

CONSIDERAȚII PRIVIND INGINERIA SOFTWARE ȘI MANAGEMENTUL PROIECTELOR TIC

Lect. univ. drd. Gabriel CUCUI - Universitatea "Valahia" Târgoviște, gabicucui@valahia.ro

Rezumat:

Le développement de l'ingénierie software s'est concrétisé dans une méthodologie de formalisation du processus de développement software, basée sur l'optimisation des coûts. La croissance des exigences concernant la qualité des systèmes informatiques a impliqué la nécessité de la définition et de l'emploi des stratégies de projection et de développement software bien structurées. L'ingénierie software vise tous les aspects concernant la production software à partir des premiers stages des spécifications de système et jusqu'à la maintenance du système après sa mise en fonction.

Dezvoltarea ingineriei software – un ansamblu de strategii, metodologii, metode și tehnici de proiectare și dezvoltare a sistemelor informaționale de diverse tipuri – s-a concretizat într-o metodologie de formalizare a procesului de dezvoltare software, bazată pe optimizarea costurilor. Progresul în domeniul TIC a creat noi provocări pentru ingineria software, care a înregistrat alte mutații în planul modelării sistemelor informaționale, metodologiile și tehnicile de dezvoltare trebuind să fie adaptate la parametrii noilor soluții de echipamente tehnice.

Realizarea, în paralel cu procesul de dezvoltare, a unei documentații eficiente este esențială pentru succesul unui produs software.

Înainte elementelor de inginerie software utilizate pentru dezvoltarea unui produs rămâne, însă, ca factor de succes, calitatea managementului proiectului informatic. Un proces de dezvoltare software complex înregistrează costuri tot mai mari, ceea ce impune realizarea unor activități de analiză și de proiectare și a metodologiilor de dezvoltare vizând diminuare riscurilor asociate strategiilor ingineriei software stabilite.

Creșterea exigențelor cu privire la calitatea sistemelor informatice, importanța rolului pe care acestea îl joacă în ameliorarea performanțelor unei organizații au condus la necesitatea definirii și utilizării unor strategii de proiectare și dezvoltare software bine structurate, care trebuie să preceadă orice activitate privind elaborarea programelor informatice.

Valorificarea la maxim a resurselor disponibile, prin utilizarea unor tehnici de îmbunătățire a calității produselor informatice și pentru minimizarea efortului depus în dezvoltarea lor sunt apanaje ale ingineriei software. Acest concept pornește de la premisa că o abordare metodologică a dezvoltării sistemelor informatice conduce la un timp de livrare mai scurt și la o un grad mai mare de atingerea a obiectivelor vizate pentru sistemele dezvoltate.

Ingineria software urmărește toate aspectele privind producția software de la primele stagii ale specificațiilor de sistem și până la mentenanța sistemului după ce a fost dat în funcțiune. Ținând cont de toate cerințele organizaționale și de restricțiile financiare ale unei organizații, ingineria software definește teorii, metode și instrumente de lucru al căror număr sporește continuu și care devin tot mai complexe datorită încercărilor de a descoperi noi soluții la problemele specifice unor organizații.

Alături de o abordare riguroasă, metodologică, uneori ingineria software recurge la o *abordarea informală* de dezvoltare a sistemelor, care s-a dovedit a fi mult mai eficientă în anumite contexte.

Ciclul de viață al unui sistem informatic

La modul general, se poate considera că o organizație este un sistem deschis, care interacționează cu mediul înconjurător. Dat fiind că organizația trebuie să-și adapteze structura și/sau comportamentul la schimbări ce intervin în mediul în care evoluează, ea se poate reprezenta printr-un sistem care este și adaptiv. Acest sistem poate face obiectul unor studii probabilistice vizând previzionarea stărilor viitoare pornind de la cele anterioare, cu un anumit grad de incertitudine, sau poate fi un sistem determinist, pentru care, cunoscând intrările, se pot determina, fără eroare, ieșirile.

Descrierea dualistă a unui sistem economic presupune separarea proceselor reale (materiale, fizice) de procesele de reglare (culegerea și prelucrarea informațiilor, fundamentarea și luarea deciziilor etc.).

Întregul proces al definirii necesității achiziționării, ori dezvoltării și implementării unui sistem informatic în cadrul unei organizații reprezintă *ciclul de viață al sistemului informatic*.

Etapile dezvoltării unui sistem informatic se suprapun, de regulă cu fazele ciclului său de viață, prezentate în cele ce urmează.

Analiza și evaluarea sistemului informațional existent este etapa de debut în cadrul ciclului de viață al unui sistem informațional, care constă în relevarea, cu ajutorul tehnicilor de interviuare, subsistemelor și a modulelor corespunzătoare cerințelor formulate prin strategia de guvernare TIC și a celor ineficiente, ori a necesităților nou apărute. Aceste nevoi observate pot determina fie înlocuirea totală a vechiului sistem, fie înlocuirea anumitor subsisteme ori module, combinată cu dezvoltarea altora noi.

Punctele de plecare în etapa de analiză sunt necesitățile de informare ale direcțiunii, ale departamentelor dintr-o organizație, ale personalului implicat. Se stabilesc subprobleme ale problemei de rezolvat, care impun cunoașterea în detaliu a cerințelor și restricțiilor de funcționare a sistemului, a structurii producției sau a activității organizației, a resurselor consumate de acestea reflectate în documentele utilizate în cadrul organizației, a modului de înregistrare a veniturilor și a cheltuielilor etc.

Analiza funcțională a sistemului presupune clarificări ale informațiilor privind intrările și ieșirile sistemului, procedeele de stocare și de arhivare a datelor, complexitatea sistemului, viabilitatea și costul acestuia, în timp ce analiza operațională urmărește caracterizarea detaliată a datelor de intrare (tip și dimensiune), a datelor existente în fișiere (lungime și stare a fiecăreia, tip de grupare, manieră de tratare-prelucrare), a dispunerii datelor în documentele de ieșire cu scopul pregătirii proceselor necesare pentru a le obține în maniera dorită, inspectarea tipului și a caracteristicilor perifericelor utilizate, evaluarea necesarului de memorie. De regulă, se întocmește un caiet de sarcini cuprinzând etapele de rezolvare și dezvoltare a fiecărei unități de prelucrare.

Urmează studiul pentru *definirea cerințelor* unui nou sistem, care trebuie să răspundă nevoilor de perfecționare enunțate în finalul fazei anterioare.

Proiectarea sistemului de realizat definește soluția propusă pentru sistemul informațional dorit, arhitectura sa globală, începând cu stipularea aplicațiilor utilizate pentru dezvoltare și implementare, continuând cu stabilirea structurii datelor, a arhitecturii aplicației software, a detaliilor procedurale, cu modelarea interfeței, cu programarea, cu prevederea metodelor de comunicație și a măsurilor de securitate necesare.

După ce proiectarea informațională va preciza modul de funcționare a sistemului din punct de vedere al utilizatorilor săi, în faza următoare, proiectarea informatică este detaliată pe nivelurile *conceptual*, vizând conturarea unei soluții independent de soluția de implementare și *logic*, în care se realizează adaptarea modelului proiectat la posibilitățile informatice existente, descrierea arhitecturii logice și fizice, a mediului de dezvoltare, a structurii bazelor de date, a programelor necesare prelucrărilor preconizate în sistem.

Implementarea presupune realizarea efectivă a programelor și a celorlalte elemente software necesare funcționării sistemului la parametri proiectați. În cazul în care se pune problema integrării unor părți de cod noi în ansamblul sistemului informațional, trebuie să se manifeste o atenție sporită realizării conexiunilor necesare. Orice corecție efectuată în urma scrierii și testării codului trebuie să fie raportată și să se regăsească în modelele proiectate în fazele anterioare.

Un rol important în utilizarea cu succes a noului produs revine formării personalului implicat în utilizarea sistemului informațional dezvoltat și furnizării unei documentații bine întocmite a sistemului, realizării unui manual al utilizatorului, al unui manual de exploatare.

De cele mai multe ori, implantarea nu se realizează direct, prin substituirea sistemului informațional existent, ci noul sistem este lăsat să funcționeze în paralel cu vechiul sistem informațional, conform unui calendar de implantare, până ce majoritatea inconsecvențelor sunt rezolvate de echipa de dezvoltare. Se practică și testarea retrospectivă a funcționării noului sistem, introducând date din perioade precedente, pentru a verifica dacă rezultatele obținute sunt aceleași cu cele furnizate de vechiul sistem.

Mentenanța sau întreținerea sistemului informațional nou include activități regrupate în trei etape principale:

- corectoare – tratarea erorilor detectate în timpul punerii în funcțiune a aplicației;
- adaptivă – modificarea aplicației într-un anumit punct;
- perfectivă – ameliorarea funcțiilor sistemului implantat, prin dezvoltarea unei anumite părți sau a unui anumit modul, la cererea utilizatorilor.

Pentru corectarea erorilor depistate, se va face apel la reingineria de software, dacă va fi necesară operarea unor schimbări importante vizând structura sistemului implantat.

Instrumente informatice de suport pentru activitatea de modelare a sistemelor informaționale

Evoluție TIC este resimțită astăzi aproape în toate domeniile economico – sociale. Ar fi fost nefiresc ca tocmai în activitatea complexă de dezvoltare a sistemelor informaționale să nu existe un suport informatic.

Cele mai cunoscute instrumente informatice de asistare a proceselor de dezvoltare software sunt reunite sub genericul *CASE* - Computer Aided Software Engineering. Termenul *CASE* a fost introdus pentru prima dată în anul 1987 de John Manley și a desemnat instrumentele informatice concepute pentru a fi utilizate în etapele de analiză și proiectare, oferind facilități de reprezentare grafică specifice acestor etape.

Un produs de tip *CASE* reprezintă, așadar, o colecție de metode, instrumente și procese utilizate în ingineria software asistată de calculator. Aceste produse oferă suport pentru automatizarea metodelor și tehnicilor folosite pe parcursul ciclului de dezvoltare de sisteme informatice, de aplicații, de programe. Un astfel de produs poate asista dezvoltatorii pe durata ciclului de viață al sistemului dezvoltat, începând cu managementul de proiect și terminând cu asigurarea calității.

În literatura de specialitate se recurge la clasificarea instrumentelor *CASE* în două categorii:

- instrumente *CASE* de nivel superior, utilizate în fazele de analiză și de proiectare ale procesului de dezvoltare a sistemelor (acestea asigură realizarea diagramelor, generarea formularelor și a rapoartelor etc.);
- instrumente *CASE* de nivel inferior, care permit proiectarea și realizarea sistemului vizat (*CASE* pentru implementare, verificare, stabilire a configurației etc.).

Componentele de bază ale unui instrument *CASE* sunt următoarele:

- depozitul de date (data repository) – o componentă obligatorie, care acumulează și stochează, în mod organizat, toate informațiile introduse de diferite persoane, la momente diferite de timp, care vor servi în etapele de analiză, proiectare și creare a codului; în structura depozitului de date se pot delimita depozitul de informații

(Information Repository) – conținând informațiile despre afacerea organizației și despre portofoliul său de aplicații - și dicționarul de date (Data Dictionary), care specifică numele (identificatorii) și descrierea datelor, gestionează controlul *accesului* la depozitul de informații, conține descrierile resurselor necesare prelucrărilor datelor;

- editorul de diagrame – componentă obligatorie facilitând realizarea și modificarea diagramelor specifice metodologiei pentru a fost creat instrumentul CASE respectiv;
- analizorul de structură – găsește și elimină erorile dificil de localizat după culegerea informațiilor, efectuând comparații între datele nou introduse și cele deja existente în baza de informații;
- instrumente pentru așa – numitul proces de *reverse engineering* – inginerie inversă – care asigură actualizarea diagramelor conform schimbărilor realizate în codul sursă;
- generatorul de cod, componentă care poate converti în cod diagramele realizate în faza de proiectare;
- navigatorul specializat, instrument pentru vizualizarea informațiilor unui ansamblu de entități care au o structură complexă, între care există un mare număr de relații;
- generatorul de documentație, care include modele de documente, oferind utilizatorilor posibilitatea de a-și concepe propriile documente într-o manieră flexibilă;
- generatorul de formulare și de rapoarte, menit să conceapă interfețele (interactivitatea) produsului cu utilizatorii;
- componentele de transformare, care permit trecerea de la un model sau o diagramă la alt model, respectiv la altă diagramă;
- instrumentele pentru managementul de proiect, ce oferă facilități destinate gestiunii configurației fiecărui proiect (proiectul de aplicație, codul și documentația unui sistem dezvoltat);
- instrumentele de verificare automată a aplicației.

Exemple de produse CASE

Printre cele mai cunoscute produse CASE se numără *Rational Rose*, creat pentru analiza și proiectarea orientată pe obiecte printr-o metodologie UML.

Produsele CASE UML sunt utilizate pentru realizarea unor sisteme de gestiune complexe, de mari dimensiuni. Sunt realizate cu ajutorul calculatorului subsisteme, modele de analiză, de proiectare și implementare, diagrame specifice UML etc.

System Architect este un produs de tip *little CASE*, adică acesta funcționează pe calculatoare personale și în rețea, necesită Windows ca sistem de operare și are drept scop realizarea unei interfețe grafice pentru aplicații de tip client – server.

Produsul *CASE ABC* include componente pentru analiza datelor (ABC DataAnalyzer), pentru desenarea diagramelor (ABC FlowCharter și ABC SnapGraphics) și pentru vizualizare (ABC Viewer).

WESTMOUNT I-CASE asigură suport pentru analiză, proiectare a arhitecturii, modelare a sistemelor – globală și detaliată - și implementare a acestora.

WIZDOM PRO este un instrument CASE creat pentru dezvoltarea aplicațiilor în ideea simplificării acestui proces complex și a activităților de mentenanță.

Un alt instrument - *Easy CASE* – tot de tip *little CASE* – asigura implementarea următoarelor tehnici de analiză și proiectare: analiza și proiectarea structurată, modelarea prelucrărilor, a datelor și a informațiilor.

Power Designer permite definirea modelelor conceptuale și a celor fizice (entități, asocieri, reguli de gestiune, prelucrări, restricții de integritate, verificări ale integrității datelor, tratări de erori etc.).

BIBLIOGRAFIE:

1. Haussaire, A., Vial, S., Bertrand, E. – *Informatique de gestion*, Dunod, Paris, 2001.
2. Kanter, J. – *Managing with Information*, Prentice-Hall, New Jersey, 1992.
3. Kroenke, D.M. – *Management Information System*, Mitchell McGraw-Hill, 1992.
4. Mallach, E.G. – *Decision Support and Data WareHouse Systems*, Irwin McGraw-Hill, USA, 2000.
5. Oussalah, C. – *Ingénierie objet, techniques et concepts*, Dunod, Paris, 1997.
6. * * * *Dicționar de informatică*, Larousse, ediția în limba română, Editura Niculescu, București, 1999.