

**STUDIU DE CAZ PRIVIND REALIZAREA UNUI PROIECT DE
INGINERIE SOFTWARE**

CONTINUT

- 1. Ingineria software**
- 2. Probleme si abordari in dezvoltarea software**
- 3. Tipuri de proiecte software**
- 4. Dezvoltarea de ansamblu a unui sistem informational (SI) ca
proiect de inginerie software (PS)**
- 5. Realizarea SI cu parcurgerea tuturor etapelor ciclului de viață**
- 6. Alte tipuri de abordări în realizarea SI**
 - 6.1. Prototipizarea**
 - 6.2. Utilizarea pachetelor de programe aplicative standard**
 - 6.3 Realizarea SI cu resurse proprii utilizatorului**
 - 6.4 Realizarea SI prin servicii externalizate (outsourcing)**
- 7. Bibliografie**

1. Ingineria software

Specialistii in calculatoare folosesc cunostintele despre calculatoare si prelucrarea sau transmisia datelor pentru a rezolva diferite probleme din cele mai diverse domenii ale activitatii umane. Unele probleme nu necesita solutii care se realizeaza cu ajutorul calculatoarelor, de aceea in primul rand **trebuie inteleasa natura problemei** printr-un proces de analiza a ei si a solutiilor posibile si dupa aceea gandita utilizarea calculatorului daca este necesara si de dorit, din motive de eficienta.

Investigarea oricarei probleme incepe cu **analiza** ei, divizand-o in subprobleme care pot fi intelese si rezolvate mai usor, ca solutii. Dupa aceea se poate oferi o solutie pentru componentele care compun acea problema, prin reunirea lor, printr-un proces de **sinteza**. Deci tehnica rezolvarii unei probleme (**problem solving technique**) are doua pari: una de analiza a problemei pentru a determina natura ei si a solutiilor posibile si alta de sintetizare a unei solutii pentru acea problema pe baza analizei facute. In rezolvarea unei probleme specialistii folosesc diferite **unelte (tools)**, ca instrumente/sisteme automate cu ajutorul carora se rezolva o problema mai repede, sau **tehnici (techniques)**, ca procedura formală sau metoda folosita in obtinerea unui rezultat. In acest cadru general de rezolvare a unei probleme **ingineria software** valorifica posibilitatile oferite de calculator (teorie, functii etc) pentru a rezolva problemele beneficiarilor din diferite domenii, folosind unelte si tehnici specifice domeniului, pentru a realiza produse software de calitate si utilitate, acceptate de beneficiar si competitive pe piata. Un produs software de calitate inalta trebuie sa posede caracteristici care sunt apreciate de catre:

- **Utilizator**, daca este usor de asimilat si de utilizat
- **Dezvoltator**, care il proiecteaza si codifica
- **Intretinator**, care asigura operarea, intretinerea si asistenta tehnica

Calitatea software-ului cuprinde caracteristici speciale, care sunt corelate, referindu-se nu numai la performantele hardware si software, dar si la nevoile si asteptarile utilizatorului [1]:

- **Portabilitate**, inteleasa ca independenta de hardware
- **Utilitate**, inteleasa ca fiabilitate, eficienta si interfata prietenoasa/umana

- **Mentenabilitate**, inteleasa ca testabilitate, claritate si intelegere, modificabilitate (structurare, extindere)

In termeni mai generali, un software de calitate este considerat atunci cand el realizeaza ceea ce utilizatorul doreste sa faca, daca foloseste eficient si corect resursele de calcul, daca este usor de asimilat si utilizat de catre utilizaor, daca dezvoltatorul poate proiecta, codifica, testa si intretine sistemul relativ usor.

De obicei, orice **proiect software** pleaca de la ceva care exista intr-o organizatie, presupunand reunirea utilizatorilor cu anumite activitati realizate cu hardware si software, cu fisiere si baze de date, cu alte sisteme de calcul amplasate local sau la distanta. Deci proiectul trebuie definit prin **cuprinderea sa ca granite, ca obiecte si activitati incluse in componenta sa, ca relatii intre aceste elemente care il compun**. Aceasta abordare sistemica (**systems approach**), dupa ce este definita, permite trecerea la constructia sistemului in cadrul unei abordari ingineresti (**engineering approach**).

Proiectele software, in abordare sistemica si inginereasca, se desfasoara similar proceselor de inginerie in constructii civile. Procesul de elaborare si dezvoltare a unui proiect de software include urmatoarele activitati [1, 2, 3]:

1. **Analiza cerintelor si definirea proiectului**
2. **Proiectarea sistemului**
3. **Proiectarea programelor**
4. **Scrierea programelor (codificarea)**
5. **Testarea unitatilor si modulelor de programe**
6. **Integrarea unitatilor si modulelor**
7. **Testarea sistemului**
8. **Livrarea sistemului (instalare, punere in functiune, instruire)**
9. **Intretinerea (si dezvoltarea)**

In situatiile ideale, activitatile se desfasoara succesiv (in cascada), pana la ultima, cand sistemul se considera realizat. In realitate, intre diferitele activitati apar reactii inverse si reluari ale unor activitati, pentru a face unele schimbari sau pentru a clarifica anumite cerinte, fie in paralel, fie succesiv.

2. Probleme si abordari in dezvoltarea software

In realizarea proiectelor de software (PS) apar multe probleme, care presupun refaceri, modificari, completari, datorita specificului relatiilor dintre utilizatori si dezvoltatori, precum si specificului domeniului. In acest sens ne vom referi la cateva probleme care apar frecvent in realizarea PS [1]:

- ✚ **Schimbarea cerintelor si restrictiilor**, se refera la posibilitatea revederii cerintelor initiale din partea utilizatorului, atat in fazele initiale cat si mai tarziu. Intarzierea in exprimarea acestor modificari poate duce la costuri si intarzieri mari in realizarea PS. Cauzele modificarilor pot fi si de alta natura:
 - Dificultatea utilizarii unor fisiere sau a bazei de date, conform cu promisiune utilizatorului
 - Dificultati in interfatarea cu aplicatii existente
 - Abateri de la performantele si functiile specificate in prospectele echipamentelor si software-ului achizitionate

Aceasta situatie impune ca dezvoltatorii sa foloseasca unelte si tehnici suficient de flexibile, pentru a face mai usoara executia corectiilor

- ✚ **Realizarea PS prin completare (phase development systems)**, prin abatere de la realizare in cascada, datorita cerintei de a realiza elemente noi ale PS care sa se integreze cu unele existente, sistem cunoscut ca realizare prin defazare sau completare. Aceste situatii apar si atunci cand nu sunt inlocuite complet sistemele manuale cu cele automate, impunandu-se inlantuirea lor.
- ✚ **Interactiunea cu alte sisteme**, in situatiile in care fie se primeste, fie se trimite informatie/date. Aceasta situatie se complica atunci cand sistemele care schimba informatii sunt dezvoltate in paralel, iar interfetele nu sunt bine definite si documentate.
- ✚ **Natura si specificul calculatoarelor si software-ului**, care se refera la natura diferita a sistemelor digitale/discrete si continui/analoge, urmand a functiona impreuna, sau logica folosita in dezvoltarea software-ului.

3. Tipuri de proiecte software

Clasificarea PS ca fiind mici, medii, sau mari, se face in raport cu gradul de complexitate, definit de cateva caracteristici de dificultate in realizarea lor. Unele caracteristici implica probleme de complexitate, dificultate, sau scala in raport cu problemele tehnice. Altele depind de expertiza dezvoltatorilor si gradul de precizie in stabilirea specificatiilor sau cerintelor pe timpul dezvoltarii sistemului. Aceste caracteristici sunt prezentate in tabelul urmator [1]:

Caracteristici	Sisteme mici	Sisteme medii	Sisteme mari
Numarul functiilor realizate	mic	mediu	mare
Noutatea functiilor	standard	Similare cu cele existente, cu unele in plus	Complet noi, cu noi abordari
Numarul utilizatorilor care cer multi-user sau acces concurrent	1	cativa	multi
Multitasking	nu	uneori	da
Interactiv vs batch access	batch cu min. de interactiv.	Foarte interactiv	Foarte interactiv
Timpul de raspuns	off-line, noncr.	moderat	Real-time
Necesitatea distribuirii prelucrarii	nu	2 calculatoare	min. 3 calc.
Capacitatea de memorare	un disc	min. 2 discuri	multe, cu controlul accesului
Dimensiunea tranzactiei	mica	medie	mare
Remote vs local	numai local	remote	Remote acc.
Tolerance for downtime	cateva ore	timp redus	da
Cerinte de securitate	nu	moderate	inalte
Interactiunea cu alte sisteme	nu	Uneori, bine definita	Frecvent, in. paralel
Numarul fazelor in dezvoltare	nu	putine	multe
Dependenta de hardware	independent	putin	da, cu restr.
Stabilitatea cerintelor	fixe, precise	cu schimbari	mica

Intelegerea acestor caracteristici si gradul de dificultate ne poate ajuta sa evaluam cat poate dura constructia unui sistem software. Orice PS este asociat cu realizarea unui **sistem informational (SI)** in cadrul unei organizatii, de aceea vom prezenta modul cum se realizeaza un SI ca proiect de inginerie software si ce abordari/solutii exista pentru atingerea acestui obiectiv.

4. Dezvoltarea de ansamblu a unui sistem informational (SI) ca proiect de inginerie software (PS)

Orice SI este realizat ca o soluție la apariția unei probleme sau a unui set de probleme în organizație. Problemele pot apărea ca o sesizare în legătură cu performanțe slabe care se obțin în anumite procese/activități, sau din nevoia valorificării unei oportunități, din care se pot obține avantaje pentru organizație, pentru a deveni mai competitivă [2, 3].

Dezvoltarea S.I. se referă la ansamblul activităților care permit realizarea unei soluții pentru un SI, prin care se rezolvă anumite probleme și/sau se valorifică anumite oportunități care apar pe piață. Procesul de dezvoltare a SI este reprezentat de un set de activități distincte, care se desfășoară într-o ordine secvențială, unele repetându-se iar altele desfășurându-se în paralel, în funcție de abordarea implementării SI (care se va prezenta într-un capitol separat, mai departe). Fiecare activitate presupune o interacțiune strânsă cu organizația (Figura 1).

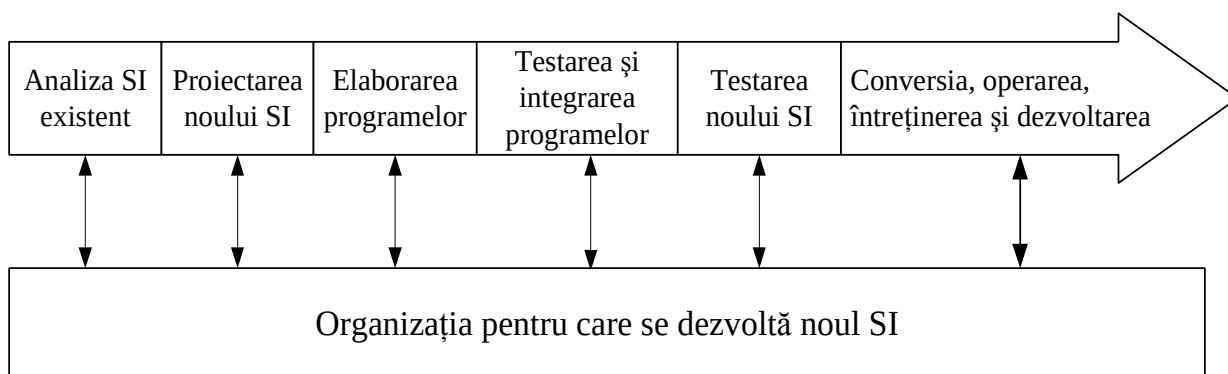


Figura 1. Procesul de dezvoltare etapizat a SI

Activitățile prezentate în această figură se desfășoară de obicei într-o ordine secvențială. Membrii organizației sunt implicați în acest proces, iar finalizarea SI implică schimbări în cadrul organizației [3]:

1. Analiza sistemului existent se poate defini ca analiza problemei/problemelor pe care organizația intenționează să le rezolve prin realizarea SI. În această fază, rolul principal îl are **analistul de sistem**, care realizează, în principal, următoarele activități:

- definirea problemei/problemelor și identificarea cauzelor care au dus la apariția ei/lor;
- specificarea soluțiilor și a fezabilității lor, prin realizarea unui studiu de fezabilitate;

- stabilirea cerințelor informaționale cerute de soluțiile propuse.

Analistul poate identifica problemele prin examinarea documentelor și procedurilor din organizație, prin observarea directă a operării sistemului existent, prin discuții și interviuri cu persoane cheie din sistem. Se ajunge la realizarea unei „hărți” (road map) a organizației, cu identificarea posesorilor și utilizatorilor primari de date în cadrul organizației. Aceștia au un interes direct în modul prin care noul SI, sau un sistem dezvoltat al celui existent, afectează datele și informațiile existente. De obicei, activitatea unui analist se finalizează cu propunerea dezvoltării SI existent sau realizarea unui SI nou, prezentând și un **studiu de fezabilitate**, prin care se demonstrează că soluția propusă este fezabilă și realizabilă, cu ce constrângeri (limitări) și cu ce resurse. Un asemenea studiu are o cuprindere asupra a trei domenii:

- **Fezabilitatea tehnică**, care se referă la componentele hardware, software și alte resurse tehnice.
- **Fezabilitatea economică**, care pune în balanță beneficiile soluției propuse și costurile realizării ei.
- **Fezabilitatea operațională**, care se referă la acceptarea și dorința de a utiliza soluția propusă, în cadrul managementului și structurilor organizaționale din organizație.

În mod normal, în faza de analiză se propun soluții alternative, fiecare cu fezabilitatea ei:

- menținerea sistemului existent;
- modificarea, pentru a îmbunătăți sistemul existent;
- dezvoltarea unui nou sistem.

Un raport scris va prezenta costurile, beneficiile, avantajele și dezavantajele fiecărei alternative, iar managementul organizației va decide care soluție este cea mai convenabilă.

O altă sarcină dificilă a analistului o reprezintă **definirea cerințelor informaționale specifice**, impuse de soluția recomandată. În principal, aceste cerințe presupun identificarea celor care au nevoie de informație, unde (în ce locație), când (la ce timp) și în ce formă și ce conținut, pe scurt: **cine, unde, când și cum** (who, where, when and how). În acest fel se ajunge la o descriere detaliată a funcțiilor și obiectivelor sistemului propus. Cerințele, cu alte cuvinte, trebuie să considere aspectele tehnice, economice și de timp, obiectivele, procedurile și procesele decizionale din organizație. Orice greșeli realizate în definirea cerințelor, au repercusiuni majore asupra fazelor următoare și o influență foarte mare asupra costurilor de realizare a SI.

2. Proiectarea sistemului este etapa în care trebuie realizată arhitectura și structura sistemului, la nivel fizic și logic, conform specificațiilor și cerințelor rezultate din faza de analiză, astfel încât să fie îndeplinite obiectivele urmărite. **Proiectantul de sistem** detaliază cerințele și specificațiile rezultate din faza de analiză, pentru toate nivelurile și componentele sistemului care va fi realizat.

Principalele tipuri de specificații care sunt definite în faza de proiectare se referă la:

- **ieșirile sistemului** (mediul pe care apar, conținutul lor și timpul la care apar);
- **intrările în sistem** (originea/sursa intrărilor, fluxul parcurs de intrări și modul de introducere și preluare a acestor intrări);
- **interfețele cu utilizatorii** (simplitate, eficiență, logica relației om – mașină, reacția inversă și tratarea erorilor în operare);
- **proiectarea fișierelor și bazei de date** (structurarea datelor ca logică și relații, volumul și timpul de răspuns, specificațiile pentru înregistrări);
- **prelucrarea datelor** (proceduri de prelucrare, modulele de programe, cerințele de raportări și timpul de răspuns etc.);
- **definirea procedurilor manuale** în prelucrarea și urmărirea fluxurilor de date și informații (ce activități, cine le realizează, când, cum și unde);
- **definirea operațiunilor de control** pentru diferite operații:
 - de intrare (caractere, limite etc.);
 - de prelucrare (consistentă, gestiunea înregistrărilor);
 - de ieșire (totaluri, eșantioane etc.);
 - procedurale (forme speciale, cuvinte de acces etc.);
 - de evaluare a performanțelor în raport cu anumite standarde
- **definirea operațiunilor de securitate**, privind controalele de acces, planurile pentru situații de excepție, auditul de urmărire.
- **documentarea** pentru operarea sistemului și pentru utilizatori.
- **conversia** la noul sistem, care referă la transferul de fișiere, inițierea și definirea unor proceduri noi, selectarea metodelor de testare pentru noul sistem, modul de trecere efectivă la utilizarea noului sistem.
- **instruirea**, care presupune pregătirea personalului care va utiliza noul sistem, alegerea metodei de instruire, realizarea modulelor de instruire și definirea facilităților care vor fi utilizate în instruire.
- **schimbările organizatorice** presupuse de implementarea noului sistem, prin proiectarea fișelor posturilor de lucru (job design), a activităților, proceselor și structurii organizatorice (compartimente și conexiunile dintre ele), precum și toate corelațiile presupuse de această structură.

Din prezentarea acestor cerințe și specificații de proiectare a SI, se poate vedea că proiectarea în sine se poate realiza pe două planuri: unul logic și altul fizic.

Proiectarea logică constă în componentele sistemului și a corelațiilor dintre ele, așa cum acestea apar utilizatorului. Prin proiectarea logică se prezintă soluțiile pentru realizarea SI, independent de implementarea lor fizică curentă. În proiectarea logică se descriu intrările și ieșirile, funcțiile de prelucrare, procedurile, modelele de date, controalele din sistem, după specificațiile prezentate mai sus.

Proiectarea fizică constă în translatarea componentelor logice, formale, a modelelor, în specificații tehnice ale sistemului pentru un anumit hardware și software sau SGBD disponibile, pentru anumiți suporti (media) de intrare și ieșire în/din sistem, ținând seama de procedurile manuale și controalele specificate. Proiectarea fizică finalizează modul de funcționare al sistemului, cu oameni și echipamente.

Ca și în construcția unei case, proiectarea SI poate oferi mai multe soluții, urmărindu-se ușurința și eficiența realizării și implementării cerințelor utilizatorului, în cadrul unor restricții de ordin tehnic, organizatoric sau financiar. În procesul de proiectare implicarea utilizatorului este esențială, în confirmarea cerințelor informaționale și a priorităților din organizație, realizându-se în acest mod înțelegerea și acceptanța lui pentru noul sistem. Gradul de implicare al utilizatorului depinde și de modul în care este abordată realizarea SI, așa cum se va arata într-un capitol următor.

Analiza și proiectarea SI, așa cum au fost prezentate, reprezintă etapele cele mai importante și de durată, însă care se concretizează, până la obținerea unui SI operațional, cu încă trei etape: programarea, testarea și conversia, urmate de etapa de exploatare și întreținere curentă.

3. Programarea reprezintă procesul prin care specificațiile rezultate din proiectare sunt transpuse în programe, pe infrastructura hardware necesară realizării SI.

În această etapă, cerințele rezultate din analiză și modulele funcționale generate de aceste cerințe în faza de proiectare, se vor regăsi în module - program realizate în faza de programare, așa cum se arată în Figura 2.

Modulele – program proiectate, conțin descrieri ale logicii de prelucrare necesare pentru a implementa un anumit modul – funcțional, care cuprinde:

- algoritmii, în detaliu, folosiți în prelucrare;
- structura și reprezentarea datelor;
- relațiile între funcțiile realizate și datele utilizate.
-

Fiecare modul poate fi codificat ca procedură, subrutină, funcție sau alt tip de obiect identificabil.

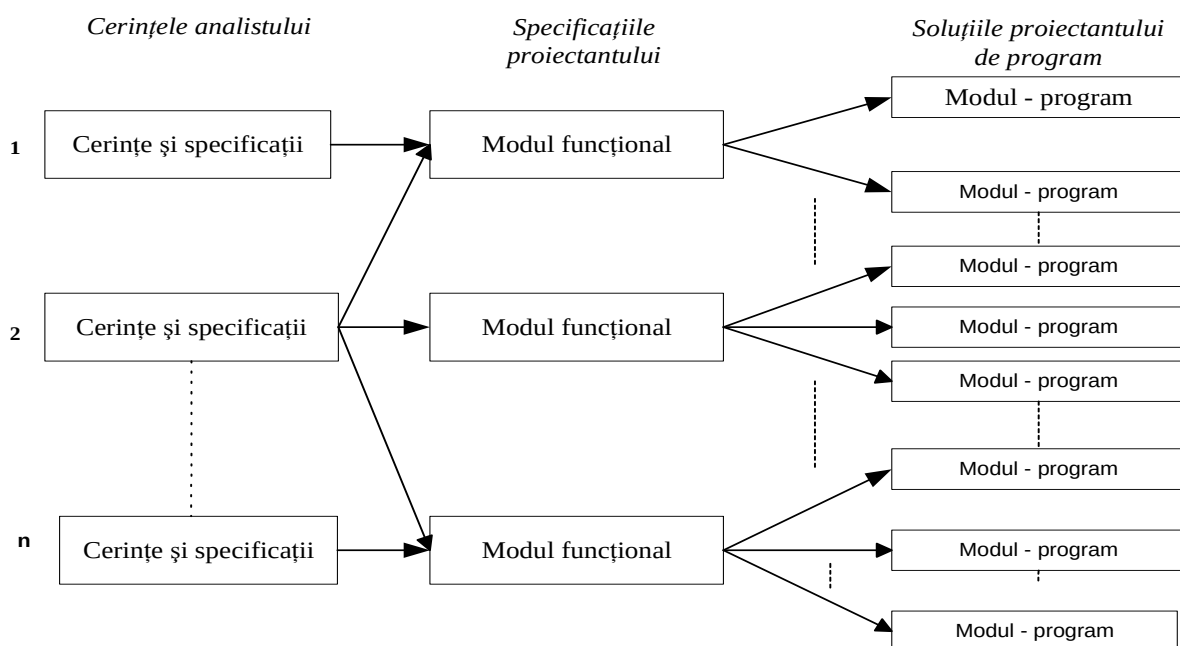


Figura 2. Corelarea cerințelor cu specificațiile de proiectare și elaborarea programelor.

4. Testarea se referă la etapa în care se verifică dacă sistemul realizează obiectivele (cerințele) impuse în etapa de analiză. În economia de timp a programului, testarea ocupă o parte foarte importantă (până la 50 %). Din această etapă pot rezulta cerințe de reproiectare a anumitor module. Testarea unui SI se poate realiza prin următoarele tipuri de activități, în special pentru sistemele mari, complexe (Figura 3):

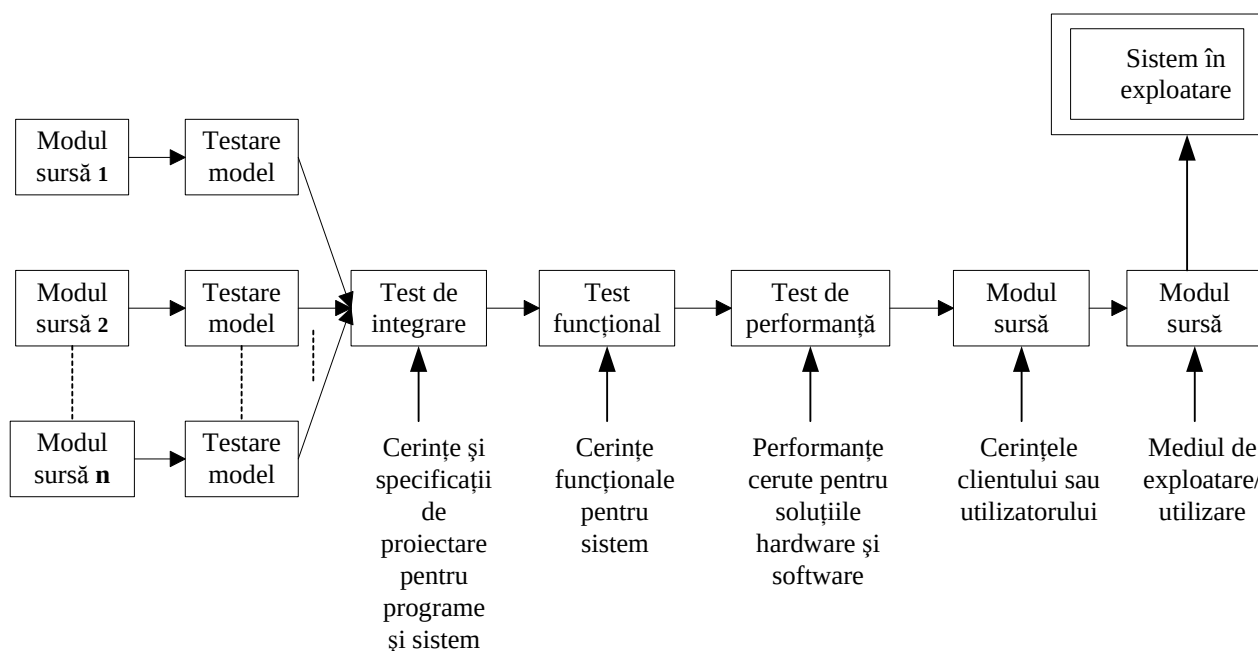


Figura 3. Fazele procesului de testare în realizarea unui SI complex

- ▶ **testarea modulelor** (a „imitațiilor de program”), în engleză „unit testing”, constă în testarea separată a fiecărui program, separat, în primul rând pentru a depista erorile care, chiar dacă sunt corectate, mai pot apărea la integrarea programului în sistem.
- ▶ **testarea integrării**, cu scopul verificării interfețelor între modulele sau unitățile de program și ca fiecare modul / componentă a sistemului funcționează în corelare cu celelalte, conform specificațiilor rezultate din proiectarea sistemului.
- ▶ **testarea cerințelor funcționale** pentru sistem, conform cu cerințele și specificațiile impuse la proiectarea sistemului.
- ▶ **testarea performanțelor**, se referă în principal la evaluarea soluțiilor software și hardware: performanțe de timp, capacitatea de a prelua sarcini la încărcării maxime (ca memorie și prelucrare), capacitatea de recuperare și restartare, după întreruperi datorate unor cauze diferite, execuția procedurilor manuale.
- ▶ **testarea de acceptare** reprezintă certificarea finală a sistemului, de către client/utilizatorul final, conform cerințelor exprimate de acesta în faza de analiză și proiectare conform unui plan de testare realizat de client și proiectant. Această acceptare este realizată de personalul tehnic al beneficiarului, de utilizatorii diferitelor structurii și de management, care a elaborat planul de testare. După acceptare, sistemul poate fi instalat și pus în exploatare.

5. **Conversia**, sau instalarea noului sistem în locul celui vechi, se poate realiza considerând mai multe strategii, în funcție de condițiile de instalare și de mediul de utilizare:

- **Strategia funcționării în paralel**, care presupune exploatarea ambelor sisteme, în paralel, până va exista siguranța că noul sistem va funcționa corect. Această abordare este mai sigură (asigurând un back-up în caz de erori sau căderi ale noului sistem), însă este foarte scumpă (resurse și personal în plus).
- **Strategia de înlocuire directă** a vechiului sistem cu cel nou, începând cu o dată fixată dinainte. Această abordare este mai ieftină, însă presupune riscuri mari, dacă noul sistem are probleme mari în exploatare, care pot cauza pagube mari.
- **Strategia de instalare prin etape pilot**, în care noul sistem se introduce treptat, pe departamente sau funcțiuni separate, prin studii/faze **pilot care, dacă se confirmă, se trece la alte** părți/funcții din organizație, fie simultan, fie etapizat.
- **Strategia introducerii pe faze**, la nivelul unităților/componentelor organizaționale sau al unor funcții la nivelul întregii organizații până la cuprinderea întregii organizații și a funcției ei.

În oricare din strategii, se deduce ușor ca este necesar un **plan de conversie**, care trebuie să cuprindă eşalonarea în timp a tuturor activităților care conduc la instalarea noului sistem. De asemenea, este foarte importantă elaborarea unei **documentații**, care să cuprindă modul în care noul sistem este utilizat, atât de către personalul tehnic și de către utilizatorii obișnuiți, sau de către managementul la diferite niveluri.

6. Operarea și exploatarea, întreținerea și dezvoltarea sistemului, reprezintă etapa care urmează conversiei și instalării noului sistem, după trecerea testului de instalare, marcând momentul punerii în funcțiune al SI, în vederea operării și exploatarei curente. Întreținerea se realizează prin menținerea în parametri proiectați și testați la instalare, în realizarea modificării sau îmbunătățirii/dezvoltării, în hardware, software, proceduri și documentație în vederea eliminării unor erori, a satisfacerii unor cerințe noi sau a creșterii eficienței în exploatarea a SI.

Întreținerea este realizată de specialiștii clientului sau în regim de outsourcing. În acest fel, se poate spune că asigurarea calității în realizarea unui SI se realizează prin implicarea clientului, a constructorului SI și a celor care asigură întreținerea așa cum se arată în Figura 4.

Clienții și potențialii utilizatori/beneficiari ai SI bazat pe noile TIC își formulează problemele care le au și care pot fi rezolvate prin TIC prin activități proprii, însă cel mai frecvent prin consultanță, pe baza contractuală cu organizații specializate, care pot realiza și studii de fezabilitate pentru SI. Clieții, după aceea, intră în relații contractuale pentru realizarea SI și rezolvarea problemelor cu organizații care dispun de competența realizării SI (analiză, proiectare, inginerie software, implementare).

După implementare, odată depășită faza de întreținere în perioada de garanție, se trece la contractarea **întreținerii și dezvoltării**, cu organizații care pot fi altele decât cea care a realizat SI, în condiții de contractare competitivă, ținând seama de costurile presupuse de asigurarea unei funcționări bune și operative a SI, cu păstrarea și eventual dezvoltarea performanțelor.

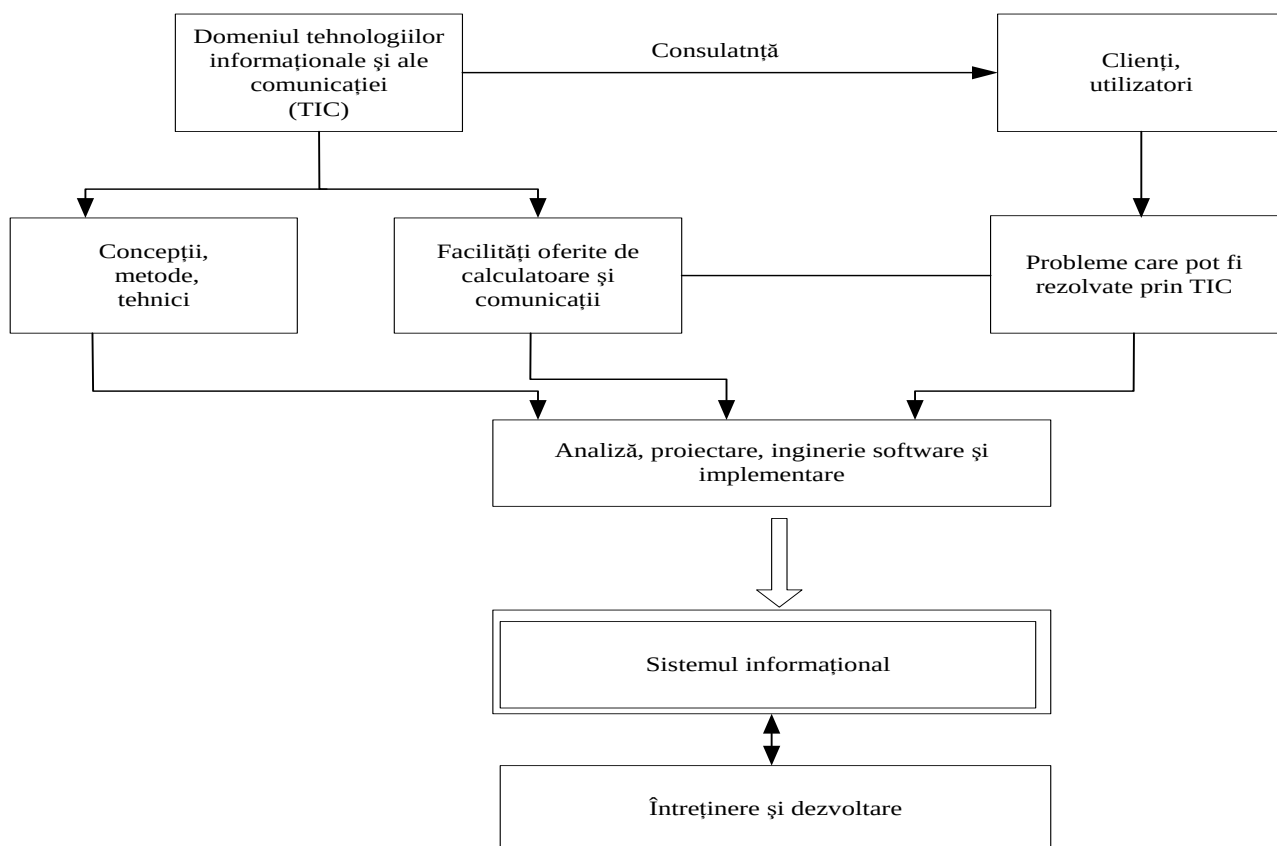


Figura 4. Relația între client/utilizator și domeniile care contribuie la realizarea SI

În realizarea SI se observă conlucrarea mai multor categorii de profesioniști care trebuie să asigure o coerență în realizarea diferitelor etape așa cum se arată în Figura 5.

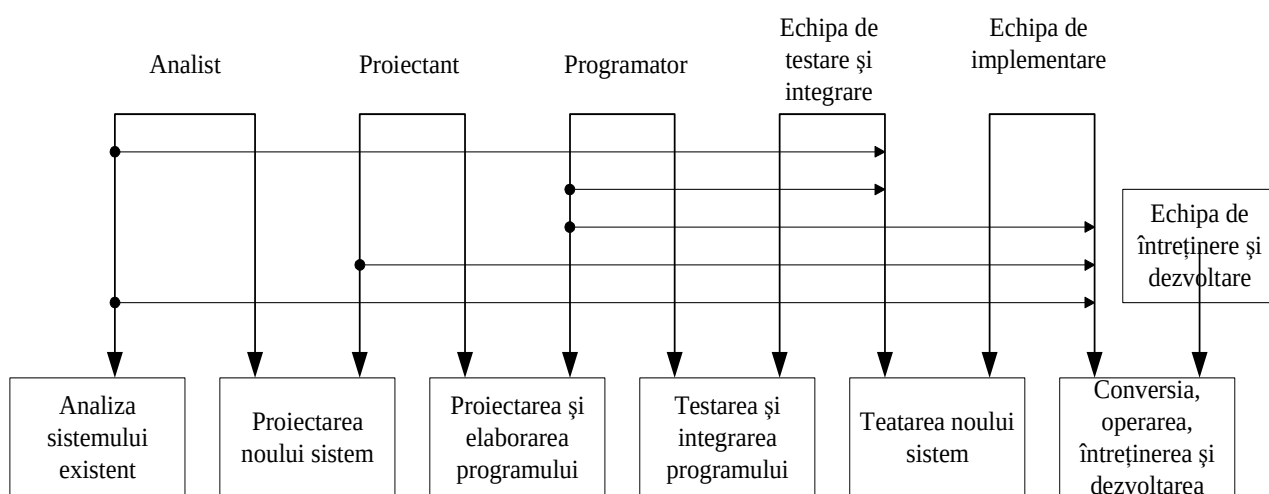


Figura 5. Corelația între diferite categorii de specialiști în realizarea SI

În primul rând, se implică **analistii**, care pot acorda consultanță, la început pentru identificarea problemelor și pentru realizarea unui studiu de fezabilitate al noului SI. Odată luată decizia de realizare a SI, analistii se implică în realizarea primei etape, cu responsabilități în urmărirea realizării cerințelor SI în faza de proiectare, în testarea de acceptare și în conversie. În continuare, sunt implicați **proiectanții**, în proiectarea noului sistem și în urmărirea fazei de programare și celei de implementare. **Programatorii**, în afara proiectării și elaborării programelor, urmăresc **testarea programelor**, integrarea lor și **testarea noului sistem**. În echipa de implementare un rol important îl au **instructorii**, responsabili cu pregătirea utilizatorilor finali în operarea și exploatarea eficientă a sistemului. În final, după conversie, intervine **echipa de întreținere**, în responsabilitatea clientului sau prin contract de outsourcing.

3. Realizarea S.I. cu parcurgerea tuturor etapelor ciclului de viață

În capitolul 6, unde realizarea S.I. a fost abordată ca mod de a planifica schimbări într-o organizație, ne-am referit la șase etape necesare dezvoltării de ansamblu a S.I. (Figura 6).

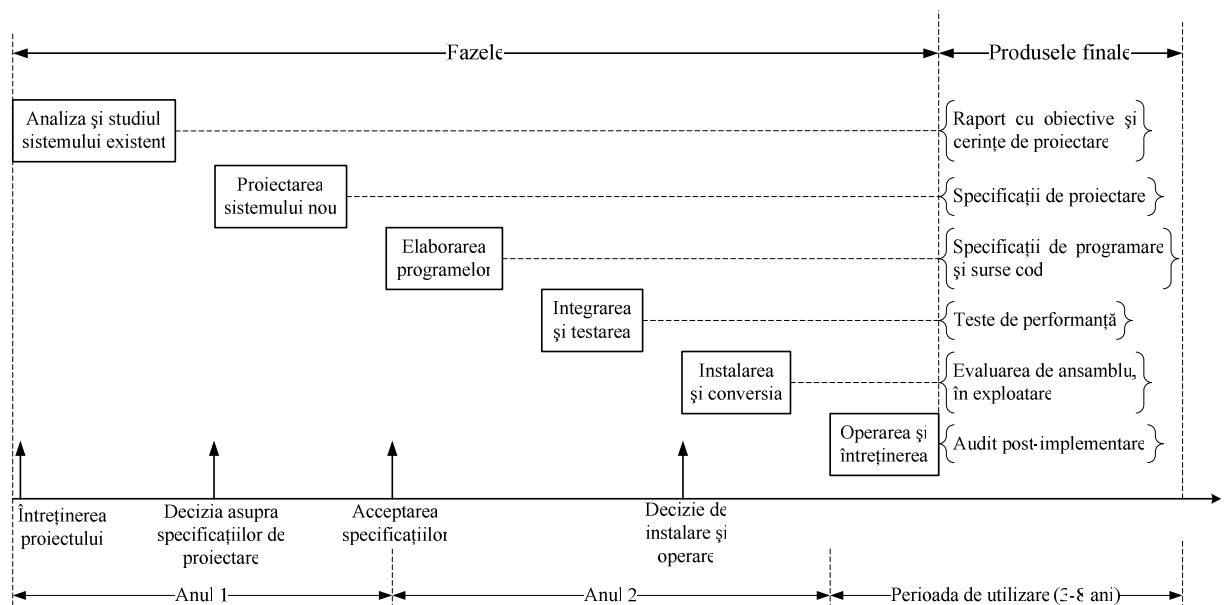


Figura 6. Realizarea S.I. prin parcurgerea tuturor etapelor ciclului de viață

Fiecare din aceste etape formale din ciclul de viață al S.I., au un specific și o finalitate, care se urmărește în timp, marcând reperele temporale și de conținut ale execuției S.I. Prin acest sistem se ajunge la un timp mediu de realizare a proiectului de S.I. de cca. 2 ani și cu o durată a vieții lui de cca. 3 – 8 ani [2]. Etapele care se parcurg sunt, în general următoarele:

1. Analiza sistemului existent, cu scopul **definirii proiectului** care se urmărește și a unui **studiu** prin care se arată transformările care trebuie făcute în sistemul existent. În această etapă se scot în evidență problemele care trebuie rezolvate și modul în care ele pot fi rezolvate prin noul sistem, sau prin schimbarea celui existent (rezultă din studiu, care trebuie să ofere alternative). Finalitatea acestei etape se realizează prin:

- definirea obiectivelor sistemului;
- variante de soluții la problemele existente;
- propunerea unui plan de realizare și a cerințelor de proiectare.

2. Proiectarea noului sistem, în care se realizează specificațiile de proiectare logică și fizică, prin care se rezolvă problemele sistemului existent, folosind metodologiile și instrumentele/tehnologiile prezentate.

3. Elaborarea programelor, în care specificațiile de proiectare sunt transpuse în programe sursă (cod), prin tehnologiile specifice în software engineering.

4. Integrarea și testarea modulelor realizate în faza de programare, prin realizarea unor teste de performanță.

5. Instalarea și conversia, constă în punerea în funcțiune și operarea noului sistem sau a celui dezvoltat, odată cu instruirea personalului și trecerea de la vechiul la noul sistem în utilizarea de zi cu zi, cu o primă evaluare a performanțelor.

6. Operarea și întreținerea, constă în utilizarea noului sistem în care se scot în evidență eventuale anomalii, iar în cadrul unei activități de audit se poate evalua modul în care noul sistem îndeplinește obiectivele stabilite în faza de analiză și studiu. În timp, anomaliile pot fi corectate și consolidate în versiuni noi ale programelor aplicative și de sistem. Atunci când sistemul ajunge la sfârșitul ciclului de viață, se poate trece la realizarea unui nou sistem cu un nou ciclu de viață.

Acest mod de realizare a S.I. este utilizat de obicei atunci când avem de-a face cu aplicații și sisteme complexe în care cerințele sunt bine structurate și definite. În aceste cazuri se impune o analiză riguroasă a cerințelor, a specificațiilor precum și un control riguros al realizării sistemului. În această abordare costurile sunt ridicate, durata execuției este mare, iar realizarea în sine greoaie, inflexibilă, prin faptul că orice revenire asupra cerințelor și specificațiilor inițiale este greu de acceptat.

5. Alte tipuri de abordări în realizarea S.I.

Alte abordări în realizarea S.I. pot diminua sau reduce dezavantajele prezentate ale abordării clasice, tradiționale. Între aceste abordări amintim pe cele mai cunoscute: prototipizarea, utilizarea pachetelor standard, dezvoltări proprii ale clientului și outsourcing-ul

5.1. Prototipizarea

Prototipizarea se poate defini prin realizarea unui sistem experimental (prototip), cu costuri reduse și într-un timp scurt pentru a convinge clientul/beneficiarul asupra avantajelor S.I. Prototipul este o versiune model a sistemului care trebuie implementat, prin dezvoltări succesive, până la momentul în care beneficiarul consideră că sunt realizate cerințele. În această abordare procesul de realizare este iterativ, repetitiv, acceptând modificări în proiectare, până se ajunge la atingerea obiectivelor urmărite. Așa cum se arată în Figura 7, pașii făcuți în abordarea de tip prototipizare sunt:

- **Identificarea cerințelor** de bază pentru aplicații sau sistem.
- **Realizarea unui prototip** de lucru, prin care se pot verifica cerințele de bază formulate de beneficiar.

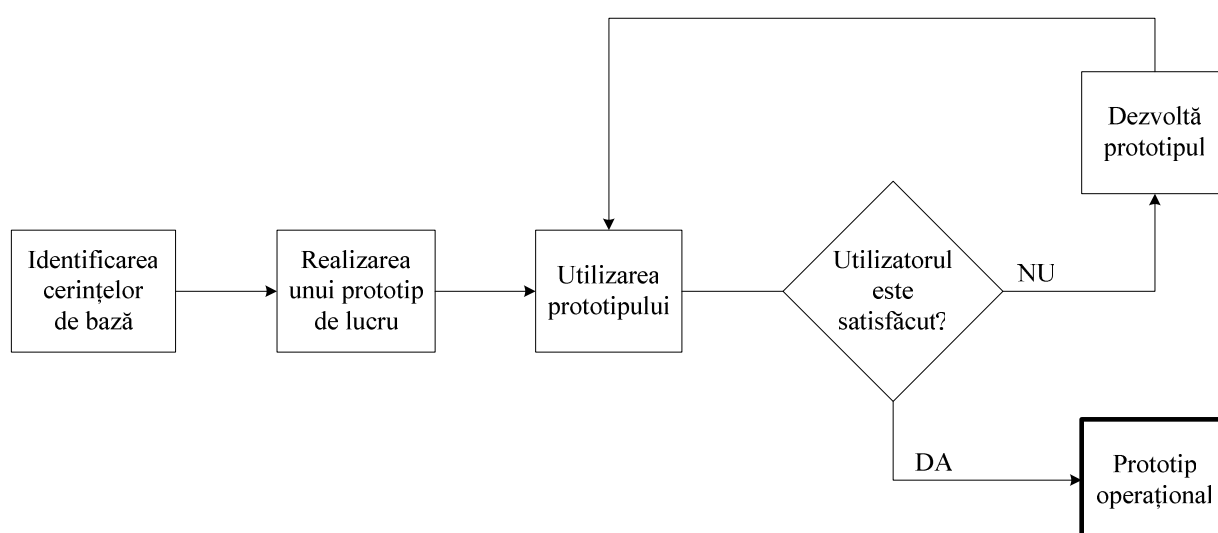


Figura 7. Procesul de realizare a S.I. prin prototipizare

În această fază se pot folosi metodologii și instrumente CASE (Computer Asisted Software Engineering) [3].

- **Utilizarea prototipului** cu scopul obținerii unor confirmări din partea beneficiarului de sistem sau a propunerii unor modificări/dezvoltări.
- **Dezvoltarea prototipului** prin rafinare și revenire la pasul precedent, până la obținerea prototipului operațional, acceptat de beneficiar.

Această abordare prin prototipizare se pretează cel mai bine pentru aplicații și sisteme de mici dimensiuni sau atunci când nu se pot defini de la început foarte precis cerințele sistemului sau specificațiile de proiectare. De asemenea, prototipizarea este adecvată proiectării sistemelor orientate pe interfață – utilizator (end – user – interface) de tip interactiv.

5.2. Utilizarea pachetelor de programe aplicative standard

Pachetele de programe aplicative standard, reprezintă o strategie alternativă în realizarea S.I., prin achiziționarea acestor produse – program, care sunt disponibile pentru vânzare sau închiriere pe piață. Aceste programe pot fi de mici dimensiuni, dar și complexe, cu multe sute de mii de linii de cod care se implementează pe servere de aplicații sau de baze de date. Această abordare s-a extins foarte mult, datorită aplicațiilor comune, utilizate în toate organizațiile (stocuri, salarii, contabilitate etc.) acoperind funcții și proceduri standard, multe rezultând și din cadrul legislativ specific mediului de afaceri. Având pachete de programe gata proiectate, cu cod integrat și testat, intervalul de timp (necesar) pentru implementare și utilizare efectivă se reduce foarte mult. Furnizorii se ocupă mai mult de întreținerea și asistența în utilizare, eventual în dezvoltarea aplicațiilor și mai puțin în analiza și proiectarea lor.

În realizarea S.I. complexe, cu cerințe speciale care nu se regăsesc în funcțiile disponibile în pachetele standard, trebuie făcute adaptări la cerințele clientului (**customizări**). Din această cauză, furnizorii de pachete prevăd facilități de personalizare/customizare, prin care se pot realiza modificări în pachete, fără a distruge integritatea pachetului. În alte situații se impune

elaborarea unor programe de adaptare și interfațare, așa cum s-a prezentat în capitolul 5. În oricare din situații, costul implementării crește foarte mult, cu procentul de modificare a codului sursă, așa cum se arată în Figura 8.

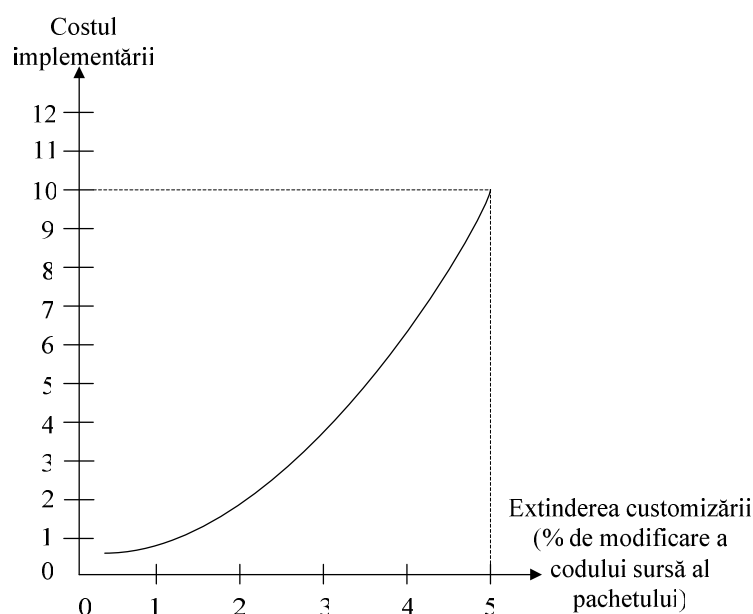


Figura 8. Costul implementării prin creșterea modificării sursei de cod a pachetului

Opțiunea pentru un pachet de programe standard se face pe baza unei selecții, care presupune evaluarea ofertelor existente pe piață, care se bazează pe mai multe criterii: funcțiile realizate, flexibilitatea, interfața – utilizator, resursele hardware și software cerute ca suport, cerințele pentru baza de date necesară, efortul de instalare și întreținere, documentația, calitatea și cotarea furnizorului, costul. Pentru evaluare este nevoie de realizarea unei **cereri de ofertă** (RFP – Request for Proposal), care cuprinde o listă de întrebări referitoare la criteriile prezentate la care furnizorul trebuie să dea răspunsuri. Procesul de achiziție a unui pachet de programe constă în parcurgerea a cinci etape:

- **Definirea cerințelor aplicației/sistemului**, cu estimarea volumului de tranzacții necesar pe termen scurt și mediu, precum și cu necesitatea interfațării cu alte aplicații din sistem, în vederea integrării, conform fazei de analiza. Pe baza acestei definiri a cerințelor se face **cererea de ofertă** (RFP).

- **Identificarea furnizorilor potențiali**, ca furnizori de software (software houses) în exclusivitate sau ca furnizori de soluții, care își asumă întreaga responsabilitate pentru implementare (adaptare, instalare, instruire, întreținere). În alegerea furnizorului trebuie abordat și efortul de proiectare a adaptărilor necesare pentru satisfacerea cerințelor: fie prin adaptarea pachetului la cerințe, fie prin adaptarea cerințelor la facilitățile oferite de pachet, ceea ce presupune adaptarea organizației la procedurile impuse de software.

- **Evaluarea alternativelor**, prin determinarea gradului de satisfacere a RFP, estimat prin punctaje. Se poate recurge și la încercarea pachetului (benchmark test), prin măsurarea timpului necesar în execuția unui număr de tranzacții (tpm – tranzacții pe minut) și a costului pe tranzacție (€/tranzacție).

- **Cumpărarea pachetului**, ca **licență software de utilizare**, deoarece prin cumpărare nu devii proprietarul acelui pachet (ca sursă și drept de utilizare nelimitat). Procedura de licențiere se negociază cu furnizorul, pe baza unor elemente de evaluare furnizate de cumpărător (numărul de utilizatori la stații de lucru, configurația serverului de aplicație etc.).

- **Instalarea pachetului** pe echipamentele existente sau pe unele noi (în completare/extindere) cu punerea în funcțiune și operarea la nivel de utilizator final.

5.3 Realizarea S.I. cu resurse proprii utilizatorului

În multe organizații, în special cu mai mulți ani în urmă, realizarea S.I. se făcea cu resursele proprii utilizatorului, cu asistență tehnică redusă, minimă, din partea firmelor specializate. Utilizatorii pot avea însă la dispoziție o serie de metode, tehnici și instrumente care să le permită să obțină mai repede decât tradițional realizarea aplicațiilor și sistemului dorit.

Dezvoltările bazate pe resurse proprii utilizatorului au o serie de avantaje:

- definirea mai precisă și înțelegerea mai exactă a cerințelor, fiind legate de propriile necesități;

- implicarea mai mare a beneficiarului și creșterea satisfacției în utilizarea propriei dezvoltări;
- reducerea gradului de corecții, prin reveniri pentru precizarea cerințelor.

În schimb, pot apărea și o serie de dificultăți, datorită neasigurării unor standarde de calitate adecvate, iar testările și documentația pot fi necorespunzătoare. Uneori se poate pierde controlul asupra datelor, prin dificultatea determinării localizării datelor pentru a asigura că aceeași dată/informație este utilizată consistent în toată organizația.

Managementul dezvoltării sistemului de către utilizator presupune crearea unui **centru informațional** (CI), prin care se realizează instruirea și suportul tehnic necesar al utilizatorilor aplicațiilor sau sistemului. CI poate dispune de o serie de instrumente, de consultanți, prin care se asigură o creștere a productivității, a reducerii paralelismelor în realizarea de aplicații (printr-o viziune de ansamblu, pe organizație), asigurând integritatea sistemului. Prin CI se poate asigura un standard de calitate, un control al costurilor și o justificare a lor.

5.4 Realizarea S.I. prin servicii externalizate (outsourcing)

Dacă o organizație nu este dispusă să utilizeze resurse proprii pentru realizarea și operarea SI, atunci se poate angaja/închiria organizații specializate în realizarea serviciilor presupuse de realizarea și utilizarea SI. Această operație de transferarea a responsabilităților realizării și operării SI unui furnizor extern se numește **externalizare** sau **outsourcing**, ea fiind motivată de creșterea costurilor pentru servicii similare realizate de utilizator. Acest concept a apărut și s-a dezvoltat odată cu creșterea costurilor de menținere a unor resurse umane cu înalt nivel profesional cerute de realizarea și operarea aplicațiilor și SI. Furnizorii de servicii externalizate beneficiază de faptul că utilizează resursele de care dispun din plin, partajând aceste resurse pentru mai mulți clienți. Un beneficiar de servicii, plătește numai acele servicii de care are nevoie, în anumite intervale de timp, în raport cu cerințele și nu permanent, dacă ar

dispune tot timpul de propriile resurse, care ajung astfel să nu fie utilizate din plin. Menținerea nivelului profesional al specialiștilor în TIC în condițiile evoluției lor rapide, au început să coste foarte mult, impunând și din acest motiv externalizarea.

Prin externalizare, organizația poate pierde controlul asupra funcțiilor SI, de aceea beneficiarul trebuie să dispună de expertiză în negocierea condițiilor de externalizare. În ciuda multor dificultăți în acceptarea externalizării, această opțiune oferă însă o serie de avantaje, în final:

- **reducerea costurilor de operare** a SI, analizând cu profunzime ce aplicații se externalizează, menținând controlul asupra unor activități și procese critice pentru organizație;
- **asigurarea accesului organizației la servicii bazate pe tehnologii avansate**, la zi, care ar fi foarte costisitoare și incomplet utilizate, prin utilizarea lor în exclusivitate;
- **asigurarea creșterii contribuției TIC la performanța organizației**, deoarece furnizorii de servicii externalizate sunt profesioniști, cu o expertiză deosebită în domeniu;
- **obținerea unor surse suplimentare pentru beneficii și profit**, prin reducerea cheltuielilor proprii și obținerea unor servicii bazate pe TIC de înalt nivel.

Serviciile externalizate așa conduc la succes atunci când se bazează pe încrederea reciprocă a partenerilor, pe o structurarea a relațiilor astfel încât să fie în echilibru bunele sentimente și controlul propriu, practicat de beneficiar. Furnizorii de servicii externalizate se orientează pe elaborarea de soluții tipizate pe diferite domenii, în cadrul unor **centre de soluții**, în care echipe de experți elaborează soluții la probleme, modele, arhitecturi, etc.

În sinteză, o comparație între diferitele abordări prezentate în realizarea SI se prezintă în Tabelul următor:

Abordarea	Caracteristici de bază	Avantaje	Dezavantaje
1. Clasică (cu ciclul de viață complet) pentru S.I.	• Specificații scrise și aprobate; • Realizare în etape succesive; • Rol limitat pentru beneficiar în realizarea SI	Abordarea este necesară în realizarea de aplicații și sisteme complexe	• Realizare de durată cu costuri mari • Nu stimulează realizarea schimburilor • Documentația voluminoasă
2. Prototipizare (proiecte pilot)	• Specificațiile au natură dinamică, elaborate și definitive în timp, pe baza experimentală • Realizare iterativă, informală, rapidă, cu utilizatorii implicați permanent	• Execuție rapidă și necostisitoare • Se recomandă atunci când specificațiile nu se pot defini clar de la început și când interfața cu utilizatorul este importantă	• Nu se folosește în cazul aplicațiilor și sistemelor complexe • Nu este riguroasă în fazele de analiză, testare și elaborarea documentația
3. Utilizarea pachetelor de programe standard	• Produse – program disponibile pe piață, care satisfac cerințele unei organizații	• Timp și resurse umane reduse pentru analiză, proiectare, programare, implementare • Reduce necesarul de resurse IT pentru aplicații și sistem	• Poate satisface numai parțial cerințele organizației • Necesitatea unor adaptări (customizări) • Adaptări extinse pot majora mult costurile
4. Realizarea SI cu resurse proprii	• Realizarea SI cu resurse proprii, folosind metode, tehnici și instrumente moderne • Cerințe bine cunoscute	• Realizarea este controlată de utilizator • Timp de realizare și corecții reduse	• Pot apărea erori și necoerențe sistemice în utilizarea datelor • Nu se asigură niveluri de calitate și standarde ridicate
5. Externalizare (outsourcing)	• SI realizat și uneori, operat de un furnizor de servicii TIC exterioare organizației	• Nu presupune asistența unui personal calificat, costisitor, în organizație • Poate oferi costuri reduse, controlabile	• Dependență mare de furnizor și o cedare a unei părți a controlului pentru funcțiile S.I.

7. Bibliografie

- 1.Shari Lawrence Pfleeger : Software Engineering. Macmillan Publishing Company, New York, 2001.
- 2.Kenneth C. Laudon, Jane P. Laudon : Management Information Systems.Pretince-Hall Inc., New Jersey, 2000.
- 3.Guran Marius : Sisteme informatice – Infrastructura informationala si de comunicatii in managementul intreprinderii moderne. Editura AGIR, Bucuresti, 2008