering – Product quality, Part 1: Quality model", 2001