

Elemente actuale privind dezvoltarea de produse software

Lect.dr. Cornelia BOTEZATU, lect.dr. Cezar BOTEZATU
Universitatea Româno-Americană, București

In an ever changing world such as ours it is only natural that software developers are forced to always use and implement new instruments and platforms, new technologies in order to meet the quality, security and availability requirements of the market. The paper is meant to be a brief introduction into software development technologies, namely CASE instruments, Microsoft Office Visio, IBM Rational Unified Process (RUP), Designer 2000.

Keywords: CASE (Computer Aided Systems Engineering) Instruments, Designer 2000, Microsoft Visio, IBM Rational Unified Process (RUP), use cases, UML.

A sistam la o evoluție de neimaginat a calculatoarelor electronice, limbajelor de programare și metodologiilor de modelare și, odată cu acestea, a instrumentelor de lucru adecvate. Trăim într-o eră a cererilor. Clientii vor de la dezvoltatorii de software tot mai mult: ei vor calitate înaltă pentru produsele oferite, modificări rapide ale acestora, servicii personalizate și ușor de întreținut, sisteme integrate și livrări la prețuri cât mai mici. Și toate acestea imediat.

Pentru a răspunde acestor cerințe, dezvoltatorii de software solicită din ce în ce mai mult instrumentele CASE în diferite etape de proiectare și realizare a sistemelor informatice. Utilizarea instrumentelor CASE nu este acum o problemă de modă, ci de eficiență în activitatea de proiectare, realizare și/sau dezvoltare de software. Într-un domeniu dinamic și inovator apare permanentă nevoie de adaptare a proceselor de desfășurare a afacerilor și a soluțiilor care le implementează. Menținerea sistemelor existente devine din ce în ce mai costisitoare, mai riscantă și mai ineficientă.

În aceste condiții producătorii mediilor de dezvoltare conectează la mediile lor unele instrumente de modelare (CASE) iar pe de altă parte realizatorii de instrumente favorizează o legare mai strânsă a instrumentelor la mediile de dezvoltare. La toate acestea se adaugă standardizarea limbajelor de modelare, cererea tot mai mare pentru aplicații complexe și sigure, evoluția nivelului pregătirii profesionale a realizatorilor de sisteme informatice, care au contribuit și continuă să contribuie la

schimbarea rolului și a nivelului de utilizare a instrumentelor CASE în procesul dezvoltării produselor software.

Dar instrumentele CASE nu sunt și nici nu pot fi creative. Ele sunt și trebuie să fie în cel mai înalt grad degrevative. De aceea, automatizarea tuturor activităților potrivite reprezintă sarcina de bază pentru orice instrument CASE. În acest fel, instrumentele sprijină utilizatorii să se concentreze asupra activităților cu adevărat creative.

Instrumentele CASE aparute în jurul anului 1970 din necesitatea de a implica utilizarea calculatorului pentru susținerea activităților de planificare, proiectare și realizare a sistemelor informatice, au cunoscut o evoluție continuă. Proiectarea și/sau dezvoltarea aplicațiilor de dimensiuni medii sau mari, începând cu faza de analiză și terminând cu fazele de testare și întreținere a acestora nu poate fi realizată eficient fără ajutorul unui instrument CASE adecvat.

Adoptarea UML ca limbaj standard de modelare a influențat benefic evoluția instrumentelor CASE și a determinat o creștere sensibilă a interesului proiectanților pentru ele. Astăzi dezvoltarea unei aplicații de dimensiuni medii sau mari nu mai poate fi realizată eficient fără instrumente CASE adecvate.

Prin instrumente CASE înțelegem aplicațiile software care-i sprijină pe analiști, proiectanți, programatori, inclusiv personalul de testare și întreținere, să analizeze, să proiecteze, să implementeze (cel puțin parțial), să modifice (extindă), respectiv să construiască teste pentru sistemele informatice.

În ultimul timp instrumentele CASE au fost tot mai mult folosite pentru a realiza modele de întreprinderi sau activitati economice, asa încât definitia data trebuie extinsa cu aceasta precizare.

Utilizarea instrumente lor CASE îmbunătătesc considerabil calitatea procesului de realizare a sistemelor informatice prin metode bine fundamentate, asigurând: diminuarea complexitatii logicii de descriere a sistemului, definirea riguroasa a obiectivelor, usurinta efectuării schimbarilor, identificarea variantei optime de proiectare, cresterea vitezei de realizare a sistemelor, a disciplinei de proiectare si a integrării, posibilitatea folosirii depozitelor modularizate, constituirea bibliotecilor de module de documente, realizarea interfetei pentru generarea de coduri, realizarea unei documentatii flexibile si dinamice, realizarea succesiva a componentelor unui sistem.

Ca urmare, creste viteza de proiectare si realizare a acestora, se îmbunătătesc calitatea testării lor, a documentatiei elaborate, se promoveaza ideea de reutilizare a unor componente si se simplifica întretinerea programelor, se îmbunătătesc managementul proiectelor si conduce spre standardizarea procesului de dezvoltare a sistemelor.

☑ *Designer 2000* (produs al firmei Oracle) este destinat analistilor de sistem, programatorilor si managerilor de proiect. El contine un set puternic si integrat de instrumente CASE, având ca obiectiv principal asistarea factorului uman în proiectarea si realizarea soft-ului cu ajutorul calculatorului. *Designer 2000* marcheaza trecerea de la tipul de analiza de sistem artizanal la standardizarea acestei activitati, cu influenta directa chiar din faza de definire a cerintelor, pâna la întretinerea fizica a produsului informatic, sustinând deci toate fazele proiectului: planificare, definire, proiectare, realizare.

☑ *Microsoft Visio Professional 2002* furnizeaza solutii de tip diagrame pentru documentarea si comunicarea unei palete largi de idei, informatii si sisteme. Diagramele create în *Microsoft Visio Professional 2002* aduc plusul de valoare necesar în sistemele existente si ajuta profesionistii la crearea altora noi. Când se folosesc pentru însotirea textelor

si a cifrelor, diagramele *Visio* fac ca mesajul sa fie mai concis, ajuta la retinerea cu usurinta a acestuia si permite depasirea obstacolelor tehnice si culturale.

☑ *Microsoft Office Visio 2003* introduce un set de diagrame si scheme care fac din acest program un adevarat „erudit” dedicat dezvoltării unei documentatii vizuale specifice celor mai diverse domenii. De asemenea, dezvoltatorii de software, site-uri Web si baze de date, precum si administratorii de retea au la dispozitie, prin intermediul diagramelor specializate oferite de *Visio*, un inteligent asistent dedicat automatizării activitatilor specifice. Se poate spune ca diagramele realizate prin *Visio* aduc o noua dimensiune informatiei din documentele Office, oferind utilizatorului posibilitatea de a transmite rapid idei si concepte într-un limbaj vizual.

☑ *IBM Rational Unified Process (RUP)* este o conceptie de dezvoltare software, gândita, dezvoltata, livrata si întretinuta ca produs software. RUP este creat si documentat folosind UML si este livrat on-line utilizând tehnologia Web, fiind la dispozitia utilizatorului. Se fac upgrade-uri în mod regulat de catre Rational Software la aproximativ 2 ani, astfel ca este mereu nou. Fiind modular si în format electronic, el poate fi configurat si personalizat pentru a raspunde cerintelor specifice fiecarei organizatii. Este integrat cu foarte multe unelte de dezvoltare software în suitele Rational, astfel încât dezvoltatorii pot sa acceseze fie asistenta proceselor în cadrul unei tei pe care o folosesc, fie sa acorde o asistenta a proceselor, consistenta si personalizata pentru echipa de realizare. RUP include multe din practicile cele mai bune ale dezvoltării de software modern, într-o forma potrivita, pentru o varietate larga de proiecte si organizatii cum sunt:

1. *Dezvoltarea de software iterativ.* Majoritatea echipelor de proiectare folosesc înca un model în cascada pentru dezvoltarea de proiecte, completând într-o ordine stricta: analiza cerintelor, modelarea, implementarea/integrarea si testarea. Aceasta abordare ineficienta opreste din munca membri importanti ai echipei, pe perioade îndelungate si amâna testarea proiectului pâna la sfârşitul realizării

lui când problemele tind să fie dificil și costisitor de rezolvat, reprezentând o amenințare serioasă pentru realizarea termenului de predare.

În contrast, RUP reprezintă o abordare iterativă care este superioară pentru următoarele motive:

- Permite luarea în considerare a schimbării cerințelor. Adevărul este că de regulă, cerințele se schimbă și au efecte negative asupra termenului de predare, a satisfacției clienților etc.
- Integrarea nu este o mare explozie la sfârșit. În schimb, elementele sunt integrate progresiv, aproape continuu.
- Riscurile sunt de regulă descoperite sau rezolvate în timpul integrării. Cu o abordare iterativă, riscurile pot fi descoperite mai devreme. Pe măsura derulării iteratiilor se testează toate componentele de proces, tratând multe aspecte ale proiectului cum ar fi uneltele, software versiune finală, talentele oamenilor etc. Se poate vedea foarte repede dacă riscurile estimate se dovedesc a fi reale și se pot descoperi riscuri noi, neprevăzute, atuncicând ele sunt mai ușor și mai puțin costisitor de rezolvat.
- Dezvoltarea iterativă furnizează management cu un mijloc de a face schimbări tactice produsului. Aceasta permite realizarea și livrarea unei variante inițiale cu funcționalitate redusă pentru a contracara o mișcare a unui competitor.
- Iterarea facilitează re folosirea. Este mai ușor să fie identificate părțile comune pe măsura ce sunt parțial create sau implementate, decât să fie recunoscute în timpul planificării. Prevederea descrierii în iteratiile inițiale permite proiectanților să remarce oportunitățile potențiale pentru re folosire și apoi să dezvolte și să aducă la maturitate cod comun pentru aceste oportunități în iteratiile următoare.
- Când se pot corecta erorile pe parcursul a câtorva iteratii, rezultatul este o arhitectură mai robustă. Pe măsura ce produsul avansează de la concepție spre realizare, erorile sunt detectate chiar în iteratiile inițiale, mai ușor decât în timpul unei fa-

ze de testare masivă la sfârșit. Strangulările de performanță sunt descoperite într-un moment când pot să fie încă rezolvate.

- Dezvoltatorii pot să învețe pe parcursul elaborării: testerii încep testarea mai repede, programatorii încep scrierea programelor mai repede etc. Într-o dezvoltare noniterativă aceiași oameni ar trebui să aștepte până să-și înceapă munca.
- Însuși procesul de dezvoltare poate fi îmbunătățit și rafinat pe parcurs.

În RUP abordarea iterativă este foarte bine controlată: numărul, durata și obiectivele fiecărei iteratii sunt planificate, sarcinile și responsabilitățile participanților sunt bine definite.

2. *Administrarea cerințelor.* Aceasta presupune o abordare sistematică pentru solicitarea, organizarea, comunicarea și administrarea cerințelor schimbătoare ale unui sistem sau aplicație software. Beneficiile unei administrări corecte ale cerințelor sunt multiple:

- Control mai bun al proiectelor complexe. Aceasta include înțelegerea mai bună a comportării sistemului precum și prevenirea degradării cerințelor.
- Calitate mai bună a produsului software și satisfacția clientului. Măsura fundamentală a calității este aceea că un sistem face ceea ce este gândit să facă.
- Costuri și întârzieri reduse ale proiectului. Repararea erorilor în cerințe este foarte costisitoare. Cu o administrare corectă a cerințelor se poate scădea numărul erorilor în fazele inițiale ale dezvoltării, scăzând astfel costurile proiectului și prevenind întârzierile.
- O comunicare mai bună în cadrul echipei. Administrarea cerințelor facilitează implicarea utilizatorilor încă din fazele inițiale ale procesului asigurându-se că aplicația satisface nevoile lor.
- De regulă este dificil să se analizeze un model de sistem orientat obiect tradițional și să se precizeze cum face sistemul ceea ce este gândit să facă. Aceasta este dificilă datorită lipsei vizibilității unor legături consistente din cadrul sistemului atunci când acesta realizează anumite sarcini. Prin RUP *use cases* fur-

nizeaza aceasta legatura prin definirea comportamentului realizat de sistem. *Use cases* nu sunt necesare în gândirea orientata obiect traditionala, dar nici nu reprezinta un element obligatoriu în RUP. Când exista totusi, ele furnizeaza o legatura importanta între cerintele sistemului si alte elemente de dezvoltare cum ar fi modelul sau testele.

- Alte metode orientate obiect furnizeaza reprezentari ca si *use cases*, dar folosesc nume diferite pentru ele cum ar fi *scenarii* sau *fire de prelucrare*.

RUP este deci o abordare condusa de *use cases*, ceea ce înseamna ca *use cases* definite pentru sistem pot servi ca baza pentru restul procesului de dezvoltare.

3. *Utilizarea arhitecturii bazate pe componente*. *Use cases* conduce RUP de-a lungul întregii perioade de viata a unui sistem informatic, dar activitatile de design sunt concentrate pe arhitectura, fie de sistem, fie software pentru sistemele intensive. Obiectivul principal al iteratiilor initiale este sa produca si sa valideze o arhitectura software. În ciclul de dezvoltare initial aceasta ia forma unui prototip arhitectural executabil care evolueaza gradat prin iteratii succesive catre sistemul final. RUP furnizeaza în acest sens o cale metodica si sistematica de concepere, dezvoltare si validare a unei arhitecturi. El ofera template-uri pentru descrierea unei arhitecturi bazate pe conceptul unor viziuni arhitecturale multiple.

4. *Modelarea vizuala a software-ului*. Se stie ca modelele sunt reprezentari simplificate ale realitatii. UML este un limbaj grafic pentru vizualizarea, specificarea, construirea si documentarea elementelor unui sistem intensiv software. El furnizeaza un mijloc standard de descriere a structurii sistemului, acoperind elemente conceptuale cum ar fi: procese de afaceri si functii sistem, precum si elemente concrete cum ar fi: clase, schema bazei de date si componente software reutilizabile. În timp ce UML furnizeaza vocabularul pentru a exprima diferite modele, el nu spune cum sa se dezvolte produsul software. Acesta este motivul pentru care firma Rational a dezvoltat RUP, un ghid pentru fo-

losirea efectiva a UML pentru modelare. El descrie modelele de care avem nevoie, de ce este nevoie de ele si cum sa fie construite. RUP 2000 foloseste UML versiunea 1.4.

5. *Verificarea permanenta a calitatii software*. Potrivit concepiei RUP calitatea proiectului nu este adaugata acestuia de câtiva oameni, ci este responsabilitatea fiecarui membru al echipei de dezvoltare. RUP se concentreaza în ceea ce priveste calitatea asupra a doua concepte:

- calitatea produsului, care reprezinta calitatea produsului principal si a tuturor elementelor componente;
- calitatea procesului care este gradul în care un proces acceptabil a fost implementat si acceptat în procesul de realizare; în plus calitatea procesului este preocupata si de calitatea elementelor (planuri de testare, de iteratii, realizarea de *use cases* etc.) produse în sprijinul echipei de realizare.

IBM RUP este deci o platforma configurabila de dezvoltare a proceselor software care ofera cele mai bune metode si o arhitectura configurabila care permite selectarea si utilizarea numai a acelor componente ale proceselor de care este nevoie pentru fiecare etapa a proiectului. RUP include utilitare pentru configurarea RUP, pentru necesitatile specifice ale proiectului, unelte pentru dezvoltarea experientei în dezvoltarea componentelor proceselor, unelte de livrare puternice si adaptabile bazate pe Web si o comunitate online pentru schimburi de experienta între utilizatori si realizatori. În acest sens RUP este o infrastructura a metodologiei de dezvoltare software si o platforma practica flexibila si de succes pentru asistarea si îndrumarea dezvoltatorilor de software.

RUP contine urmatoarele componente:

Pentru configurarea procesului:

- *Constructor RUP (RUP Bilder)*. Un proces simplu, în patru etape, care ajuta la selectarea unei configuratii de baza, folosirea plug-in-urilor în alte componente ale procesului, configurarea viziunilor procesului si publicarea RUP-ului personalizat. Avantaj: permite managerilor de proiect si conducatorilor de echipa sa selecteze si sa foloseasca

doar acele componente ale procesului care sunt relevante pentru proiectul lor.

◆ *RUP Plug-ins*. Elemente care pot fi downloadate si care contin o varietate de componente de proces adaptate la instrumente, tehnologii sau domenii specifice. Avantaj: fiti mai elastic la schimbarile de tehnologie si combinati plug-in-uri multiple pentru a crea o configuratie a procesului potrivita pentru mediul specific al proiectului dumneavoastra.

☑ Pentru conceperea procesului:

◆ *IBM Rational Process Workbench*. Organizatorul RUP permite integrarea facila a continutului si asistentei procesului din organizatia dumneavoastra în configuratiile RUP. Modelatorul RUP permite schimbari în modelul procesului prin adaugarea, stergerea sau modificarea componentelor sau activitatilor. Avantaj: un nivel ridicat de integrare a procesului organizatiei dumneavoastra în asistenta RUP.

☑ Pentru livrarea procesului:

◆ *MyRUP si Process Views*. Permite accesarea de subseturi de informatii de proiecte de proces adaptate pentru diferiti indivizi sau roluri. MyRUP permite fiecarui individ sa-si contureze propria sa viziune. Avantaj: permite fiecarui utilizator sa acceseze asistenta cea mai relevanta pentru proiectul lor, îmbunătățind adoptarea procesului în cadrul echipei.

◆ *Asistenti de instrument (Tools Mentors)*. Oferă asistenta pentru utilizarea instrumentelor Rational pentru îndeplinirea scopurilor RUP. Avantaj: face procesul practic si ajuta membrii echipei să înțeleaga cum sa foloseasca mai bine instrumentele Rational în maniera cea mai eficienta.

◆ *Ajutor extins (Extended Help)*. Oferă asistenta RUP adaptata la context direct din instrumentele Rational. Avantaj: promoveaza folosirea mai eficienta a instrumentelor, accelerând reîntoarcerea investitiei.

☑ Pentru comunitatea si piata:

◆ *IBM Rational developer Network*. Este un portal Web creat în mod specific pentru echipele care folosesc instrumente Rational RUP Plug-In Exchange articole si lucrari, specializari bazate pe Web si forumuri de discutie, ofera o resursa mondiala si o comunitate pe n-

tru utilizatorii de RUP.

Un produs software modern, *JBuilder 6* include o serie de îmbunătățiri menite sa usureze munca programatorilor si sa ofere asistenta realizatorilor pe tot parcursul proiectului. Acest produs este disponibil în patru variante (*JBuilder 6 Personal*, *JBuilder 6 Professional*, *JBuilder 6 Enterprise* si *Enterprise Studio, Java Edition*). În sfârșit, *Enterprise Studio, Java Edition* ofera în plus fata de varianta *Enterprise*, pachetele *Rational Rose* cu integrare pentru *JBuilder*, *Rational Unified Process* si *Macromedia Dreamweaver UltraDev*.

Schimbarile majore si continue din domeniul proiectarii si dezvoltarii de produse software, antrenate de cele din domeniul tehnologiilor informatice si a complexitatii crescândă a problemelor de modelat si automatizat, fac sa apara mereu noi produse software care sa asiste si sa ghideze factorul uman pe toata perioada de conceptie si realizare a unui produs software. Acestea sunt tot mai mult încorporate în medii integrate de proiectare si programare, contribuind la cresterea eficientei activitatilor de dezvoltare software, prin cresterea calitatii si performantelor acestora, reducerea timpului de realizare si a costurilor implicate.

Bibliografie

- Mai-lan Tomsen Bukovec and Blair Shaw, **Office 2003**, Host an Interactive Visio Drawing Surface in .NET Custom Clients, sept.2003
 Philippe Kruchten, What Is the Rational Unified Process?, oct. 2003
 Philippe Kruchten, *The Rational Unified Process* (2000).
 Grady Booch *et al.*, *UML Users' Guide*, 2000
 Ivar Jacobson *et al.*, *The Unified Software Development Process*, 1999
Rational Unified Process 2000, Rational Software, Cupertino, CA(2000)

*** <http://www.rational.com/rup/>

*** <http://www.ibm.com/>