

IMPORTANȚA LIMBAJULUI UNIFICAT DE MODELARE UML

Autori: Nina SAVA, Maxim TINCU

Universitatea Tehnică din Moldova

Abstract — Lucrarea de față abordează principiile de bază ale modelării, indicând metodologiile de modelare a limbajului UML.

Cuvinte cheie — aplicație, cod, programator, proiect, model, metodologie

I. Introducere

Aplicațiile de dimensiuni mari care execută un lucru complex în business procese și mișcă compania înainte, trebuie să fie mai mult decât o totalitate de cod strânsă într-un fișier text. Ele trebuie să fie structurate într-un mod care ar permite extinderea, securitatea și execuția solidă sub condiții de stres, iar structura sau arhitectura lor trebuie să fie clar definite ca programatorii să poată rapid să găsească eroarea apărută după ce autorii originali au început lucrul la alte proiecte [1]. Cu alte cuvinte aceste programe trebuie să fie proiectate pentru a funcționa perfect pe diferite platforme. Desigur, o arhitectură bine concepută este un beneficiu în orice program, nu numai în cele menționate mai sus. Un alt avantaj al structurii este că aceasta permite reutilizarea codului. Timpul de proiectare este cel mai scurt timp pentru a structura o aplicație ca o colecție de module sau componente independente. În cele din urmă, întreprinderile construiesc o bibliotecă de module, unde fiecare dintre ele reprezintă o funcție. Atunci când o altă aplicație necesită aceeași funcționalitate, programatorul poate importa rapid modulul său din bibliotecă.

II. Modelare

Modelarea este proiectarea aplicațiilor software, înainte de a trece la partea codului. Modelarea este o parte esențială a proiectelor mari și un ajutor pentru proiectele medii și mici. Cu ajutorul modelării, cei responsabili pentru succesul unui proiect se pot asigura că funcționalitatea este completă și corectă, sunt îndeplinite nevoile utilizatorilor și programul suportă cerințele pentru extindere, securitate și alte caracteristici. Sondajele arată că proiectele mari au o probabilitate mai mare de eșec, de fapt, este mult mai probabil că un proiect mare nu va reuși să îndeplinească toate cerințele sale la timp și în bugeul propus. Dacă dumneavoastră proiectați un astfel de proiect, trebuie să creșteți șansele de succes, iar modelarea este singura modalitate de a vizualiza structura aplicației în raport cu cerințele ei, înainte ca echipa să înceapă scrierea codului.

III. Abstractizare

Modelele ne ajută lăsându-ne să lucrăm la un nivel mai ridicat de abstractizare. Un model poate face acest lucru prin ascunderea diferitelor detalii, aducând pe prim plan diferite aspecte ale prototipului. Se poate concentra pe diferite aspecte ale cererii, cum ar fi business procese care automatizează diferite lucruri sau reguli. UML permite să specificăm, vizualizăm și să modelăm documente software, inclusiv structura și design-ul lor, într-un mod care permite îndeplinirea tuturor cerințelor. Folosind oricare dintre numărul mare de instrumente UML, puteți analiza cerințele aplicației dumneavoastră și să proiectați o metodă care le îndeplinește, reprezentând rezultatele folosind cele 13 tipuri de diagrame. Puteți modela orice tip de aplicație care rulează pe orice tip de hardware, sistem de operare, limbaj de programare și de rețea. Construit pe concepte fundamentale, aceasta include diferite limbaje orientate pe obiect, cum ar fi C++, Java, C# [1].

IV. Modele Și Metodologii

Procesul de colectare și analiza cerințelor unei aplicații și încorporarea lor într-un program de proiectare este foarte complex, iar la momentul dat industria suportă mai multe metodologii care specifică modul în care aceasta să fie implementată. Indiferent de metodologia pe care o utilizați pentru a efectua analiza, puteți să utilizați UML pentru a afișa rezultatele. Folosind XML, puteți transfera modelul UML dintr-un instrument într-un depozit, sau într-un alt instrument pentru rafinare sau pentru următorul pas în dezvoltarea aleasă. Toate acestea sunt datorate standartzării.

V. Primul Start

UML definește 13 tipuri de diagrame, împărțite în 3 categorii: 6 tipuri de diagrame reprezintă structura aplicației statice, 3 reprezintă tipuri generale de comportament iar 4 reprezintă diferite aspecte a interacțiunilor. De la început pentru dezvoltarea în UML trebuie să știți 3 lucruri:

- a) Selectarea metodologiei;
- b) Selectarea instrumentului;
- c) Antrenare.

O metodologie definește procesul pe care îl utilizați pentru a aduna cerințele aplicației, analiza lor și proiectarea care satisface acestor cerințe. Există mai multe metodologii și fiecare din ele se diferențiază într-un fel sau altul. Există mai multe motive pentru care o metodologie poate fi mai bună decât alta pentru proiectul dvs. De exemplu, unele sunt mai potrivite pentru aplicații în întreprinderi mari, în timp ce altele sunt construite pentru a proiecta sisteme mici. Pe de altă parte, unele metode mai bine suportă un număr mare de arhitecți și designeri care lucrează în același proiect, în timp ce alte metode lucrează mai bine atunci când sunt utilizate de către o persoană sau un grup mic de oameni.

Deoarece cele mai multe instrumente bazate pe UML pune în aplicare o metodologie specială, în unele cazuri, s-ar putea să nu fie practic pentru a alege un instrument și apoi să încercați să folosiți o metodologie care nu a fost construită pentru acest instrument. Însă, unele metodologii au fost implementate pe mai multe instrumente astfel încât acest lucru nu este strict o alegere de un mediu. S-ar putea găsi un instrument atât de bine potrivit pentru proiectul dvs încât sunteți dispuși să schimbați metodologiile pentru a o folosi. Puteți găsi o metodologie care vă place, care nu este pusă în aplicare într-un instrument, care se potrivește dimensiunii proiectului dvs, sau bugetului, astfel încât trebuie să vă schimbați. În cazul în care sunteți în așa situație, încercați să alegeți o metodologie alternativă, care nu diferă prea mult de cea pe care și ales-o.

Dacă nu aveți buget necesar pentru a angaja arhitecți-UML cu experiență, veți avea nevoie de antrenare în UML. Antrenarea este cea mai bună metodă de a obține cunoștințe cum se folosește instrumentul dat cu metodologia aleasă de dumneavoastră. Dacă decideți să nu mergeți pe această cale, consultați pagina de ajutor pentru un curs care satisface nevoile dumneavoastră.

Concluzie

Limbajul unificat de modelare UML are un scop general de dezvoltare, un limbaj de modelare în domeniul ingineriei software, care este destinat să ofere o modalitate standardă de a vizualiza proiectarea unui sistem. Cu ajutorul lui putem vizualiza interacțiunile utilizatorilor, procesele și structura sistemului pe care dumneavoastră doriți să o construiți. Vă va ajuta la economisirea de timp și va asigura că toți membrii echipei lucrează eficient pentru a construi aplicația după cerințele necesare.

Bibliografie

[1] Introduction to OMG's Unified Modeling Language UML Recommendation - http://www.omg.org/gettingstarted/what_is_uml.htm