

Java versus .NET

Data: 15.07.2002

Autori: Despina Eftimescu, SecureNet S.R.L, Craiova (de@secure-net.ro)

Compania: SecureNet S.R.L., Craiova, tel.: 0251/410555, <http://www.secure-net.ro>

Despre Autor:

Despina Eftimescu a studiat informatica la Universitatea de Vest din Timisoara. Din 1994 lucreaza ca si inginer software si consultatnt in materie de strategie software pentru Internet la München, in Germania. Dupa o colaborare de peste 5 ani cu firma SecureNet GmbH din München, Despina Eftimescu a fondat si conduce in calitate de Administrator filiala romana a acesteia, SecureNet SRL in Craiova.

Abstract:

J2EE și .NET sunt platformele viitorului. Dar care platforma i se potrivește cui? Sau: „Cine vrea să cheltuiască bani pe produse Java?”

O privire comparativa asupra tehnologiilor Java si .Net, singurele alternative profesionale pentru dezvoltarea de aplicatii in mediul de afaceri. Dupa succesul inregistrat de Java in ultimii zece ani, Microsoft incearca sa introduca pe piata .Net, o creatie proprie si proprietara, care sa concureze tehnologia Java.

Este .Net cu adevarat o alternativa? Si daca da, pentru cine?

Evolutia ingineriei software in ultimii 5 ani a minimalizat utilizarea limbajelor 4GL, asa cum limbaje ca Smalltalk au fost inlocuite de Java, iar conceptul de „fat client” a fost înlocuit de o arhitectură multinivel. In compensatie s-au impus diferite tehnologii web si XML (Extended Markup Language). Inițiative de tip middleware cum ar fi Distributed Computing Environment (DCE) sau Common Object Request Broker Architecture (CORBA) sunt înlocuite treptat cu infrastructuri tehnice de mare complexitate.

Astăzi au rămas în esență numai două platforme tehnologice pentru aplicații noi: Java 2 Enterprise Edition (J2EE) și Microsoft .NET.

Pentru a lua o decizie corectă pentru viitor, utilizatorii trebuie să ia în considerare două criterii importante:

- potențialul de dezvoltare și comprehensibilitate susținut de o anumită platformă
- oferta de soluții proprii ale platformei, necesare pentru a se menține pe piață

Limbajul Java a cunoscut o dezvoltare deosebită în ultimii cinci ani, de la un „capriciu” al împătimiților pentru limbajele orientate obiect la o tehnologie larg răspândită. Această evoluție se datorează mai puțin farmecului limbajului și mai mult fortei platformei tehnice asociate.

Recent și Microsoft oferă o platformă tehnologică asemănătoare. Cei din Redmond au început cu Distributed InterNet Architecture (DNA), care avea însă multe defecte.

Pe lângă problemele cu registrul și conflictele de la nivelul DLL-urilor, a apărut conceptul de model obiect componentă (COM), care a devenit prea complex din cauza suportului pentru diferite limbaje. De asemenea procesarea

distribuită cu soluții DCOM pe baza conceptului Microsoft RPC și a registrului Windows nu s-a dovedit a fi compatibilă cu Internetul.

.NET a învățat de la Java

Începând cu .NET, Microsoft a realizat o importantă schimbare față de DNA, compania lui Bill Gates deschizând problema tehnologiei predecesoare. La capitolul arhitectură și funcționalitate .NET și J2EE sunt foarte asemănătoare, dar Microsoft oferă o soluție tehnică mai modernă, prin implementarea tehnologiilor web și a limbajului XML. De asemenea, noul limbaj C# și mașina virtuală (CLR) sunt idei provenite din Java.

Există și alte diferențe de importanță strategică:

1. J2EE nu este un produs, ci o specificație, pentru care diferite companii oferă produse. Aplicațiile sunt independente de proprietarul suportului middleware. Astfel, companiile obțin nu numai o independență față de un anumit furnizor, dar pot să-și dezvolte propriile platforme tehnologice.
.NET este o colecție de produse ale unui singur producător și rulează numai împreună cu Windows. Se asigură integrarea diferitelor componente și utilizarea unor caracteristici speciale ale sistemului de operare Windows.
2. J2EE este independent de conceptul de sistem de operare. Portabilitatea este asigurată de Java Runtime Environment, iar serverul de aplicații și alte produse middleware pot fi programate în funcție de sistemul de operare.

Pe lângă aceste aspecte, mai există și alte criterii importante în luarea unei decizii în privința acestor două tehnologii, cum ar fi nivelul de comprehensibilitate și dezvoltare, ceea ce analiștii de la Gartner numesc „completeness of vision”. Avantajul J2EE constă în existența interfețelor API (Application Programming Interface), care creează o independență tehnologică a aplicațiilor. Aceasta facilitează dezvoltarea ulterioară a tehnologiei cu efecte secundare reduse. Modelul componentelor Java este mai metodic și mai elaborat, iar arhitectura bazată pe conectori oferă baza pentru o mai mare interoperabilitate decât facilitățile corespondente în tehnologia .Net.

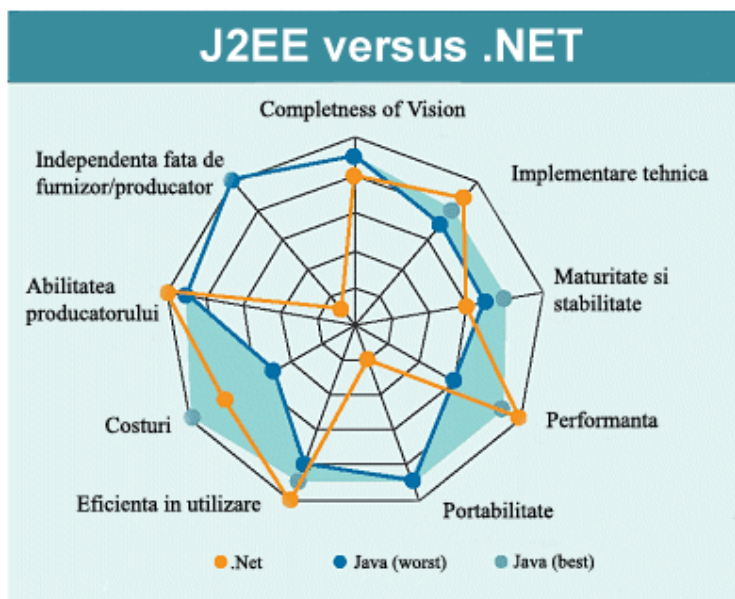
.Net versus Java: criterii de comparație

Înainte de a lua o decizie, responsabilul IT trebuie să mai treacă sub lupă și implementarea tehnică a infrastructurii informatice respective. Sub acest aspect .NET posedă câteva avantaje, deoarece utilizează de la început tehnologii moderne cum ar fi XML și serviciile web. Prin dezvoltarea propriei mașini virtuale Microsoft rezolvă problemele datorate interpretorului din Java. Multe funcționalități ale sistemului de operare Windows pot fi utilizate direct, cum ar fi serverul web IIS, Active Directory, OLEDB și Windows Load Balancing. Cuplarea eficientă cu sistemul de operare este cauza performanțelor îmbunătățite ale aplicațiilor .NET, comparativ cu cele ale

aplicațiilor J2EE, deși este dificil de estimat obiectivitatea testelor respective. Totuși performanțele pot varia, astfel performanțele mașinii virtuale Java pe același suport hardware variază cu un factor egal cu trei, iar această variație e chiar mai mare pentru serverele de aplicații Java.

Experiența arată ca produsele software complexe au nevoie de ani de zile pentru a fi dezvoltate. Asta se întâmplă cu bazele de date, cu aplicațiile de monitorizare a tranzacțiilor sau cu aplicațiile Java, ceea ce face ca dezvoltarea unui nou produs construit pe un nucleu de componente mai vechi să nu constituie un dezavantaj.

Produsele J2EE au căpătat în timp un grad acceptabil de maturizare. Pentru .NET experiența practicilor în dezvoltarea aplicațiilor proprii lipsește, pentru a face o comparație.



Diferitele variante de implementare a specificatiei J2EE ne confrunta cu diferite calitati ale produselor finale, care variaza între "Java best" (cea mai buna solutie) și "Java worst" (cea mai slaba solutie).

Dincolo de criteriile de performanță, managerii IT trebuie să țină cont de eficiența platformei și de productivitatea furnizată în dezvoltarea aplicațiilor. Dacă se măsoară productivitatea numai pe baza „numărului de linii de cod”, .NET prezintă avantaje clare în fața J2EE. Crucial este însă cât dintre aceste linii de cod trebuie să fie scrise la mână. Aici intervine atât procesul automatizat de dezvoltare cât și inteligența mediului de dezvoltare software.

În Java există diferențe semnificative între uneltele de dezvoltare, care sunt subliniate de furnizori în campaniile lor de marketing. IDE (Integrated Development Environment) cum ar fi Jbuilder, Forte și mai nou ECLIPSE oferă suport optimal pentru dezvoltare, debug și versionare de proiecte oricât de complexe, lasând totuși programatorului întregul control și transparența perfectă față de codul Java rezultat. Uneltele „Visual Studio .NET” de la Microsoft pot fi comparate cu cele mai eficiente medii de dezvoltare din Java.

Costurile cu licențele și cu întreținerea sunt un factor ce nu trebuie neglijat în luarea deciziei pentru sau contra unei tehnologii. Costurile soluțiilor J2EE ale unor furnizori ca IBM sau BEA sunt transferate în categoria mainframe-urilor. Nu trebuie să uităm ofertele open-source cu preț zero, care necesită doar puțin curaj în implementare. Forte, Turbine, Struts, Cactus, Junit, Eclipse – sunt doar câteva nume de frameworkuri, module sau unelte gratuite care nu mai au nevoie de nici o recomandare suplimentară în lumea programatorilor și a inginerilor de software.

Pe scara prețurilor de la gratuit la scump, Microsoft .NET este așezat undeva pe la mijloc.

Mai apar și cheltuieli cu personalul calificat care să opereze o infrastructură atât de complexă. Mâna de lucru este scumpă și comparabilă pentru ambele tehnologii. Trebuie avut în vedere că J2EE și Java sunt în schimb pe piața de câteva ani de zile, fapt care a „produs” deja mina de lucru experimentată. Tehnologia .Net este în stadiul de nou-născut, iar specialiștii în acest domeniu abia se formează.

Pentru a analiza dacă o tehnologie sau un concept este complet, trebuie luat în considerare dacă ofertantul continuă să susțină și să dezvolte propria tehnologie pe piață. Luând în considerare valoarea strategică a acestor tehnologii, care sunt momentan singurele în măsură să susțină dezvoltarea de aplicații de tip B2B sau B2E, este foarte puțin probabil să nu se găsească investitori pentru ambele linii de dezvoltare, indiferent de situația financiară a creatorilor lor.

Furnizorii de tehnologie Java sunt capabili să sfideze puterea de dezvoltare și marketing a celor de la Microsoft. Pentru ei aceasta este de fapt o strategie de supraviețuire. Dar ei nu depind în mod necesar de existența companiei Sun. În cazul insolvenței companiei Sun, rolul său poate fi transferat altei companii.

Grupul de producători de soluții Java și-a împărțit capacitatea de producție. Orice produs Java este dezvoltat în paralel de 6-8 producători. Astfel există aproximativ 40 de producători de servere de aplicații. Dar competiția accelerează și diversificarea ideilor. Nici puterea de inovație a comunității open-source nu trebuie omisă. Proiecte ca Apache arată gradul de dezvoltare de produse eficiente open-source.

Schimbările ulterioare sunt scumpe

Vizualizarea situației actuale nu este uneori suficientă în luarea deciziilor. Companiile investesc milioane de Euro în infrastructură și în aplicații noi. Dacă o platformă tehnologică a fost aleasă, costurile generate de modificare sau înlocuire ulterioară sunt foarte mari. Studiul Gartner arată că apariția Microsoft .NET va produce mutații la nivelul comunității Java, dar nu va înlocui Java. Este însă improbabil ca Java să rămână dominant în așa măsură încât .NET să nu obțină procente din piață. Miile de documente descărcate de pe Internet arată un interes activ în .NET. Ținând cont și de eforturile de marketing

si vinzari depuse de Microsoft pentru introducerea pe piata a alternativei tehnologice .Net, este probabil că cele două platforme tehnologice vor coexista în viitor.

Multe companii au sesizat ca Microsoft .NET reprezintă un mare pas înainte, dar nu au motive să renunțe la Java. Cine a investit deja în Java găsește puține motive să o schimbe și să treacă la .NET. .NET nu oferă chiar atât de multe avantaje până la urmă.

Potentiali clienti sau o posibila impartire a pietei intre Java si .Net

Mai rămân companiile care nu s-au decis încă în care din cele două tehnologii vor investi. Aici mai degrabă obiective strategice vor afecta decizia. Companiile mai mici, cele care dispun de echipamente cu procesoare Intel ieftine si cu sisteme de operare Windows, vor opta pentru .NET pentru noile aplicații. Companiile mari, ce dispun de sisteme mainframe dotate cu sisteme Unix vor opta pentru o platformă ce rulează pe aceste mainframe-uri, aceasta fiind J2EE.

Pentru mulți dintre utilizatori, inventarea limbajului Java a fost un eveniment. Dacă ei au trebuit să testeze în trecut propriile aplicații pe mai multe sisteme de operare, acum ei le testează pe implementarea standard de la Sun, cu eventuale teste adiționale pe medii de producție dezvoltate de IBM, BEA sau Oracle.

Dar și .NET atrage producătorii de software. Aceasta depinde de infrastructura clientului respectiv. Dacă o companie cumpără o aplicație Java, atunci (cel puțin în prezent), trebuie utilizat și un mediu J2EE ce depinde de extensiile stabilite de fabricant. Prin cumpărarea mai multor pachete software o companie ajunge să posede mai multe servere de aplicație. Cu .NET o infrastructură pentru toate aplicațiile este suficientă. Se mai pune doar întrebarea dacă .Net asigura si calitatea de care este nevoie in mediul de afaceri, Microsoft avind pina la acest moment experiente (nu totdeauna laudabile) doar in mediul utilizator (home office sau consumer).

Cerințele legate de cunoștințele dezvoltatorilor de aplicații sunt comparabile. În afara de limbajul și uneltele specifice, mai trebuie stăpânite cunoștințe despre programarea orientată pe obiecte, arhitectura bazată pe straturi, design de componente și implementare iterativă. Cei care au dezvoltat aplicații Microsoft în trecut pot trece ușor la .NET. Totuși, un dezvoltator „clasic” care a implementat în trecut în Visual Basic soluții bazate pe „fat client” ar putea întâmpina dificultăți în implementarea cu .Net. În plus VB.NET nu mai este identic cu vechiul Visual Basic.

Aceleași probleme apar și la migrarea programelor Microsoft vechi la .NET. Pentru unele module se oferă asistenți de migrare, dar aceasta este valabil numai dacă aplicațiile vechi s-au dezvoltat pe o arhitectură pe trei nivele, pe bază de componente.

Deoarece tendința merge către achiziționarea de software în loc de dezvoltare internă, companiile au puține șanse să primească tehnologie omogenă. Astfel încât se creează un mixaj de soluții, pentru mainframe, PC, aplicații Java și .NET, prioritatea fiind abilitatea de a integra platforme eterogene.

Pentru sarcinile de integrare între tehnologii se poate utiliza XML sau serviciile web, care sunt suportate atât de Java cât și de .NET. Chiar și aplicațiile mainframe pot fi integrate cu un anumit cost. Cea mai mare rezistență la integrarea cu alte programe o vor avea aplicațiile din era „fat client”, adică aplicațiile Microsoft cu tehnologii mai vechi.