

A Microsoft .NET fejlesztési platform és annak hazai bevezetése

A felsőoktatás támogatása

NACSA SÁNDOR

Microsoft Magyarország Kft.

snacsamicrosoft.com

A Microsoft két évvel ezelőtt jelentette be a .NET-nek nevezett, stratégiai fejlesztési programot, amelynek központi technológiai eleme volt az akkor ún. „technology preview” változatban bemutatott .NET fejlesztési platform, mely két fő összetevőt jelentett. A Microsoft .NET Framework képviselte a lehető legáltalánosabb és a lehető leghosszabb távra szóló fejlesztői környezetet, míg a Visual Studio .NET a lehető leghatékonyabb munkavégzés szintjén mindezt megjelenítő, integrált fejlesztőrendszert.

2001. végére mindkét technológiai összetevő termékévé érett. 2002. február 13-án megtörtént a világbemutató, majd május 7.-én a helyi bejelentés is. A *Computerworld-Számítástechnika* ehhez készített melléklete jól összefoglalja a szakmai tudnivalókat (lásd <ftp://mshu.datanet.hu/public/.NET/MSDotNet-CWSzT.pdf>).

Nemcsak a Microsoft, hanem az egész iparág történetében példátlan méretű rendszerekről van szó. A .NET Framework esetében elég annyit megemlíteni, hogy annak teljes könyvtárában kb. 4500 osztály található, a programozási interfészek száma pedig az 50 ezerhez közelít. A Visual Studio .NET rendszerméretét pedig jól jellemzi az, hogy a fejlesztői munka minden fázisára, a tervezéstől a tesztelésig terjedően, teljes értékű és

rendkívül magas szintű támogatást nyújt. A legapróbb munkaelemekben is igen fejlett, beépített automatizmusok segítik a fejlesztőket abban, hogy csak a megoldandó problémával kelljen foglalkozniuk, és ne vesszenek el sem a .NET Framework, sem pedig a régi és újrafelhasznált, illetve újonnan keletkezett és továbbfejlesztendő szoftverek részleteiben.

Ha egyetlen kifejezéssel élve minősíteni kellene a .NET fejlesztési platformot, mint terméket, akkor dr. Alain Kay nevezetes bölcselete, „Az egyszerű dolgoknak egyszerűeknek kell lenniük, az összetetteknek pedig lehetségeseknek” (Simple things should be simple, complex things should be possible) fejezi ki legtömörebben a lényegét. Ahol egyáltalán lehetséges volt a platform fejlesztői a végletekig leegyszerűsítették a dolgokat. Így például egy XML Web szolgáltatás előállításában annyiban különbözik egy szokásos szoftver szolgáltatás előállításától, hogy a szolgáltatást nyújtó szoftver komponens metódusait egyszerűen el kell látni az ún. WebMethod attribútummal, a különbségek implementálásáról pedig maga a rendszer automatikusan gondoskodik. Ha belegondolunk, ez így természetes, hiszen egy webes szoftver szolgáltatás esetében ugyanolyan egyszerű dologról van szó, mint a szokásos, a saját gépünkön belül, programunk által felhasznált

szoftver szolgáltatás esetében.

Az összetettség egyszerűből kiinduló lehetőségként való felkínálására jó példa az, amikor szoftverünk egyes részeit más, távoli gépeken, rendszereken kívánjuk futtatni, és meg kell szerveznünk a most már távoli részek közötti kapcsolatot is. Az ehhez szükséges, ún. remoting funkcionális bevezetése programunkba a Visual Studio .NET-ben megfelelő attribútumok és más kódkiegészítések segítségével egy komponensre kb. 10 percnyi munkával végezhető el. A mögöttes, ennél sokkal nagyobb mértékű átalakításokról – az attribútumok és a kódkiegészítések alapján – maga a rendszer gondoskodik, ismét csak automatikusan.

Lényeges tehát, hogy az új platform összetevőinek mérete, kiterjedése a használatot egyáltalán nem nehezíti. A lehető leghatékonyabb munkavégzés támogatása éppen a méret és kiterjedés megfelelő bővülésével vált lehetségessé, miközben az általánosság és a távlatosság igényei eleve nagy méretet és kiterjedést jelentettek. Más oldalról közelítve: a .NET fejlesztési platform egyáltalán nem olyan, mint a legelső termékváltozatok „szoktak lenni”. Itt már a V1-ben minden elem teljesen kiérlelt formában jelenik meg, nem kell tehát az n-ik változatra várni ahhoz, hogy az adott elem elérje a lehető legegyszerűbb használatnak megfelelő funkcionális fejlettséget a rendszerben.

Az állítás igazolására csak elő kell venni mások legújabb fejlesztési platformjait, és össze kell hasonlítani azokat a .NET fejlesztési platformmal. Már az XML Web szolgáltatások támogatása téren szembeötlő lesz a különbség. Hosszadalmas, és tegyük hozzá, hogy hibák bevitelére alkalmas műveletssorokkal szemben itt a WebMethod attribútummal való ellátás az egyetlen feladat.

Bevezetési sajátosságok és problémák

Bármennyire is leegyszerűsítette a Microsoft a .NET fejlesztési platformot, annak **megfelelő** bevezetése még mindig példátlan méretű problémát jelent. A szokásos termékeknek ugyanis van megfelelő előzményük, ami mindenkinek jól ismert viszonyítási alap. A szokásos termékek ráadásul egyáltalán nem ilyen, „mindent átfogóak”, hanem ehhez képest meglehetősen szűk területekre korlátozottak. A szokásos termékekre a hosszú, publikus evolúció a jellemző, ami könnyít a bevezetésben, de – tegyük hozzá – komoly nehézségeket okoz a **teljes** megoldásrendszer műszakilag legmegfelelőbb kidolgozásában, mivel mindig kompromisszumokat kell kötni és szinte lehetetlen meghaladni a korábbi verziók koncepcionális korlátait.

Az állítás illusztrálására vegyük a .NET Framework gerincét alkotó, ún. Common Language Runtime (CLR) példáját. Még az ún. akadémiai szférában is sokan elintézik a dolgot annyival, hogy „no igen, ez olyasmi, mint a JVM (Java Virtual Machine)”. Mi több vannak közöttük olyanok, akik úgy vélik, hogy a JVM csak jobb lehet, mint ez a még csak most megjelent CLR, mivel a JVM már évek óta a piacon van, és ezért kellően „kiforrott”, hogy az „elterjedtségéről” ne is beszéljünk. (A valósághoz legközelebb álló „akadémiai” dolgozatot lásd a <http://research.microsoft.com/~emeijer/Papers/CLR.pdf> címen.)

A valóság ezzel szemben az, hogy a Microsoft egy JVM szintű, de nem Java-s dolgot már évekkel ezelőtt megjelentethetett volna, ha csak egyszerűen a „nekem is erre van szükségem” („me too”) megközelítés vezérelte volna. A JVM-et ugyanis eredetileg interpretatív megoldásként alakították ki, csak az ún. beágyazott kisrendszerekhez szánták és kizárólag egy adott nyelv, a Java igényeiből indultak ki. A CLR ezzel szemben eleve az

ún. just-in-time fordítás elvének legjobb felhasználására lett tervezve, a legkisebektől a lehetséges legnagyobb rendszerekig terjedő felhasználásban, és a legkülönbözőbb nyelvek lehető legteljesebb körének igényei szerint.

A JVM és a CLR közötti minőségi különbség még szembeötlőbb, ha azt is figyelembe vesszük, hogy a JVM-hez valóban hasonlítható dolgok már régóta léteznek, míg a CLR-hez hasonlíthatók korábban nem voltak és még mindig nincsenek. Így pl. az ún. p-kódon alapuló absztrakt futtató motorok az eredeti Pascal rendszerek mellett az ún. UCSD Pascal-ban, sőt a későbbi Modula rendszerekben is megtalálhatók voltak.

A CLR-nek ugyanakkor legfeljebb előzménye van, az is csak egy, a Xerox PARC-ban a '70-es évek végén kidolgozott kódfuttató motor. Csak magas, funkcionális szinten való rokonságról van azonban szó, abban az értelemben, hogy a Xerox PARC kódvégrehajtójára is több, jellegében is különböző nyelvről (mint SmallTalk, LISP, Mesa) lehetett fordítani, és a szokásos veremsszervezésű eljárásvégrehajtás mellett a dinamikus memóriakezelésnél ugyancsak alkalmazta az automatikus visszavétellel és tömörítéssel kombináló háttér megoldást, az explicit allokációt. Teljesen más volt viszont abban, hogy esetében tisztán hardver általi kódvégrehajtásról és dinamikus memóriakezelésről volt szó, szemben a CLR tisztán szoftverben működő rendszerével.

A CLR, mint tisztán szoftverben lévő mechanizmus ezért egy sor olyan dologra is képes, amit hardverben – legalábbis versenyképes, „gazdaságos” módon – sem korábban, sem most, sem a jövőben nem lehet megvalósítani.

Az egyik ilyen jellemző az ún. típuson alapuló biztonság (type safety). Röviden arról van szó, hogy a CLR-rel futtatandó kód teljesen biztonságos lehet, vagyis eleve garantált, hogy nem fog „belepiszkítani” más, vele párhuzamosan, de tőle

logikailag függetlenül futtatandó kódba. Az ilyen CLR kód az ún. menedzselt kód. Eleve ilyen kódot kapunk a C#, a Visual Basic .NET vagy a J# használatakor. Ez a kód tette lehetővé az ún. alkalmazási tartomány (application domain) bevezetését, ami viszont mentesít bennünket attól, hogy az operációs rendszer processz mechanizmusával tudjuk csak megvédeni logikailag független kódjainkat egymástól. A processz mechanizmus ilyen célú alkalmazásának ugyanis nagy ára van. Az alkalmazási tartományok közötti kommunikáció jóval hatékonyabb, mint a processzek közötti kommunikáció. Másként fogalmazva: egy processzben nyugodtan lehet akár több száz alkalmazási tartomány, míg egy nem .NET-es rendszerben az ennek megfelelő, több száz processzben való futtatás már túlzottan alacsony hatékonyságú, netán teljesen elképzelhetetlen lenne.

Már a CLR fenti jellegzetességeiből is jól látszik, hogy még a CLR, mint az egész egyetlen része esetében sem lehet viszonyítási alapként felhasználható, már létező dolgokat figyelembe venni. Az alkalmazási tartományok példáján azt is láthatjuk, hogy egy-egy jellegzetességnek lényegi és nagyjelentőségű kihatásai vannak a rendszer egészére nézve. Nincs mese, a maguk teljes valójában meg kell ismertetnünk a fejlesztői, sőt az egész szakmai társadalommal a lényegi elemeket ahhoz, hogy teljes értékű bevezetésről beszélhessünk. De megengedhetjük-e ezt magunknak, legalábbis ilyen egyszerűen?

A válasz egyértelműen nem. Annyi mindent kellene ugyanis az alapoktól kezdődően megismertetnünk az érintettekkel, amennyi tanulási szándékot, de nem utolsó sorban időt egyszerűen nem várhatunk el tőlük. Feladatunk ezek után az volt, hogy a meglévő megismerési szándékokhoz és rendelkezésre álló időkhöz „optimalizáljuk” a bevezetés egész rendszerét, mégpedig úgy, hogy a

lényegi elemek legalább egy második körben esedékes, úgynevezett „mélyvíz” során teljesen elsajátíthatók legyenek. Ez, az esetünkben menet közben kialakuló felismerés határozta meg a .NET fejlesztési platform hazai bevezetésének egész rendszerét.

Hazai megoldás a felkészülésre

A megfelelő hazai megoldás kialakítását nagyban segítette a német Microsoft bevezetési tervezetének korai szakaszban való megismerése. A német leányvállalat egy, a lényegi tulajdonságokat átfogó módon ismertető, erősen moduláris ugyanakkor testreszabható, minimálisan egy hetes, általános tanfolyam fejlesztésével gondolta megoldani a bevezetési problémát. Egyúttal azt is feltételezte, hogy ez kellő lökést ad majd a reguláris képzésnek is, mely természetesen sokkal mélyebb, készségszintű és ennek megfelelően sokkal terjedelmesebb, így önmagában nem alkalmas a bevezetés támogatására.

A német Microsoft nem kevesebbet, mint 1,5 millió márkát áldozott erre a célra. Az első modulkészletet már a 2001. januárjában tartott, ún. train-the-trainer tanfolyamon ismerhette meg két szakember a Microsoft fejlesztői közösség szakmai fejlődését támogató, de a cégtől független, ún. MSDN regionális szervezettől, és három oktató két hivatalos Microsoft oktató-központtól.

Az elképzelés az volt, hogy az MSDN regionális szervezet munkatársai a modulokban rejlő, már feldolgozott tudást jól tudják majd hasznosítani a teljesen saját fejlesztésű, szakmai alapozó konferencia kidolgozásához, míg az oktató-központok –delegált és kiképzett munkatársaik révén– kedvet kapnak majd a reguláris .NET tanfolyamok előtti, bevezető képzés meghirdetéséhez. Ez utóbbi megegyezett a német elképzelésekkel, ahol eleve az oktatóközpontokon keresztül gon-

dolták megvalósítani a bevezetést, mégpedig a várható, nagy volumenű igényekhez igazítható, (ottani viszonylatban) igen alacsony és ezért kelően vonzó áron.

A szakmai alapozó konferenciára (2001. május 30.) vonatkozó elképzelést sikerült messze túlszárnyalnunk:

- (1) Egyrészt a dr. Charaf Hassan által vezetett regionális MSDN csapat három, konferencia előtti szakmai szemináriumra (ún. tutorial) is vállalkozott a következő témákban:
 - Microsoft .NET, mint szolgáltatás alapú fejlesztési platform
 - COM+ és .NET együttműködés
 - Szoftvertervezés UML alapokon
- (2) Másrészt a hazai fejlesztői konferenciák történetében először került sor három párhuzamos szekcióban előadásokra, és ennek megfelelően sikerült egyetlen előadási napba sűríteni a .NET fejlesztési platformmal kapcsolatos minden alapvető ismeretet. Több témához szakmai ismertető cikket, ún. whitepaper-t is mellékeltek az előadók. Az előadásokról pedig videó felvétel készült, amelyet mindenki megkapott a későbbiekben CD-ROM-on, így az általa – más, számára fontosabb témájú előadás miatt – nem végighallgatott előadást is megismerhette némi késéssel. Ilyen kiszolgálással a következő témákról kaptak részletes áttekintést a résztvevők:
 - Technológiák:
 - .NET Framework
 - COM+ és .NET együttműködés
 - ADO.NET
 - ASP.NET
 - Protokollok és szabványok:
 - XML – Adatkezelés új megközelítésben
 - Simple Object Access Protocol (SOAP)

- UDDI, WSDL, DISCO
 - Felhasználó-orientált XML webszolgáltatások: Hailstorm (ma: .NET MyServices)
 - BizTalk Server használata saját alkalmazásunkban
- Nyelvek és felhasználás:
- Visual Basic .NET
 - C++ a .NET-ben (felügyelt kiterjesztések stb.)
 - ATL és MFC újdonságok
 - C#
 - Komponens és .NET alapú fejlesztések nagyvállalati környezetben.

Ugyanakkor gyakorlatilag megbukott a másik, oktató-központokon keresztüli bevezetés elképzelése. Az egyik oktató-központ vezetője ugyan erőteljesen forszírozta azt, hogy már az alapozó szakmai konferencián is szerepeljenek, amikor azonban munkatársaival kellett egyeztetni kiderült, hogy még egy-egy témában sincs idejük a megfelelő felkészülésre. Ebből nyilvánvaló lett, hogy a bevezetésben a továbbiakban kizárólag a (jó értelemben vett) szakmai marketing oldalán rendelkezésre álló eszközökre lehet és kell támaszkodni.

Gyorsan kellett dönteni. Meg kellett keresni a piacon aktuálisan kapható, legjobb .NET könyvet, és gondoskodni annak lehető legrövidebb átfutási idejű fordításáról. Erre a célra a Microsoft Press kiadásában David S. Platt, egy nemzetközileg ismert, ún. evangélista tollából készült, „Bemutatkozik a Microsoft .NET” c. könyv tűnt ígéretesnek, mert ez a Visual Studio .NET Béta 2 ún. korai változatának megfelelően készült, és így, az akkor megjelent Béta 2-vel egy csomagba lehetett tenni. A Szak Kiadó vállalkozott arra, hogy mindössze három hónap alatt megjelenteti a magyar változatot. Így semmi akadálya nem volt annak, hogy az alapozó szakmai konferencia anyagának video-felvételével egyetemben egy komplett

„*.NET felkészülési csomag (.NET Readiness Kit)*” készüljön el, és hogy minden érintett fejlesztő teljesen ingyenesen, a maga erejére támaszkodva és a neki legmegfelelőbb időbeosztásban megkezdhesse a felkészülést.

És az eredmény? A csomaghoz 3000 példányban megrendelt könyvön túlmenően maga a kiadó is eladott további 700 példányt (2002. júliusáig) a szokásos könyvkereskedelmi csatornákon keresztül, így legalább 4000 főre tehető azon szakemberek száma, akik részt vettek ebben a felkészülésben. Ennyi embert bevezető oktatással, még ha ingyen adjuk, akkor sem lehetett volna elérni. A csomag fajlagos költsége ugyanakkor kellően alacsony volt ahhoz, hogy a Microsoft Magyarország a .NET technológiák bevezetésére biztosított marketing költségvetéséből fedezni tudta annak előállítását és disztribúcióját. Legutóbb azt is sikerült megtudni, hogy mennyire használták ezeket az anyagokat a fejlesztők. A csomag egyes részeinek (könyv, konferencia, Béta 2) használatára vonatkozó kérdések összesítése azt mutatta, hogy több mint 80%-uk érdeklődésben felhasználta a teljes csomagot, tehát legalább részben használt minden részt.

Természetesen egészen más kérdés az ismeretek elsajátításának mértéke. A csomag, még érdemi felhasználás esetén is, csak a lényegi ismeretek elsajátításának lehetőségét biztosítja. További szakmai marketing eszközökre volt tehát szükségünk ahhoz, hogy támogassuk a begyakorlást, remélve ezzel a jobb elsajátítást, és a szakmai ismeretek elmélyítésében való előrelépést. A felkészülési csomag megjelenését (2001. szeptember) követő lépéseket ennek megfelelően tettük meg.

Gyakorlati segédlet

2001. november 27-én megrendezésre került a

„Visual Studio .NET a gyakorlatban” c. fejlesztői konferencia, mégpedig a következő témákban:

- Fejlesztői szekció:
 - A Visual Studio .NET programozási nyelvei
 - Adatkezelés, XML és a Visual Studio .NET
 - Web alkalmazások fejlesztése
 - XML Web szolgáltatások Visual Studio .NET-ben
 - Windows alkalmazások készítése
- Tervezői szekció:
 - Adatmodellezés üzleti szabályok alapján
 - Vizuális architektúra tervező eszközök
 - Alkalmazásfejlesztés vállalati sablonokkal
 - Web szolgáltatások tesztelése
 - A .NET biztonsági szolgáltatásai

Erre a Microsoft Corporation által, a világban tevékenykedő MSDN regionális igazgatók igényei szerint és velük együttműködésben kidolgozott, és a magyar MSDN csapat által hazai viszonyainkra adaptált rendezvényre egy teljesen saját fejlesztésű workshop-ot építettünk, melynek résztvevői a már .NET fejlesztéseket végző, vagy éppen megkezdni szándékozó szakemberek voltak. A 2002. januárjában rendezett workshop-ok tapasztalatai alapján a dr. Charaf Hassan vezette MSDN csapat kidolgozott egy komplett projekt praxist bemutató multimédiás csomagot „*NET projekt a gyakorlatban*” címmel. Ennek két modulja (Web alkalmazás, illetve Biztonság) már 2002. április 16-án megjelent a Számítástechnika c. szakmai hetilap nyolc oldalas Visual Studio .NET mellékletéhez csatolt CD-ROM-on, mint illusztráció.

A teljes, 10 modulból álló projekt gyakorlatot csak három hónap múlva, júliusban tudtuk kiadni, mivel – többek között – kiderült, hogy túlzottan sok CD-ROM-ra lenne szükség az elhelyezéséhez, és ezért a lényegesen nagyobb kapacitású DVD-t kellett választanunk. A minőségi DVD gyártás pedig jóval nagyobb körültekintést és hosszabb

átfutási időt igényel.

A modulok a következők:

- Tervezés
- Visual Basic.NET programozás
- Visual C# programozás
- Adat- és XML kezelés
- Web alkalmazás
- Windows alkalmazás
- Biztonság
- Framework lehetőségek kihasználása
- Web szolgáltatások
- Tesztelés

Minden modulhoz lényegre törő, de alapos szakcikk készült, mely megvilágítja a szükséges háttérismereteket. Ezt követően részletes útmutató található, amely az adott feladat megoldásának gyakorlatát részleteiben bemutatja. Ugyanezt a gyakorlatot mindenki „élőben”, saját számítógépének képernyőjén is megtekintheti, mégpedig pontosan ugyanolyan képi minőségben, mintha a saját gépén futó Visual Studio .NET-ben neki magának mutatná be azt a demonstrátor (igaz, ehhez ügyelni kell arra, hogy a Windows Media Player full screen-ben legyen). Mivel minden egy tipikus, háromrétegű alkalmazás példáján keresztül, és a fejlesztési munka gyakorlati fázisainak megfelelően kerül bemutatásra, a multimédiás anyag segítségével a gyakorlati készségek szintjén elsajátítható a Visual Studio .NET használata.

A kidolgozás alapelve az önjáró (self-paced) jelleggel történő elsajátítás maximális támogatása volt. Ez olyan jól sikerült, hogy a multimédiás anyag az eredeti rendeltetésén túli célokra is kiválóan alkalmas. Így például alkalmazható a fejlesztő csoportok tanfolyami rendszerű, belső képzéséhez is. A maximális felhasználás haté-

konyságának jó példája az, amikor a tanfolyammal megbízott, belső fejlesztő:

- csak slide-okat készít,
- a demonstrációkat teljesen a projekt gyakorlatban található felvételekre bízta,
- a tananyag a projekt gyakorlat kinyomtatott változata,
- az önálló gyakorláshoz pedig az abban található útmutató ad kész „munkafüzetet”,
- az előadások közbeni jegyzetelés csak minimálisan szükséges, mivel szinte jegyzetszerű szócikkek kísérik az egyes modulokat,
- a belső előadónak tehát csak a tisztán előadói és konzulensi feladatokkal kell megbirkóznia.

Az „élő” video-demonstrációk önmagukban is igen hasznosak. Nem kell külön demonstrációt készíteni ahhoz, hogy az egyes fejlesztési feladatok hatékony támogatását bemutassuk. Emellett az egyes videofelvételek azt is jól mutatják, hogy egy gyakorlott fejlesztő milyen gyorsan végez a teendőikkel. Elegendő csak a videofelvétel időtartamára tekinteni (bár a demonstrátorok között érthetően érezhető az ütembeli különbség, van aki olyan gyorsan mondja el, hogy mit csinál, amilyen gyorsan beszéd nélkül fejlesztene, van aki arra is tekintettel van, hogy ez egy előadás). Ennél meggyőzőbb bizonyíték aligha lehet arra, hogy mennyire példa nélkül álló az ún. Rapid Application Development (RAD) terén elért színvonal a Visual Studio .NET-ben. Így a döntéshozóknak is „kézzelfoghatóvá” válnak az egyébként csak marketing felhajtásnak tűnő kijelentések.

Végezetül akár önálló tanulással, akár házon belüli, csoportos fejlesztés keretében sajátította el valaki a .NET projekt gyakorlatot a későbbi, már éles fejlesztési munka során is visszanezheti a gyakorlat bármelyik részét, amennyiben valamilyen fejlesztési technikában már nem lenne kellően magabiztos. Így akár egy mindig rendelkezésre

álló háttértámogatást is nyújt a projekt gyakorlatot tartalmazó DVD. Éppen azért kellett CD-ROM-ok helyett DVD-t alkalmazni, hogy egyetlen lemezen elférjen a teljes gyakorlat, és így a fenti összetett és sokrétű felhasználáshoz akár mindig rendelkezésre állhasson.

Megfelelő bevezető videofelvételek és a korábbi .NET felkészülési csomag alapkonferencia előadásainak elhelyezésére is volt még hely, ezért azok is ugyanerre a lemezre kerültek, mivel a projekt gyakorlat feltételezi azok ismeretét. Ugyancsak el lehetett még helyezni ide a 2002. május 7-i bejelentési konferencia két, üzleti jellegű elemeket is tartalmazó előadásának felvételeit. Hasonlóképpen elért még a *Számítástechnika* c. lap április 16-i mellékletének teljes anyaga.

Ezzel a bevezetéshez szükséges szakmai minimum mellett a megfelelő piaci és stratégiai vonatkozásokat, valamint az alapvető termékinformációkat is kész, könnyen emészthető információs anyagok formájában megtaláljuk egy helyen. Így a döntéshozók is jól kiszorgálhatók erről az egyetlen lemezről. A fejlesztőknek mindig késznek kell lenniük a megfelelő döntéshozói tájékoztatásra. Ehhez azonban nem várhatjuk el tőlük, hogy holmi lemezeket keresgéljenek. Készen ott kell lenni az információnak a fejlesztők által leggyakrabban használt lemezen. Így alakult ki az összesen 4,3 Gb-át terjedelem.

Teljes induló készlet a felsőoktatásnak

Ezt már csak egy második DVD-vel kellett kiegészíteni ahhoz, hogy előálljon egy komplett induló készlet, az ún. **Starter Kit**. Ide kerültek a korábban már említett „Visual Studio .NET a gyakorlatban” c. fejlesztői konferencia előadásainak videofelvételei. Ugyancsak itt került elhelyezésre az ún. általános platform kiterjesztések

jegyében rendezett, újabb fejlesztői konferencia (2001. december 3.) előadásainak felvételei is. Nemzetközi előadók – szinkrontolmácsolásban – a következő témákkal szerepeltek itt:

- Mit értünk a .NET-en?
- Ami a [WebMethod]-on túl van
- Bevezetés a .NET MyServices-be
- A .NET Framework lényege
- .NET a mobil eszközökhöz.

A második DVD-n található a 2002. január 22-én rendezett Alkalmazási Fórum két előadása is:

- A Microsoft .NET az üzletfejlesztési tanácsadó és szervező szemével
- A Microsoft .NET lehetőségei a szoftverfejlesztési üzlet szempontjából.

És végezetül itt került elhelyezésre a nemzetközileg jól ismert, Microsoft-tól független evangelista, David Chappel 2002. február 8-án „*Ismerkedés a .NET-tel*” témában rendezett, komplett egy napos szemináriumának felvétele, magyar szinkrontolmácsolással.

Ez a második lemez, mint látható, előadások segítségével a lehető legalaposabban és legsokoldalúbban ismerteti a Microsoft .NET-et. Mivel erre hely is volt itt is elhelyeztük az első lemez legfontosabb előadásait. Ezzel annak a fejlesztőnek, vagy szakembernek a maximális kényelmét kívántuk biztosítani, akinek saját, célzott előadást kell készítenie fejlesztőknek, szakembereknek, vagy éppenséggel döntéshozóknak. Amennyiben valamelyik előadás ezen a lemezen egy az egyben felhasználható, akkor ez természetesen még könnyebb – csak le kell játszani a közönségnek, akár néhány slide erejéig. Ezért is kerültek erre a lemezre, és nem az elsőre, a „Visual Studio .NET a gyakorlatban” c. konferencia előadásai.

Az „Induló készlet a .NET alapú fejlesztések megkezdéséhez” megnevezés tehát olyan módon

is segíti a bevezetést, hogy egészen a döntéshozóig terjedően nyújt magas fokon kidolgozott előadási tartalmat, amelynek testreszabása is könnyen elvégezhető. Az induló készletet a www.developer.hu címen keresztül igényelhetik a fejlesztők. Az eszközzel még nem rendelkező fejlesztők a Visual Studio .NET 60 napos próbaváltozatát is megkapják. Semmi akadálya tehát annak, hogy mindenki kellő gyakorlatot szerezzen, mielőtt megkezdéné a .NET alapú, éles fejlesztést.

Ma már **a cégek azonnal hadra fogható tudást várnak el a frissen végzett hallgatóktól** is. Hogyan lehet a .NET technológiát jobban megismerni és megtanulni a felsőfokú képzésben még csak éppen résztvevőknek? A kérdésre adott válasz azonos. Úgy, hogy kipróbálják. A Magyar Kormány és a Microsoft Magyarország által megkötött Campus szerződés értelmében minden hallgató és oktató rendelkezik Visual Studio.NET licenccel, így nem kell mást tenni, mint letölteni a www.campus.hu címről, vagy telepítőkészletet rendelni szintén ezen az oldalon keresztül.

A mai informatikai fejlődéssel egy felsőoktatási tanmenetnek nehéz lépést tartani, így komoly szerepe van az öntanulásnak. A European Microsoft Windows Academic Program igyekszik az ehhez szükséges információkat biztosítani a www.emwac.bme.hu oldalon. A Microsoft IT Academy Program pedig indexbe felvehető tantárgyként oktat Hivatalos Microsoft Tananyagokat, melynek során egy 120-150 ezer Ft értékű tanfolyamon vehetnek részt a hallgatók ingyenesen. További információk a www.emwac.bme.hu oldalon található.