

CALITATEA PRODUSELOR PROGRAM (sau produse software)

Calitatea unui produs sw trebuie interpretata in conformitate cu conceptul de "scop de utilizare" considerand atat atributile interne (caracteristicile produsului) cat si cele externe (scopul folosirii). Din diferitele definitii a calitatii, ne referim aici la "totalitatea functiunilor si caracteristicilor unui produs sau serviciu care are abilitatea de a satisface cerintele stabilite anterior sau generate.

Diferite definitii ale calitatii:

1. **calitatea** este construirea din timp a solutiilor problemelor inainte ca ele sa apara, si proiectarea lor intr-un produs sau serviciu.
2. **calitatea** este satisfacerea clientului
3. **calitatea** este productivitatea angajatilor care sunt pregatiti, primesc instrumente si instructiuni de care au nevoie pentru a-si executa sarcinile
4. **calitatea** este flexibilitatea de a modifica sistemul software pentru a obtine cererile utilizatorilor si ale beneficiarilor
5. **calitatea** este eficienta de a face lucrurile rapid si corect
6. **calitatea** este obtinerea unui orar la termen
7. **calitatea** este un proces de imbunatatire continua
8. **calitatea** este o investitie in a face economie, pt ca a face bine un lucru de la inceput este mai putin costisitor decat sa fie corectat ulterior

De aceea, evaluarea calitatii sw-ului tine cont de urmatoarele puncte de vedere:

1. cel al **managementului**, care este interesat mai degrabă în calitatea totală decât în calitatea unei caracteristici și are nevoie să balanțeze imbunatatirea calitatii cu criteriile de management;
2. cel al **utilizatorului**, care consideră că calitatea sw-ului este data de proprietatile ce satisfac corect și eficient necesitatile prezente și viitoare ale celui care cumpără și folosește produsul sw;
3. cel al **dezvoltatorului**, care consideră că calitatea este data de conformitatea cu cerințele funcționale și de performanță explicit fixate, cu standardele de dezvoltare explicit documentate și cu caracteristicile implicate ale unui sw dezvoltat într-un mod profesional;

Din aceste puncte de vedere rezulta o lista de criterii ale calitatii necesare produsului final. Aceste criterii sunt impartite in 5 grupe: performanta, dependenta, cost, intretinere si criteriile utilizatorului final. Performanta, dependenta si criteriile utilizatorului final sunt specificate de cerintele funcționale sau furnizate de domeniul problemei. Celelalte 2 tipuri sunt dictate de client si furnizor.

Criterii de performanta – includ cerinte de viteza si spatiu impuse sistemului

CRITERII DE PERFORMANTA	Definitie
Timp de raspuns	Cat de repede stie sistemul de cererea utilizatorului dupa ce aceasta a fost transmisa?
Iesirea	Cat de multe task-uri realizeaza sistemul intr-o perioada de timp?
Memoria	De cată memorie are nevoie sistemul pentru a rula?

Criterii de dependenta - determină efortul depus pentru minimizarea caderilor sistemului și a consecințelor acestora

CRITERII DE DEPENDENTA	Definitie
Robustete	Abilitatea de a trata o intrare invalidă de la utilizator
Siguranta	Diferența dintre comportamentul specificat și cel observat
Valabilitate	Procentul din timpul sistemului folosit pentru realizarea sarcinilor normale
Toleranța implicită	Abilitatea de a opera în condiții eronate
Securitate	Abilitatea de a răspunde la forțari a sistemului
Protectie	Abilitatea de a nu pune în pericol viații omenesti, chiar în prezența erorilor și eșecurilor

Criterii de cost – includ costul dezvoltării, repartiției și administrării sistemului, precum și cele generate din considerații manageriale.

CRITERII DE COST	Definitie
Costul de dezvoltare	Costul dezvoltarii sistemului
Costul de repartitie	Costul instalarii sistemului si pregatirea utilizatorilor
Costul upgrad-arii	Costul translatarii datelor de la sistemul anterior, in cazul reengineering. Acest criteriu rezulta din cerintele de compatibilitate cu sistemul initial.
Costul de intretinere	Costul necesar rezolvarii bug-urilor si imbunatatirii sistemului
Costul de administrare	Banii necesari administrarii sistemului

Criterii de intretinere – determina cat de greu este sa modificam sistemul dupa tranzitia acestuia la client.

CRITERII DE INTREȚINERE	Definitie
Extensibilitate	Cat de usor este sa adaugam noi functionalitati sau noi clase la sistem?
Modificabilitate	Cat de usor modificam functionalitatea sistemului?
Adaptabilitate	Cat de usor portam sistemul la un nou domeniu al problemei?
Portabilitate	Cat de usor portam sistemul la o platforma diferita?
Lizibilitate	Cat de usor se intlege sistemul citind codul acestuia?
Urmarea cerintelor	Cat de usor mapam codul la cerintele sistemului?

Criteriile utilizatorului final cuprind calitati dorite de utilizator si care nu au fost acoperite de criteriile de performanta si de dependenta.

CRITERII UTILIZATORULUI FINAL	Definitie
Utilitate	Cat de bine suporta sistemul munca utilizatorului?
Utilizare	Cat ii este de usor utilizatorului sa foloseasca sistemul?

Totusi, unele criterii sunt contradictorii, ca de exemplu: calitate inalta vs cost mic si proiectantii trebuie sa faca anumite compromisuri. Exemple:

COMPROMIS	Motivatia
Spatiu vs viteza	Daca sw-ul nu indeplineste timpul de raspuns impus sau cerintele de iesire, poate fi expandata mai multa memorie pentru a mari viteza software-ului (eg caching).
Termenul de livrare vs functionalitate	Daca dezvoltarea a depasit orarul, manager-ul proiectului poate livra la timp produsul dar cu mai putina functionalitate de cat era planificata initial, sau sa furnizeze cu toata functionalitatea specificata dar depasind termenul
Termen de livrare vs calitate	Daca testarea a depasit orarul, managerul proiectului poate livra sw-ul la timp, dar cu bug-uri stiute (si apoi sa le rezolve) sau sa livreze mai tarziu, dar cu cat mai multe bug-uri rezolvate
Termen de livrare vs staff	Daca dezvoltarea a depasit orarul, managerul proiectului poate adauga resurse pentru a creste productivitatea, dar care determina cresterea costului de dezvoltare. Pentru majoritatea proiectelor, aceasta optiune este valabila la inceputul proiectului.

(aceste criterii sunt verificate in faza de proiectare)

Modele de calitate

Definitie. Un **model de calitate** este multimea caracteristicilor si relatiilor dintre ele care furnizeaza baza pentru specificarea cerintelor de calitate si evaluarea calitatii sau ca un set structurat de proprietati necesare ca un obiect al unei clase sa indeplineasca scopurile definite.

Avantajul modelului de calitate este dat de descompunerea valorii (un proces, un produs sau o organizatie) intr-o lista de caracteristici, subcaracteristici si masuri care se aplica prezicerii/asigurarii/ verificarii obtinerii unui scop definit despre obiectul inainte/in timpul/dupa producerea sa.

Clasificarea acestor modele in functie de:

1. Numarul de straturi:

- 1.1. **2 straturi** (Boehm si McCall), care arata o lista de caracteristici impartite intr-un set de subcaracteristici; avem o descriere apropiata de punctul de vedere al utilizatorului.
- 1.2. **3 straturi**, unde este adaugat un nivel tehnic al metricilor subcaracteristicilor. Acest nivel permite o evaluare intr-o maniera extrem de subiectiva, la fel de importanta ca cea obiectiva.

STRAT	BOEHM	McCALL	ISO	IEEE	DROMEY
1	Caract. de nivel H	Factor	Caracteristica	Factor	Atribut de nivel H
2	Caract. primitive	Criterii	Subcaracteristica	Subfactor	Atribut subordonat
3 ¹	(Metrica)	(Metrica)	(Metrica)	Metrica	

2. Numarul de relatii intre primele 2 straturi:

- **relatie 1:n**, ca in ISO 9126 – fiecare caracteristica are propriul sau set de subcaracteristici
- **relatie n:m**, ca in FCM- McCall Factor Criteria Model – fiecare caracteristica este legata de una sau mai multe caracteristici.

Procedura de calcul a factorului de calitate (QF)

Aceasta procedura returneaza o valoare numERICA care reprezinta parerile utilizatorilor, dezvoltatorilor si manager-ilor asupra calitatii sw-ului.

Pas1. Utilizatorii, manager-ii si dezvoltatorii completeaza niste chestionare de calitate

Pas2. Strangerea acestor chestionare

Pas3. Calcularea factorului de calitate

Instrumente necesare calcularii QF:

1. chestionarul (forma Q1): instrumentul tehnic de strangere a datelor
2. o lista de tabele - pe care le putem automatiza cu o foaie de calcul:
 - tabelul A: contine o lista folosita pentru selectarea acelor mai relevante subcaracteristici de calitate si raporteaza rezultatele chestionarelor
 - tabelul B: este tabelul de calculare a QF-ului, folosit impreuna cu
 - tabelul C: folosit pentru calcularea prioritatilor caracteristicilor
 - tabelul D: listeaza ponderile generice pentru fiecare rang

Dupa ce au fost completate chestionarele, avem urmatorii pasi:

1. Determinarea subcaracteristicilor de calitate cele mai importante. ISO 9126 da urmatoarele nivele:

Marcaj	Rata	Rata globală
3	Excelent	Satisfacător
2	Bine	
1	Moderat	
0	Slab/Absent	Nesatisfacător

Valorile coloanei tabelului A trebuie copiate in coloana 9 din tabelul B pentru subcaracteristicile selectate.

2. Determinarea prioritatii caracteristicilor. Aceasta prioritate (intre 1 si 6) se determina in functie de numarul descrescator al subcaracteristicilor alese din chestionarele individuale. Cand 2 sau mai multe caracteristici au aceeasi prioritate, poate fi aleasa una din urmatoarele optiuni:
 - a) numarul total al subcaracteristicilor este egal cu numarul caracteristicilor – vom da acelasi rang. De exp, siguranta si utilizarea au ambele rangul 3. Urmatoarele caracteristici in ordine vor avea rangul 5.
 - b) numarul total al subcaracteristicilor este diferit de numarul caracteristicilor – consideram procentul caracteristicii respective. De exp, siguranta si functionalitatea au ambele rangul 3. Vom da rangul 3 sigurantei si rangul 4 functionalitatii, deoarece $n/3 > n/5$, unde n este numarul de subcaracteristici alese din totalul fiecarei caracteristici (3 si 5 in ordine).
 - c) numarul total al subcaracteristicilor este numarul total al mai multor caracteristici – in acest caz vom lua mai intai criteriul a) si apoi criteriul b). De exp, daca siguranta, intretinerea si portabilitatea au rangul 3, rangul final va fi: siguranta va avea rangul 3, celelalte 2 vor avea 4. Urmatoarea caracteristica va avea rangul 6.
3. Asignarea ponderilor caracteristicilor. Fiecarei pozitii de rang i se asociaza o pondere, pentru fiecare grup de interes. In cazul rangurilor egale, ponderea este egala cu media aritmetica a ponderilor.

¹ (Metrica) inseamna ca arhitectura modelului respectiv nu mentioneaza oficial acest strat, chiar daca exista si este necesar evaluarielor

Caracteristica [1]	Nr de subcaracteristici [2]	Prioritate (U,M,D) [3]	SCV [4]	Ponderea prioritara [5]	CV [6=4*5]
Functionalitate	5	1	15	p1	15p1
Siguranta	3	4	9	(p4+p5)/2	9*(p4+p5)/2
Utilitate	3	4	9	(p4+p5)/2	9*(p4+p5)/2
Eficienta	2	6	6	p6	6p6
Intretinabilitate	4	2	12	(p2+p3)/2	12*(p2+p3)/2
Portabilitate	4	2	12	(p2+p3)/2	12*(p2+p3)/2

$$TCV = \sum_{i=1}^6 CV_i$$

si astfel $TCV_{max} = 15p_1 + [9/2(p_4+p_5)]2 + 6p_6 + [12/2(p_2+p_3)]2$

Cum $p_n = p_6(6-n+1)$ cu $1 \leq n \leq 6$ este posibil sa exprimam TCV prin p_6

$$TCV_{max} = 15(6p_6) + 9(3p_6 + 2p_6) + 6p_6 + 12(p_6 + 4p_6) \text{ si } p_6 = TCV_{max} / 249$$

Daca $TCV_{max} = 100$ atunci $p_6 = 0.401606425$

Deci tabelul D ar fi

RANG	ONDERE
1	2.409638
2	2.008032
3	1.606425
4	1.204819
5	0.803212
6	0.4016064

Suma valorilor din coloanele 3-57 din tabelul C trebuie copiate in tabelul C, coloana 8. In tabelul C, coloana 9 inseram rezultatul impartirii valorilor din coloana 8 la 3 (numarul de puncte de vedere examineate); aceasta valoare trebuie reprodusa in linia corespunzatoare a tabelului B coloana 11 (col. "Pondere prioritara").

4. Suma valorilor subcaracteristicilor (SSV). Trebuie sa punem in tabelul A, coloana 10 suma valorilor subcaracteristicilor pentru fiecare caracteristica:

$$SSV = \sum_{i=1}^6 SSV_i, \text{ unde } n \text{ este numarul subcaracteristicilor, intre 2 si 5 (functionalitate)}$$

5. Calcularea valorii caracteristicilor (CV). Se inmulteste coloana 10 cu coloana 11 din tabelul B si valorile obtinute se pun in coloana 12.

6. Calcularea valorii totale a caracteristicilor (TCV). Aceasta valoare se obtine de adunarea valorilor din coloana 12 si reprezinta nivelul de calitate estimat pentru proiect

$$TCV = \sum_{i=1}^6 CV_i$$

In final factorul de calitate este obtinut din urmatoarea formula:

$$QF = TCV / TCV_{max}$$

TEHNICI DE CONTROL A CALITATII (FOLOSITE IN FAZA DE TESTARE A PRODUSELOR PROGRAM)

Siguranta este o masura a succesului cu care comportamentul observat al sistemului este conform cu comportamentul specificat al sau. Siguranta sw-ului este probabilitatea cu care un sistem sw nu provoaca esecul sau pentru o perioada de timp in anumite conditii specificate. **Esecul** este orice deviere a comportamentului observat de la comportamentul specificat. O **eroare** indica faptul ca sistemul este intr-o stare in care procesarea mai departe a acestuia va conduce la esecul acestuia. O greseala, numit si **defect** sau bug este cauza mecanica sau algoritmica a unei erori.

In continuare prezintam 3 tehnici folosite pentru cresterea sigurantei unui sistem sw:

1. **Tehnici de evitare a defectelor** – incearcă sa previna aparitia erorilor si esecurilor, gasind defectele inainte ca sistemul sa fie livrat. Aceste tehnici cuprind:

a) **Metodologii de dezvoltare** – evita defectele furnizand tehnici ce minimizeaza introducerea defectelor in modelele sistemului si a codului. Aceste tehnici includ reprezentarea neambigua a cerintelor, folosirea abstractizarii si incapsularii datelor, minimizarea cuplarii intre subsisteme si maximizarea coerentei subsistemelor, definirea din timp a cerintelor sistemului.

- b) **Gestiunea configuratiei** – evita defectele cauzate de modificarea nedisciplinata a sistemului. De exp, o greseala care se face frecvent este modificarea interfetei sistemului fara sa anunte toti dezvoltatorii componentelor apelante.
- c) **Verificarea** - incarca sa gaseasca defectele inaintea oricarei executari a sistemului. Verificarea este posibila in anumite cazuri, cum ar fi verificarea unui kernel al unui sistem de operare => verificarea are limite. Nu poate fi aplicata pentru a asigura calitatea sistemelor mari si complexe. Mai mult, aceasta presupune ca cerintele sunt corecte.

O **revizie** este inspectia manuala a partilor sau a tuturor aspectelor unui sistem fara sa se execute sistemul. Sunt 2 tipuri de revizii:

- **walkthroughs** (revizii tehnice formale). Intr-un walkthrough a codului, dezvoltatorul prezinta neoficial API-ul, codul si documentatia asociata componentei, echipei de revizie. Aceasta comenteaza maparea analizei si proiectarii codului la cod folosind cazuri de utilizare si scenarii din faza de analiza.
- **inspectia** – este similara cu walkthroughs numai ca prezentarea componentei este oficiala si dezvoltatorul nu are voie sa participe. Totusi, dezvoltatorul poate fi chemat daca echipa de revizie are nevoie de clarificarea definirii si folosirii structurilor de date sau a algoritmilor. Echipa de revizie verifica interfata, codul componentei conform cerintelor, eficienta algoritmilor conform cerintelor nefunctionale.

2. Tehnici de detectare a defectelor

- incarca sa gaseasca defectele in sistem, dar nu incarca sa recupereze din esecurile generate de ele. In general, aceste tehnici sunt aplicate in timpul dezvoltarii, dar in unele cazuri sunt folosite si dupa furnizarea sistemului.

Sunt 2 tipuri de astfel de tehnici:

- a) **depanare** (debugging) presupune ca defectele pot fi gasite de la inceputul unui esec neplanificat. Sunt 2 tipuri de depanari:
 - depanarea corectitudinii- cu scopul de a gasi orice deviatie a cerintelor functionale observate fata de cele specificate
 - depanarea performantei – se adreseaza devierii intre cerintele nefunctionale observate si cele specificate, cum ar fi timpul de raspuns
- b) **testarea** incarca sa creeze esecurile sau erorile intr-un mod planificat. Aceasta tehnica permite dezvoltatorului sa detecteze esecuri in sistem inainte ca acesta sa fie livrat. Notam ca aceasta definitie a testarii implica faptul ca un test are succes daca a detectat erori. Aceasta definitie este folosita in toate fazele de dezvoltare. O alta definitie a testarii este ca “demonstreaza ca erorile nu sunt prezente”. Aceasta definitie se foloseste numai dupa dezvoltarea sistemului cand se verifica ca indeplineste cerintele functionale si nefunctionale.

Activitatile de testare: testarea unitatilor (componentelor), testarea integrata si testarea sistemului.

3. Tehnici de toleranta la defecte

Toleranta la defecte este recuperarea din esec in timp ce sistemul se executa. Aceasta tehnica permite unui sistem sa revina dintr-o esuare a unei componente trimitand informatii despre starea eronata inapoi la componentele apelante, presupunand ca una din ele stie ce sa faca in acest caz. Sistemele de baze de date furnizeaza tranzactii atomice de recuperare din esec in timpul unei secvente de actiuni ce trebuie executate impreuna sau nu. Redundanta modulara se bazeaza pe faptul ca escurile sistemului provin de la esecul componentelor. Sistemele modulare redundante sunt construite din asignarea mai multor componente de a realiza o aceeasi operatie. Sistemul poate continua chiar daca o componenta a esuat, deoarece celelalte componente realizeaza inca functionalitatea necesara.

GESTIUNEA RISCURILOR

Definitie.1 Riscurile sunt probleme potențiale ce rezulta dintr-o arie de incertitudine.

2. “Riscul este posibilitatea de a suferi pierderi”(Webster). In procesul de dezvoltare, pierderea descrie impactul asupra proiectului, care ar putea duce la scaderea calitatii produsului, cresterea costurilor, depasirea termenului de predare sau esec.

3. Riscul de dezvoltare sw a fost definit ca expunerea la unul sau mai multe tipuri de riscuri:

- risc performanta, sau esecul de a obtine toate beneficiile anticipate ale sistemului si software-ul dezvoltat
- risc cost sau depasirea semnificativa a costului estimat
- risc orar sau esecul livrarii produselor sw la milestone-urile planificate si datele cerute de utilizator
- risc suport sau livrarea unui produs care necesita costuri de intretinere mari datorate deficienelor in intretinere, flexibilitate, compatibilitate sau siguranta

Riscurile se diferențiază de celelalte evenimente ce au loc în timpul unui proiect software prin 3 aspecte:

1. O pierdere asociată cu evenimentul (riscul). În acest caz pierderea se numește **impactul riscului**. Evenimentul trebuie să creeze o situație care să duca la pierderea calității, controlului, timpului, și.m.d. De ex., o modificare radicală a cerintelor conduce la pierderi de timp și bani dacă proiectarea nu este suficient de flexibilă pentru a putea fi modificată rapid și ușor.
2. Probabilitatea (0-1) ca evenimentul să apară. Când probabilitatea riscului este 1, riscul se numește **problema**.
3. Gradul modificării care va avea loc. Pentru fiecare risc trebuie să luăm o serie de măsuri care să reducă sau să eliminate impactul riscului.

Există 2 surse majore generatoare de riscuri:

1. **riscuri generice** sunt comune tuturor proiectelor software, cum ar fi neînțelegere cerintelor, pierderea personalului cheie sau alocarea unei perioade de timp insuficientă pentru testare.
2. **riscuri specifice proiectului** rezultă din vulnerabilitățile proiectului respectiv.

Riscurile pot fi:

1. **organizatorice (de întreprindere)** sunt cele care pun în pericol supraviețuirea produsului
2. **manageriale** contin orice incertitudine legată de planul proiectului și duc la prelungirea perioadei de realizare a proiectului
3. **tehnice** sunt legate de modelele sistemului, inclusiv modificări a funcționalității sistemului, a cerintelor nefuncționale, arhitecturii sau implementării sistemului.

Boehm² arată primele 10 elemente de risc software:

1. probleme ale personalului
2. orar și buget nerealiste
3. dezvoltarea funcțiilor și proprietăților greșite
4. dezvoltarea unei interfețe user greșite
5. poleire (gold-plating)
6. modificarea continuă a cerintelor
7. probleme ale componentelor externe furnizate
8. probleme ale task-urilor externe realizate
9. probleme în realizarea performanței real-time
10. capabilități computer-stiință

Scopul gestiunii riscurilor este de a identifica problemele posibile în proiect și rezolvarea acestora înainte să aibă un impact important asupra datei de livrare și bugetului. Gestiona riscurilor este o activitate dinamică și se desfășoară pe tot ciclul de viață al procesului de dezvoltare (se accentuează aspectul continuu al gestiunii riscurilor). Începe cu planificarea riscurilor în primele faze ale proiectului și continua cu monitorizarea și rezolvarea riscurilor din fază de întreținere. Necesită participarea activă a întregii echipe a proiectului.

Sunt săptă **principii** care furnizează un cadru pentru gestiunea efectivă a gestiunii riscurilor:

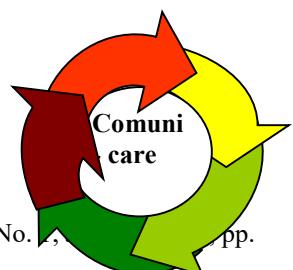
1. Perspectiva globală
2. Vederea în perspectivă
3. Comunicarea deschisă
4. Management integrat
5. Proces continuu
6. Vederea comună asupra produsului
7. Lucrul în echipă

În literatura de specialitate există câteva metodologii de gestiune a riscurilor: Boehm, SEI, Richard Fairley, Rockwell Risk Management Process.

În continuare vom prezenta procesul lui Boehm de gestiune a riscurilor care presupune realizarea a 2 categorii de activități:

1. **Evaluarea riscurilor** care conține 3 funcții:

- **Identificarea riscurilor.** Identifică elementele sau evenimentele (cum ar fi cerintele, tehnologii noi de dezvoltare sau o modificare în sistemele tinta) care pot avea un impact negativ asupra proiectului. Riscurile identificate sunt



² Barry Boehm, "Software Risk Management: Principles and Practices", *IEEE Software*, Vol. 8, No. 1, pp. 32-41.

descrise intr-un document (planul RMMM- vezi anexa) care va ghida procesul, facand parte din planul general al proiectului.

- **analiza riscurilor.** Pentru aceasta se pot folosi modele si tehnici ca: analiza de decizie, analiza costului riscurilor, analiza orarului, analiza sigurantei. Apoi fiecare risc este evaluat pentru a vedea impactul potential al riscului asupra proiectului. Fiecare risc este masurat in 2 moduri: probabilitatea aparitiei riscului si consecintele acestuia asupra proiectului.
- **prioritizarea riscurilor.** Se determina gradul de expunere a riscurilor, folosind cele 2 atribute asociate mai sus: probabilitatea de aparitie si consecintele acestuia in cazul in care apare. Atunci un risc face parte din una din urmatoarele categorii:
 - *probabil, impact potential inalt* (riscuri critice). Pentru acest tip de riscuri, managerii si dezvoltatorii ar trebui sa dezvolte planuri si sa le supravegheze indeaproape.
 - *improbabil, impact potential inalt*. Si astfel de riscuri ar trebui supravegheate, totusi, nu sunt necesare planuri de combatere a lor.
 - *probabil, impact potential mic*.
 - *improbabil, impact potential mic*.

2. Controlul riscurilor contine 3 functii:

- **planificarea gestiunii riscurilor.** Pasii gestiunii fiecarui risc se bazeaza pe probabilitatea aparitiei si pe impact. Aceste pasi pot determina cresterea costului proiectului, in termeni de resurse si durata proiectului.
- **rezolutia riscurilor.** Presupune rezolvarea sau reducerea riscurilor. Se folosesc tehnici ca decizii de personal, estimari cost/orar, monitorizarea calitatii, evaluarea tehnologiilor noi, prototipizarea (din timp, pentru a reduce riscurile mari), clarificarea cerintelor, simulare/modelare.

Exista 3 strategii pentru reducerea riscurilor:

- evitarea riscului, modificand cerintele pentru performanta sau functionalitate
- transferarea riscului alocand-ul celorlate sisteme
- presupunand riscul, prin acceptarea si controlul acestuia cu resursele proiectului
- **monitorizarea riscurilor.** Furnizeaza vizibilitatea si rezolutia in timp a riscurilor. Tehnici ca supravegherea milestone-urilor, supravegherea riscurilor critice si reevaluarea continua a riscurilor. In orice moment, manager-ul si fiecare dezvoltator ar trebui sa-si stabileasca primele 3 riscuri (dupa prioritati), fiind o activitate dinamica, ie rezolvarea unui risc presupune stabilirea altui risc ca fiind critic.

3. Comunicarea – furnizeaza informatii si rezultate externe si interne ale proiectului pe activitatile riscului, riscurile curente si riscurile care vor apare.

Fiecare risc trece secvential prin aceste activitati, dar activitatile pot fi concurente (eg., riscurile sunt prioritizate in paralel in timp ce noi riscuri sunt identificate si analizate), si iterativ (de exp, rezolvarea unui risc poate genera aparitia unui alt risc) in timpul ciclului de viata al proiectului.

In acelasi articol, Boehm recomanda tehnici de rezolvare a fiecarui risc identificat:

1. **nivelul staff-ului:** echipa cu experienta; spirit de echipa; pregatire; planificarea oamenilor cheie
2. **orar si buget nerealiste:** estimarea detaliata a orarului si a costului; proiectarea costului; dezvoltare iterativa; reutilizarea software-ului; clarificarea cerintelor
3. **dezvoltarea functiilor si proprietatilor gresite:** analiza organizationala; formularea conceptelor operationale; prototipizare; scrierea din timp a manualelor utilizatorului
4. **dezvoltarea unei interfete user gresite:** prototipizare; scenarii; analiza task-urilor
5. **poleire (gold-plating):** clarificarea cerintelor; prototipizare; analiza cost-beneficiu; proiectarea costurilor
6. **modificarea continua a cerintelor:** separarea modificarilor importante; ascunderea informatiei; dezvoltarea incrementala
7. **probleme ale componentelor externe furnizate:** proiectare competitiva sau prototipizare; verificarea referintelor
8. **probleme ale task-urilor externe realizate:** verificarea referintelor; inspectii; analiza compatibilitatii
9. **probleme in realizarea performantei real-time:** modelare; prototipizare; instrumentare; concordanta
10. **capabilitati computer-stiinta:** analiza tehnica; analiza cost-beneficiu; prototipizare, verificarea referintelor

Anexa. RMMM - Risk Mitigation, Monitoring and Maintenance Plan (Pressman)

Planul documenteaza toate operatiile de analiza a riscurilor si este utilizat de capul de proiect ca parte a planului general al proiectului.

I. Introducere

1. Obiectivele/scopurile documentului
2. Trecere in revista a riscurilor principale

3. Responsabilitati

Personalul managerial
Personalul tehnic

II. Tabela riscurilor

1. Descrierea riscurilor identificate

2. Factorii care influenteaza asupra probabilitatii de verificare a riscurilor si asupra consecintelor acesteia

III. Reducerea, supravegherea si guvernarea riscurilor

n. Riscul n

Reducerea probabilitatii

Strategia generala de abordare a verificarii riscului

Actiunile specifice îndreptate spre reducerea riscului

Supravegherea

Factori de supraveghere

Strategii de supraveghere

Guvernarea

Planul de urgență

IV. Scheduling al planului de riscuri