



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Proiect nr. 154/323 cod SMIS – 4428 cofinanțat de prin Fondul European de Dezvoltare Regională “Investiții pentru viitorul dumneavoastră”.

Programul Operațional Sectorial Creșterea Competitivității Economice - POS CCE



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Sisteme CAD/CASE

16. Instrumente și medii integrate de proiectare software

Introducere

- Dezvoltarea software-ului pe criterii de eficiență impune îmbunătățirea productivității muncii prin folosirea în toate fazele ciclului de elaborare a programelor a unor instrumente software care să asiste dezvoltatorul în activitatea de elaborare a software-ului de bună calitate, într-un termen cât mai scurt și la un cost redus.
- Inițial, preocupările au fost îndreptate spre folosirea instrumentelor cu precădere în cadrul activităților de codificare, testare și depanare a programelor. Ulterior, aceste preocupări s-au extins și asupra celorlalte faze. Astfel, a avut loc un proces de agregare a instrumentelor disparate, specializate pe anumite etape ale proiectării, în instrumente complexe (medii integrate, produse de tip CASE), care să asiste cât mai adecvat procesul de realizare a produselor software.
- Un mediu integrat de dezvoltare constituie un ansamblu integrat de instrumente de programare, care asistă dezvoltatorul pe parcursul activităților de codificare, obținere a programului executabil, testare și depanare.

Etape de proiectare și tipuri de instrumente CASE

- În tabelul de mai jos este reflectată corespondența între etapele ciclului de dezvoltare și categoriile de instrumente care oferă facilități pentru aceste etape.

<u>Etapa</u>	<u>Instrument</u>
Analiză	CASE
Proiectare	CASE
Implementare	Medii de dezvoltare, CASE (parțial), instrumente pentru gestiunea versiunilor și configurațiilor.
Testare și depanare	Medii de dezvoltare, instrumente de testare

Instrumente software disparate

- Instrumentele de programare sunt entități software folosite în realizarea altor programe software (modele, cod sursă, cod executabil). În această categorie sunt incluse instrumentele software puternic specializate pentru anumite subactivități ale proiectării software. Ele au fost concepute să funcționeze în mod independent de alte instrumente software și au fost foarte populare înainte de apariția mediilor integrate de dezvoltare. Și astăzi sunt folosite în mod curent instrumente izolate, care acoperă necesități specifice, nefiind incluse în categoria mediilor integrate sau produselor CASE.
- Putem diferenția trei categorii de astfel de instrumente:
 - Instrumente de programare
 - Instrumente de testare
 - Instrumente pentru gestiunea versiunilor și configurațiilor

Instrumente de programare

- **Editoare pentru codul sursă** - facilitează editarea programelor. Editoarele cuprind în mod standard funcții de creare , inserare de text, căutare , salvare , deschidere de fișiere, copiere, dar pot avea și facilități suplimentare de verificare a elementelor sintactice (*syntax highlighting*).
- **Compilatoare** - realizează transformarea codului sursă în cod obiect, semnalând erorile de sintaxă. Este integrat cu editorul, facilitând corectarea erorilor depistate prin poziționarea în codul sursă pe liniile implicate. Programatorul are la dispoziție opțiuni privind compilarea optimizată după diferite criterii sau platforme hardware, utilizarea unor versiuni standard ale limbajului de programare, etc.
- **Editoare de legături** - folosite pentru obținerea codului executabil prin legarea subprogramelor referite în codul sursă. Adesea este apelat în mod transparent pentru utilizator, după rularea compilatorului, dacă acesta nu a semnalat erori.
- **Interpretoare** - efectuează prelucrarea și execuția directă a instrucțiunilor sursă ale programelor.
- **Asambleoare** – permit accesul la orice funcție hardware sau a sistemului de operare prin instrucțiuni de nivel scăzut (limbaj mașină, funcții ale sistemului de operare).
- **Preprocesoare și precompilatoare** - folosite pentru a mari portabilitatea limbajelor de programare și a compilatoarelor.
- **Emulatoare** care permit rularea unor instrucțiuni specifice altor platforme hardware.

Instrumente de testare

- **Instrumente de capturare/redare** - pot înregistra o sesiune de testare într-un fișier script, permițând redarea (repetarea) acesteia. Pot fi efectuate teste multiple în maniera automată și se pot efectua comparații asupra rezultatelor.
- **Instrumente de execuție automată a testelor** - asemănătoare cu cele de mai sus, dar cazurile de test sunt specificate de utilizator în fișiere script.
- **Analizor de acoperire** - evaluează gradul în care structura codului testat a fost acoperită prin cazurile de test. Astfel de instrumente sunt utile pentru identificarea porțiunilor de cod netestate.
- **Generator de cazuri de test** - este un instrument care, pe baza unor informații precum cerințe, modele ale datelor, modele obiectuale ș.a., poate genera scenarii de test semnificative.
- **Generator de date de test** - sunt instrumente care servesc la popularea fișierelor și bazelor de date în vederea testării. Popularea se face în general cu date aleatoare, dar unele instrumente prevăd și posibilitatea specificării unor condiții. Instrumentele sunt utilizate în general pentru popularea cu volume mari de date, necesare testărilor operaționale și la capacitate maximă.
- **Analizor logic / de complexitate** - servește la cuantificarea complexității unor porțiuni de cod. Multe astfel de instrumente oferă și reprezentări grafice ale căilor posibile în structura codului. Sunt utile pentru determinarea scenariilor de test necesare pentru atingerea anumitor puncte din cod din rutine complexe.
- **Instrumente de trasare a erorilor** - permit gestiunea informațiilor privitoare la erorile detectate și stadiul corectării lor și centralizarea acestor informații pentru urmărirea tendințelor acestor defecte. Pe baza acestor tendințe se pot efectua îmbunătățiri în procesele de dezvoltare și/sau mentenanță ale organizației.
- **Instrumente de gestionare a testării** - au rolul de a asista planificarea și organizarea elementelor implicate în testare, precum fișiere script, cazuri de testare, rezultate.

Instrumente pentru gestiunea versiunilor

- Procesul de gestiune a versiunilor are rolul de a asigura lucrul consistent și coerent cu elementele de realizare a programelor (fișiere sursă, fișiere binare, executabile, documente).
- În cazul lucrului în echipă, gestiunea versiunilor cuprinde gestiunea centralizată a fișierelor (astfel încât membrii echipei să utilizeze cele mai noi versiuni validate) și gestiunea lucrului concurent (are rolul de a asigura consistența fișierelor atunci când acestea sunt modificate concomitent de mai mulți membri ai echipei).
- Instrumentele de gestiune a versiunilor sunt construite, în general, în jurul unui *repository*, care conține versiunile succesive ale fișierelor unuia sau mai multor proiecte. Operațiile uzuale într-un astfel de instrument sunt:
 - **check-out** – extragerea unui fișier din depozit în vederea modificării ;
 - **check-in** – actualizarea fișierului modificat, însoțită eventual de analiza diferențelor (*diff*) și integrarea modificărilor concurente (*merge*), cu semnalarea conflictelor.
- Gestiunea configurațiilor se aplică în cazul dezvoltării unui produs software în mai multe configurații (de exemplu pentru mai multe platforme hardware sau sisteme de operare, sau pentru limbi diferite) în mod simultan.

Medii integrate de dezvoltare clasice

- Mediile integrate de dezvoltare IDE (*Integrated Development Environment*) cuprind ansamblul de instrumente necesare implementării într-un anumit limbaj de programare.
- În componența unui mediu integrat de dezvoltare intră, în mod obligatoriu, următoarele instrumente:
 - Editor
 - Compilator
 - Editor de legături
- Pe lângă acestea sunt prezente și facilități de testare și depanare, care asistă programatorul în activitatea de depistare și corectare a erorilor prin:
- Intreruperea execuției programului în anumite puncte, stabilite de utilizator (*breakpoints*)
- Execuția pas-cu-pas a programului (*trace*), cu posibilitatea parcurgerii subprogramelor apelate (*trace into*) sau nu (*step over*).
- Inspectarea conținutului variabilelor (*watch*) în cursul rulării.

Medii integrate de dezvoltare vizuală

- Mediile integrate de dezvoltare vizuală (sau medii vizuale de dezvoltare) reprezintă o evoluție a mediilor integrate, vizând asistarea programatorului în realizarea interfețelor grafice ale aplicațiilor.
- Obiectivul acestor medii de programare este de a scuti programatorul de scrierea codului specific lucrului cu interfața grafică, acesta concentrându-se pe funcționalitatea aplicației.
- Programele realizate cu aceste medii de programare sunt construite în maniera orientată pe evenimente. Spre deosebire de programele clasice, în care se definește succesiunea de prelucrări de la intrarea în program până la terminarea sa, orientarea pe evenimente presupune scrierea de cod pentru tratarea evenimentelor generate de utilizator prin interacțiunea cu elementele de interfață (de exemplu click pe buton, modificarea unui câmp text etc.) În acest sens programatorul scrie proceduri specifice, care vor fi apelate automat la apariția evenimentelor tratate.
- Exemple de mediile de dezvoltare vizuală: Visual J# (Microsoft), JBuilder (Borland-Inprise), JDeveloper (Oracle) , Visual C++ (Microsoft), C++Builder (Borland-Inprise) .