



KULCSSZAVAK HELYETT

A net mélyére ásni, bugyiraiban tényleg intelligensen kutakodni még a Google sikere ellenére is komoly problémákat okoz. Van-e megoldás? » 19. oldal

TESZT A KEZDETEKTŐL

A módszertan neve: acceptance-test-driven development (ATDD), azaz magyarul kb. átvételi vizsgálatokon alapuló fejlesztés. » 15. oldal

495
forint

SZÁMÍTÁSTECHNIKA

ICT-STRATÉGIA DÖNTÉSHOZÓKNAK • WWW.COMPUTERWORLD.HU
ALAPÍTVÁ 1969 • 2011. OKTÓBER 11. • XLII. ÉVFOLYAM 41. SZÁM



COMPUTERWORLD

TESZTELJÜNK A MINŐSÉGÉRT

A költséghatékonyság és a gyors reakcióidő kényszerít sok esetben a szoftver tesztelésére fordított erőforrások sínylik meg. Pedig a hibák kijavításának ára a program kiadása után akár többszörösére is növekedhet, nem is beszélve a felhasználónál felszínre kerülő sérülékenységek és hibák miatti presztizsvesztésekről. Hogyan békíthetjük ki a költséghatékonyságot a minőséggel?

Összeállításunk a 10-12. oldalon



917705871151006 11041

DVD Authoring
CD, DVD sokszorosítás
Egyedi CD, DVD írás
Csomagolás és logisztika



H-8000 Székesfehérvár
Aszalvölgyi u. 7.
Tel.: +36-22/533-571
Fax.: +36-22/533-599
E-mail: vtcd@vtcd.hu www.vtcd.hu

KREATÍV ÉS
KÖLTSÉGHATÉKONY
VIDEÓK

saját vírusvideók, reklámfilmek és PR spotok
kis cégeknek is elérhető áron
a forgatókönyvtől az utómunkáig
profi stábtól, megbízható referenciákkal

AKTUÁLIS

05 HYDE TECH CORNER
Ezen a héten Szekeres Viktor, a Gloster telekom Javító és Szerelő Kft. ügyvezető igazgatója és Krasznay Csaba, a HP Magyarország IT-biztonsági tanácsadója kommentálja a hét híreit, eseményeit.

06 ELHUNYT STEVE JOBS
Az Apple egyik társalapítója és volt vezérigazgatója 56 éves korában távozott.

07 PARADIGMAVÁLTÁS? A SZIGNATÚRÁK VÉGE

08 A NAGY ADAT-KÉSZÜLÉK
Nagy appliance-bejelentések és ködös cloud-stratégia a múlt héten lezajlott OpenWorld 2011-en.

FÓKUSZ

10 OLVASSUNK A SOROK KÖZÖTT!
A költséghatékonyság és a gyors reakcióidő kényszerít sok esetben a szoftver tesztelésre fordított erőforrások sínlyik meg. Pedig a hibák kijavításának ára később akár többszöröse is növekedhet, nem is beszélve a felhasználónál felszínre kerülő sérülékenységekről. Hogyan békíthetjük ki a költséghatékonyságot a minőséggel?

ÜZLET

13 A SZOFTVERTESZTELÉS OKTATÁSA HAZÁNKBAN
Vajon hol és hogyan képezik azokat a szakembereket, akik a megfelelő kreativitással és szakértelemmel tölthetik be a ma még gyakran üresen álló pozíciókat?

14 AZ ISTQB MAGYARORSZÁGON

TECHNOLÓGIA

15 TESZTELNI AZ ELSŐ PILLANATTÓL

16 „HAGYOMÁNYOS” VS. KÖZÖSSÉGI TESZTELÉS
Mire jó a közösségi szoftvertesztelés? Cikkünkben az előnyökről és a hátrányokról egyaránt lerántjuk a leplet.

18 FELHASZNÁLÓKHOZ IGAZODVA
A fejlesztők újabb és újabb adaptív felületekkel bővítik az EuroOffice irodai programcsomagot. A jövőben a netbookok és a mobil eszközök kerülnek a fókuszba.

19 KÉRDÉSEK KULCSSZAVAK HELYETT
A világháló mára mindennapjaink részévé vált, átalakította az emberi kommunikációt, társadalom- és gazdaságformáló szerepe felbecsülhetetlen.

HORIZONT

21 RENDSZERGAZDÁK HŰSVÉTI NYÚLKÉNT
A vállalatoknak érdemes támogatni az önkéntes munkavégzést, mivel profitálhatnak belőle.

ÁLLANDÓ ROVATAINK

04 VÉLEMÉNY
Fabiányi Gábor: Szoftvertesztelés tesztközelből – A tesztelés elhanyagolásának következményei komolyak lehetnek. Mire figyeljünk mindenképpen oda?

05 HÍRMOZAIK

06 ESEMÉNYEK

06 SZEMÉLYI HÍREK



IMPRESSZUM COMPUTERWORLD-Számítástechnika

Kiadja IDG Hungary Kft.
1075 Budapest Madách I. út 13-14. A ép.
HU ISSN 0237-7837 Postacím: 1374 Budapest 5, Pf. 578
Internet: www.idg.hu
Bankszámlaszám 10300002-20328016-70073285
Felelős kiadó Bíró István ügyvezető – ibiro@idg.hu
Műszaki vezető Babinecz Mónika – mbabinecz@idg.hu
Nyomás és kötészet D-Plus Kft.
1037 Budapest, Csillaghegyi út 19-21.
Ügyvezető igazgató Németh László

SZERKESZTŐSÉG

Főszerkesztő Dervenkár István – idervenkar@idg.hu
Vezető szerkesztő Odrovics Szonja – szodrovics@idg.hu
Szalay Dániel – dszalay@idg.hu

Olvasószervező, korrektor Sz. Erdős Judit – jerdos@idg.hu

Munkatársak Dávid Imre – idavid@idg.hu
Egri Imre – iegri@idg.hu
Kis Endre – ekis@idg.hu
Mallás Judit – jmallasz@idg.hu
Szlágyi Szabolcs – sszlagyi@idg.hu
Tóth Livia – ltoth@idg.hu
Vass Enikő – evass@idg.hu

Szerkesztőségi ügyelet Cseresznye Anita – acseresznye@idg.hu
Telefon: 577-4302, fax: 266-4343
Internet: www.computerworld.hu

Újságíróink szakmai képzésének háttérét a NetAcademia Oktatóközpont biztosítja. www.netacademia.net

TIPOGRÁFIA

Berényi István – iberenyi@idg.hu

HIRDETÉSFELVÉTEL

Hirdetési igazgató Melovics Csaba – csmelovics@idg.hu
Telefon: 577-4310, fax: 266-4274
Lapreferens Rodriguez Nelsonné – irodriguez@idg.hu
Telefon: 577-4311
Kereskedelmi asszisztens Bohn Andrea – abohn@idg.hu
Telefon: 577-4316, fax: 266-4274
e-mail: keriroda@idg.hu

TERJESZTÉS ÉS ÜGYFÉLSZOLGÁLAT

Terjesztési igazgató Babinecz Mónika – mbabinecz@idg.hu
Telefon: 577-4301, fax: 266-4343
MediaShop: mediashop.idg.hu
e-mail cím: terjesztes@idg.hu

MARKETING

PR-munkatárs Kovács Judit – jkovacs@idg.hu

JOGI KÖZLEMÉNYEK

Szerkesztőségünk a kéziratok lehetőségei szerint gondozza, de nem vállalja azok visszaküldését, megőrzését.

A COMPUTERWORLD-ben megjelent valamennyi cikket (eredetiben vagy fordításban), minden megjelent képet, táblázatot stb. szerzői jog védi. Bármilyen másodlagos terjesztésük, nyilvános vagy üzleti felhasználásuk kizárólag a kiadó előzetes engedélyével történhet. A hirdetések a kiadó a legnagyobb körültekintéssel kezeli, ám azok tartalmáért felelősséget nem vállal.

TERJESZTÉSI, ELŐFIZETÉSI, ÜGYFÉLSZOLGÁLATI INFORMÁCIÓK

A lapot a Lapker Rt., alternatív terjesztők és egyes számítástechnikai szaküzletek terjesztik. Előfizethető a kiadó terjesztési osztályán, az InterTicketnél (266-0000 9-20 óra között), a postai kézbesítőknel (06/80-444-4444; hirlapelőfizetes@posta.hu, fax: 303-3440)
Előfizetési díj egy évre 16 440 forint, fél évre 8220 forint, negyed évre 4110 forint.

Lapunkat a MATESZ auditálja

Olvasóink szokásait a Nemzeti Médiaanalízis méri fel.

A Computerworld az IVSZ hivatalos médiapartnere.



A szerkesztőségi anyagok vírusellenőrzését a NOD32 Antivirus programmal végezzük, amelyet a szoftver magyarországi forgalmazója, a Sicontact Kft. biztosítja számunkra.



Mennyit bukik az Amazon a táblagépen?

Centre pontosan tudjuk, mennyit veszít Kindle Fire-en: 10 dollár 63 centet. Ám valószínű, hogy áttételesen sokkal többet nyer majd.

[» computerworld.hu/cikk/kindlefire](http://computerworld.hu/cikk/kindlefire)

iPhone 5 helyett egyelőre 4S

A hardvere erősebb, mint az iPhone 4-é, operációs rendszere pedig az iOS 5. A tőzsde azonban csalódott volt: gyengültek az Apple-részvények.

[» computerworld.hu/cikk/iphone-s4](http://computerworld.hu/cikk/iphone-s4)

Földi sugárzású mobil-TV

Okostelefonokra szánt tévéadásokkal, játékokkal és e-magazinokkal hódítanak meg a felhasználókat.

[» computerworld.hu/cikk/mobil-tv](http://computerworld.hu/cikk/mobil-tv)

Digitális CW, PCW és GS Honeycombon

Lapjaink digitális változatának terjesztésével foglalkozó Digitalstand Andorid Honeycombon is elérhetővé tette lapkinálatát.

[» computerworld.hu/cikk/digi-cw](http://computerworld.hu/cikk/digi-cw)



Szoftvertesztelés tesztközelebből



Fabiányi Gábor
marketingmenedzser,
Kürt Zrt.

A programfejlesztési projektek egyik kulcskérdése a tesztelés, mégis, súlyához képest általában kevés figyelmet és erőforrást szentelnek neki. A következmények komolyak lehetnek. Nem feltétlenül kell nagy méretekben gondolkodni, kisebb projektek esetén is alapvető fontosságú eleme a fejlesztésnek a megfelelően megtervezett, kontroll alatt tartott és persze jó minőségben és alapossgal végzett tesztelés.

Sokszor még nem világos az érintetteknek, hogy pontosan mit is kell érteni tesztelés alatt. Sokan hiszik azt, hogy a tesztelés a fejlesztési folyamat végén sorra kerülő átadás/átvételi ellenőrzés, ahol egyrészt a funkciók működését vizsgálják, másrészt azt, hogy a végeredmény többé-kevésbé megfelel-e a megrendelő által elvártaknak.

Valójában a tesztelésre, illetve a különböző célú (kódelemző, teljesítményfigyelő, funkcionális, regressziós stb.) teszteljárásokra elejétől kezdve, a fejlesztés teljes folyamatában szükség van. Ezzel elkerülhető, hogy a projekt végén szembesüljünk olyan problémákkal, amelyek már nem, vagy csak nagyon nagy ráfordítással orvosolhatók.

Vegyük sorra azokat a szempontokat, amelyekre mindenképpen ügyelni kell, ha egy szoftverfejlesztési projektbe fogunk, és a tesztelést a helyén szeretnénk kezelni!

Szakember. A munkát tesztelésben jártas szakemberekre kell bízni. Egy komolyabb szoftver tesztelése rendkívül komplex feladat, a különböző fázisok különböző feladatai eltérő kompetenciákat követelnek meg, tehát nem elegendő, ha például kódtesztelő szakemberre bízunk az egész folyamatot.

Menedzsment, dokumentálság. Nagyon fontos, hogy legyen valaki, aki a tesztelési folyamatot,

a szerteágazó feladatokat összefogja, menedzseli, és aki nagyobb léptékű rálátással bír a projektre. Erre a pozícióra is teszteléshez értő szakember kell, pusztán projektmenedzsment-tapasztalatoknál többre van szükség.

Megfelelő időt szükséges szánni a tesztelési terv elkészítésére, amely részletesen tartalmazza, hogy mely fázisban, mikor, melyik teszteljárást kell alkalmazni, milyen eszközökkel és erőforrásokkal. A megfelelő tesztelési terv sok időt és energiát spórol meg a későbbiekben. Ugyanilyen fontos a teszteredmények jegyzőkönyvekkel történő dokumentálása, és azok visszavezetése a projektbe.

Átgondolt, kimerítő tesztelés. Általában elvárható lenne, hogy a projekt leelejétől folyamatosan fusson az elkészült kódok, részek tesztelése (ezt biztosítja például a KÜRT szoftverfejlesztés-minőségbiztosítási módszertana, amely a FrontEndArt valós statikus forráskód-elemzését is tartalmazza), de ez többnyire elmarad. Azonban ennél is nagyobb hiba szokott lenni, hogy egyáltalán nem definiálják, hogy a projekt mely pontján kell elkezdenni a tesztelést.

Legalább ilyen lényeges a tesztelés „mennyisége”. A tesztelési tervben előírtak szerint kell minden fázisban az adekvát tesztmetódust alkalmazni, és ezekkel minél átfogóbb, a készülő termék minél nagyobb ke-

resztmetszetét górcső alá vevő vizsgálatot végezni.

Már egy nem túl komplex szoftver esetén is szinte végtelen számú tesztesetet, vagyis szenáriót kellene megvizsgálni, ha le akarjuk fedni az alkalmazás teljes funkcionalitását, környezeti összefüggéseit. Mivel ez lehetetlen, különböző megoldások állnak rendelkezésre, a feladatot megkönnyítendő. Ilyenek például az automatizált tesztelőeszközök, nagy létszámú tesztelői csapat alkalmazása, teszt-optimalizáló módszerek, de meg kell említeni, hogy sokat segítene az is, ha szakszerűen határoznák meg a feltétlenül tesztelendő eseteket.

Lényegesegek a teljesítménytesztetek, amelyeknél ügyelni kell arra, hogy a terheléses tesztet végző eszközöket hol helyezzük el az informatikai rendszerben. Például olyan szoftver esetén, amelyhez internetes hozzáféréssel csatlakozhatnak a felhasználók, nem lehet az internet és a fejlesztett rendszer közé pozicionálni a tesztet végző eszközöket, mert így a teszt hamis eredményt adhat.

A tesztek egy harmadik fontos csoportját alkotják a regressziós próbák, amelyek azt vizsgálják, hogy egy újonnan kifejlesztett modul hogyan működik együtt a korábban fejlesztett modulokkal, illetve a meglévő rendszerekkel. Bonyolultsága és költségessége miatt ezt a tesztfajtát is gyakran mellőzik. A költségek csökkentése érdekében itt is hatékony

megoldás lehet a reprezentatív esetek, illetve lényeges szoftverelemek tesztelése.

Infrastruktúra (futtatókörnyezet, tesztelőeszközök, tesztelési menedzsmenteszközök stb.). Szoftverek tesztelésénél ez mindig kardinális kérdés, éles rendszeren nem szabad tesztelni, a megfelelő infrastruktúra általában nem nagyon áll rendelkezésre, csak időszakosan van rá szükség, könnyen változtathatónak kéne lennie és drága.

Ezekre a problémákra megoldást jelenthet egy felhőalapú tesztkörnyezet bérbevétele, amely gyakorlatilag az összes kérdésre választ ad.

Ha nem lehetséges tesztelni. Akadnak olyan rendszerek, amelyeknél a folyamatos, gyors fejlesztések mellett nem megvalósítható a tesztelés. Ezen projektek-nél együtt kell élni a várható hibákkal, az üzemeltetésnek fel kell készülnie a gyakoribb telefonhívásokra, szükség esetén mentések igénybevételére, régebbi szoftververziók gyors visszaállítására. Az ilyen szoftverek esetén költségesebb lehet az üzemeltetés, mert olyan üzemeltetési folyamatokat kell kialakítani, működtetni és tesztelni, amelyek az átlagosnál jobb incidenskezelést, mentésvisszatöltést vagy akár katasztrófahelyzet-kezelést biztosítanak. Külső szakértő bevonása nélkül számos szervezet nem képes ilyen folyamatok megfelelő szintű kialakítására. 🚩

Hyde Tech Corner

Ezen a héten Szekeres Viktor, a Gloster telekom Javító és Szerelő Kft. ügyvezető igazgatója és Krasznay Csaba, a HP Magyarország IT-biztonsági tanácsadója kommentálja a hét híreit, eseményeit.

Összeállította: Tóth Livia

Heti összeállításunkból megtudhatják, mi áll a hazai lakossági optikai ellátottság 6 százalékos penetrációja mögött, valamint az is kiderül, hogy mi a helyzet a HTC legújabb biztonsági hibájával.

6 százalékos a hazai üveg-szálas penetráció

Jelentősen nőtt Európában az FTTH, vagyis az otthonig elérő ultragyors üveg-szálas technológia előfizetőinek a száma – derült ki az FTTH Council Europe legfrissebb adataiból, amit a szervezet a Párizsban megrendezett Broadband World Forumon tett közzé a múlt héten.

computerworld.hu/cikk/6-szazalekos-penetracio

SZEKERES VIKTOR, A GLOSTER TELEKOM JAVÍTÓ ÉS SZERELŐ KFT. ÜGYVEZETŐJE

A hazai lakossági optikai ellátottság nagy növekedés előtt álló piac, jelenlegi 6 százalékos elterjedtségével még csak kevesek kiváltsága. Az előnyök lakossági oldalról nyilvánvalók: soha nem látott mértékű sáv-

szélesség-emelkedés és minden más technológiához képest stabilabb internetcsatlakozás. Egy ilyen magas beruházás ellenben csak jelentős hozzáadott értékkel rendelkező szolgáltatások nyújtásával együtt térülhet meg; ilyen lehet egy HD-s IP-tévé és online videotéka, lakossági videotelefonálás vagy videós őrző-védő szolgáltatás. Lehetővé válna a hang- és videokapcsolattal támogatott valódi távmunka is, ami növelhetné az esélyegyenlőséget az elmaradottabb régiókban. Emellett olyan, lakossági felhőalapú szolgáltatások előfeltétele is a megfelelő sáv szélesség, mint például a Dropbox vagy a Box.net, ahol sok-sok gigabájnyi adatot (fotók, videók, dokumentumok) tárolhatunk biztonságosan, úgy, hogy azokat olyan gyorsan érhetjük el, mintha csak otthon, a saját gépünkön tárolnánk a fájlokat.

Súlyos biztonsági hiba HTC-telefonoknál

Az Android-alapú mobilkészülékekkel foglalkozó Android Police weboldalán egy olyan bejegyzés jelent meg, amely rövid időn belül komoly visszhangot váltott ki világszerte. Három kutató ugyanis a HTC egyes okostelefonjainak esetében jelentős biztonsági rést fedezett fel, amelynek kihasználásával viszonylag egyszerűen kérdezhetők le adatok az érintett készülékekről. A HTC jelenleg is vizsgálja a problémát. computerworld.hu/cikk/biztonsagi-hiba-htc

KRASZNAY CSABA, A HP MAGYARORSZÁG IT-BIZTONSÁGI TANÁCSADÓJA

Az okostelefonokkal kapcsolatban a biztonsági szakembe-

rek az egyik legkomolyabb fenyegetésnek már régóta a mindent lefedő adatgyűjtést tekintik. Noha feltételezhetjük, hogy az összegyűjtött információkat anonim módon kezelik, és az adatgyűjtés elsődleges célja a felhasználói élmény növelése, valamint a szolgáltatások fejlesztése, mégis felvetődik a kérdés, hogy vajon miért nem tudjuk pontosan, milyen információkat szerez meg rólunk a gyártó, fejlesztő? Ez talán a modern adatvédelem egyik legérdekesebb kihívása. Elképzelhető, hogy az ilyen hírek fogják kikényszeríteni azt, hogy a gyártóknak előbb-utóbb adatkezelési hozzájárulást is kell kérniük a vásárlóktól.



„Vajon miért nem tudjuk, mit tudnak meg rólunk?”

Krasznay Csaba
HP MAGYARORSZÁG

A másik érdekesség a hírrel kapcsolatban az, amit *Rik Ferguson* a hivatkozott cikkben is feszeget: a mai alkalmazásfejlesztőknek figyelemmel kell lenniük arra, hogy az alapvető biztonsági funkciókat beépítsék a programjaikba. Ezek egyike a megfelelő autentikáció, különösen akkor, ha ennyire szenzitív adatokról van szó. A világon rohamosan terjednek az oltó, ellenőrizetlen szoftverek, tehát a felhasználók érdekében különösen fontos lenne a kiskalkulációk megfelelő védelme, legyen az gyártói vagy éppen bármilyen más, a Marketből leltethető program. 🇳🇵

HÍRMOZAIK

Díjjal startolt

A BalaBit napokban megjelent High-Speed Reliable Logging naplózó technológiája forradalmasítja a globális naplózópiacot, másodpercenként akár 1 000 000 logüzenetet is képes kezelni. Az új technológia hivatalos megjelenésével egy időben rögtön elnyerte az „Év legjobb magyar gyártói innovációja” díjat az ITBN konferencián.

Linux-születésnap

Október 5-én volt a 20. évfordulója annak, hogy *Linus Torvalds* közzétette a Linux operációs rendszer első hivatalos változatát. A hobbiprojektként indult megoldáson mára ezernél is több fejlesztő dolgozik, a kernel pedig 13 milliónál is több kódsort tartalmaz. A Linux 1.0-s változata 1994-ben jelent meg, és ugyanebben az évben indult útjára a SUSE Linux is, amelyet 2003-ban vásárolt fel a Novell.

Önkéntes iskolatuning

Az Európai Önkéntesség Éve alkalmából a CNW Zrt. 46 munkatársa szeptember 23-án a komáromi Kempelen Farkas Alapítványi Középiskola megépítésében vállalt szerepet. A csapat egyik fele a konkrét felújítási munkálatokban vett részt, míg a másik fele – a CNW munkatársainak szakmai képzettségére építve – a Kempelen informatikai rendszerének karbantartását, felújítását, interaktív táblák üzembe helyezését valósította meg, valamint bemutatót tartott az e-learning előnyeiről.

REGISZTRÁLJON

Ha szeretné hétről hétre a legfontosabb szakmai résztvevőkhöz eljutni az Ön cégével kapcsolatos információkat, regisztráljon Céginfó szolgáltatásunkra oldalunkon.

ceginfo.computerworld.hu



„Óriási sáv szélesség-emelkedés és stabilabb internetcsatlakozás.”

Szekeres Viktor
GLOSTER TELEKOM KFT.

Elhunyt Steve Jobs – gyászolnak a legnagyobbak

Az Apple egyik társalapítója és volt vezérigazgatója 56 éves korában távozott.

Írta: **Wiezner István–Horváth Péter**

Az almás cég hivatalosan is bejelentette, hogy a legendás *Steve Jobs* tegnap örök nyugalomra tért. Nem tudni, hogy az Apple társalapítójának halálához egész pontosan mi vezetett, de 2004-ben hasnyálmirigyrákkal műtötték, 2008-ban pedig májátültetésen is átesett. A cégnél betöltött vezérigazgatói posztjáról hivatalosan augusztus végén mondott le, bár *Tim Cook* már korábban is számos alkalommal helyettesítette főnökét annak egészségügyi szabadságai alatt.

A bejelentésre számos fontos személy és vállalat reagált, illetve természetesen a Samsung is elásta egy pillanatra a csatabár-

dot, a vállalat szóvivője szerint Jobs „innovatív szellemére és jelentős teljesítményeire mindig emlékeznek fognak szer- te a világon”. Megemlíten- dő, hogy az Apple cupertinói alkalmazottjainak tegnap nem-

egyeniség volt, aki nagyon fog hiányozni. *Barack Obama* pedig megdöbbenését fejezte ki az Apple-vezér halálhíre kapcsán, hozzátéve: a világ egy igazi lát- nokot veszített el. Szerinte Jobs a jelenkori amerikai feltalálók

egyik legnagyobbaként elég bátor volt a többiek- től eltérő módon gondolkodni, ren- díthetetlenül hitt abban, hogy képes megváltoztatni a világot, és elég tehetséges volt ennek a kivitele- zéséhez is. „Steve gyakran mondta, hogy minden nap- ját úgy élte, mint- ha az az utolsó lenne. Ennek kö- zönhetjük a technológiát alapja-iban megváltoztató eredménye- it, életünk átalakulását” – tette hozzá az amerikai elnök.

Bill Gates arról mesélt, hogy 30 évvel ezelőtt találkozott és dolgozott együtt Jobsszal, bará- tók és egymást inspiráló ver- senytársak is voltak életük jelen- tős részében. „Nekünk, akiknek szerencsénk volt együtt dolgoz- ni vele, mindig is nagy megtisz- tetetés lesz a közös múlt. Na- gyon fog hiányozni” – tette hoz- zá a Microsoft első embere.

EDISON ÉS JOBS
Mark Zuckerberg számára Steve Jobs mentor és barát is volt egy- ben. „Köszönöm, hogy meg- mutattad, hogy amit építesz, az megváltoztathatja a világot” – tette hozzá a Facebook feje.
Bob Iger, a Disney vezérigaz- gatója szerint Jobs halálhíre azért nagy tragédia, mert si- kerei ellenére eddig úgy tűnt, hogy csak most indul be iga-



csak a volt vezetőjük halála mi- att volt örült napjuk, de egy darabig még az utcára sem me- hettek ki, mivel egy közeli bá- nya alkalmazottja lövöldözni kezdett – az ámokfutása során három embert megölt, hetet pedig megsebesített.

GYÁSZOL AZ IT-VILÁG

A rememberingsteve@apple.com címre írva bárki rész- vétét fejezheti ki. Sok híres- ség, cégvezető, művész és po- litikus már pár mondatban meg is emlékezett Jobs tragi- kus haláláról. A család hivata- los közleményében leírja, hogy Steve otthon hunyt el, szeretei körében, békésen távozott. „Köszönjük mindenkinek, aki imádkozott érte a betegsége alatt, és hálásak vagyunk a sok támogatásért, kedves szóért. Tudjuk, sokan gyászolnak most velünk” – teszik hozzá.

Ronald Wayne, az Apple társ- alapítója szerint Steve igazi

SZEMÉLYI HÍREK

Ali Shah



Vezetőt vált az Ericsson Magyar- ország. Az Egye- sült Államokból ér- kező *Ali Shab Ro- land Nordgren*

váltja a vezérigaz- gatói székben, aki Stockholmban, az Ericsson központjában folytat- ja tovább munkáját. Az új vezető 1996-ban, azt követően csatlako- zott az Ericssonhoz, hogy a virgi- niai George Mason Egyetem infor- mációtechnológia szakán doktorá- tust szerzett. 2007 óta töltötte be az észak-amerikai régió Stratégia és Marketing szervezetében a széles- sávú stratégiaigazgatói pozíciót.

Somfalvi Csaba



Új szakember csat- lakozott a Gemius Hungary-hez *Som- falvi Csaba* szemé- lyében, aki client service director po- zícióban irányítja

a cég teljes körű sales és key account tevékenységét. Somfalvi kilenc éve kezdte online szakmai karrierjét az Index.hu kötelékeiben, azóta pedig már dolgozott HF Media Zrt.-nél, majd a HVG-nél, ahol 2011 márci- usa óta a new business manageri po- zíciót töltötte be.

zán, kreativitása pedig valódi örökség a világ számára. *Rupert Murdoch* kijelentette: Steve Jobs volt generációjának legnagyobb cégvezetője. *Steven Spielberg* egyenesen *Thomas Edison*hoz hasonlította az Apple első emberének munkásságát, „aki a világot az emberek ten- nyerébe helyezte”.

John Lasseter és *Ed Catmull* a Pixar nevében adott ki nyi- latkozatot, amelyben Jobsot valódi látnoknak nevezik, aki már akkor látta a fantázi- át a cégükben, amikor még ők maguk sem. Az ő biztatásának köszönhetik mai sikereiket, mert mindig azt mondogatta: „csak csináljátok nagyszerűen”. Ezt tette ő is – minden nap. 🍌

ESEMÉNYNAPTÁR

Október 11–12. BALATONFÜRED
Internet Hungary 2011
» www.internethungary.com

Október 12. BUDAPEST
Ügyfélkapcsolat-kezelés workshop
» www.multisoft.hu

Október 12–14. RÖJTÖKMUZSAJ
Vezérigazgató Találkozó
» www.cebc.hu

Október 13–14. BUDAPEST
Hungarian Software Testing Forum
» <http://computerworld.hu>

Október 18–19. BUDAPEST
Budapest Calling
» <http://budapestcalling.hu>

További események
» www.computerworld.hu/esemenyek

Paradigmaváltás? Búcsú a szignatúráktól

Kristóf Csaba ■ A Webroot bejelentette azt a legújabb antivírus termékcsaládját, amellyel szakított a sokéves múltra visszatekintő szignatúraalapú technikák használatával, és a cloud technológiákra helyezte a hangsúlyt. Az antivírus-fejlesztő cégek is gyakran hangsúlyozzák, hogy a szignatúraalapú technikák önmagukban már nem képesek felvenni a küzdelmet napjaink gyorsan szaporodó kártékony programjaival szemben. A vírusok, férgek, trójaiak olyan nagy mennyiségben vannak jelen, hogy a szignatúra-adatbázisok nem képesek lépést tartani a trendekkel, legalábbis olyan módon nem, hogy közben ne korlátoznák észrevehető módon a számítógépek teljesítményét. Ezért kiegészítő technológiákra van szükség, amelyek a heurisztikus, viselkedés-, hírnév- és hardveralapú módszerektől egészen

a felhőalapú megoldásokig terjedhetnek.

A Webroot gondolt egy nagyot, és úgy határozott, hogy a kliensekről teljes egészében mellőzi a szignatúra-adatbázisokat. Noha már akadt olyan cég, amely régebben próbálkozott ezzel (például a Panda Security a Cloud Antivirussal), azonban még egyik vállalat sem érezte elérkezettnek az időt arra, hogy teljesen megváltjon a rég bevált technikáktól. A Webroot viszont mindent egy lapra tett, és az új, SecureAnywhere termékcsaládjával igyekszik bebizonyítani, hogy felhőalapú védelmi technológiája megérett az önálló életre. Azt azonban meg kell jegyezni, hogy a SecureAnywhere a cloud eljárások mellett erőteljesen épít a viselkedéselemzésre és a sandbox-technikákra is.

Drian Czarny, a Webroot termékmenedzsermentért felelős ale-

nke szerint amíg 2009-ben körülbelül 40 ezer kártékony kódot detektáltak, addig mostanában napi 100 ezer vírusmintával kell megbirkózniuk, amelyek ellen már nem lehet hatékonyan védekezni szignatúrákkal. Nemcsak a vírusok nagy száma miatt, hanem azért sem, mert a számítógépes kártevők a hagyományos védelmi vonalakat sokszor képesek megkerülni.

A SecureAnywhere egy 1 MB-nál kisebb méretű fájl formájában tölthető le a számítógépekre. A kliensprogram legfontosabb feladata, hogy folyamatosan kommunikáljon a Webroot szervereivel, és azok segítségével feltárja az ismert károkozókat. Az ismeretlen vagy gyanús alkalmazások kiszűrését pedig a háttérben futó, viselkedéselemző összetevők végzik.

A teljesítmény javítása érdekében a SecureAnywhere a telepítés után ellenőrzést hajt végre, amely

során feltérképezi a rendszeren lévő állományokat, majd ezt követően már csak a változásokra figyel. Amennyiben egy gyanús program letöltését észleli, akkor annak futtatását egy elkülönített, sandbox környezetben engedélyezi annak érdekében, hogy ki tudja értékelni a szoftver viselkedését. Ha azt észleli, hogy káros alkalmazásról van szó, akkor blokkolja annak működését, miközben a rendszerben nem következik be nemkívánatos változás.

Ha pedig a védett számítógép nem csatlakozik az internethez, és ezáltal a cloud-összetevők nem tudják elvégezni a feladatukat, a SecureAnywhere egy sandboxot épít fel, majd naplózza a programok működését. Amikor újra kapcsolódik a gép az internethez, akkor a SecureAnywhere ellenőrzi a cloud-adatbázist, és ha káros változás következett be, „visszagörgeti” a módosításokat. ■

Vállalati alkalmazások, másképp

Az üzleti siker kulcsa a lehetőségek felismerése és a cselekvés a megfelelő pillanatban. Egy kisvállalkozás vezetője még egymaga átlát mindent, amíg a cég mérete meg nem haladja a képességei, Excel táblái és jegyzetei korlátait. De tudja-e mindenki a cégnél, hogy melyik ügyfél mit rendelt és milyen határidővel? Tudja-e az értékesítő, hogy az addig eladott termékek alapján mit érdemes még ajánlani az ügyfélnek? Tudjuk-e, hogy egy ügyfélnek milyen panaszai, hibabejelentései voltak? Idejében eljut-e minden szükséges

szerezési adat a termelés és beszerzés tervezéséhez? Követhető-e a beszerzési folyamat? Átlátható információt kapnak-e a vezetők a cég működéséről?

Az értékesítési feladatok összehangolására és átláthatóságára CRM-rendszer jelent megoldást, amely egységesen nyilvántartja az ügyfeleket és elérhetőségeiket egy alkalmazásban. Jegyzi az ügyfelek részére történt rendeléseket, emellett figyelmeztet a rendeléssel kapcsolatos tennivalókra, fontos tudnivalókra is.

A cég működésének átfogó menedzselésére pedig az ERP- (vállal-

latirányítási) rendszer a jó választás. Biztosítja az üzleti folyamatok kezelését, a raktárkészlet és a termelés menedzselhetőségét, követi a beszerzési folyamatot a megrendeléstől a számla kifizetéséig. Átláthatóvá teszi a vállalatot, vezetői jelentéseket készít és elérhetővé teszi az információt akár az irodában, akár a mobilkészülökön. A jól informáltság versenyelőnyt jelenthet, hiszen növelhető az üzleti agilitás, és költségcsökkentés érhető el.

Mindez ma már kevesebbe kerül, mint egy rendszergazda fizetése, naprakész frissítésekkel,

folyamatos karbantartással, évente 0 nap szabadsággal.

Az IQSYS Zrt. beruházási költség nélküli CRM és ERP cloud szolgáltatást indít, használatárra nyos díjazással.

A szoftveres szolgáltatás és a kommunikációs vonal egy csomagban is megszerezhető, biztonságos és megbízható adattárolást, illetve az üzleti információk védelmet nyújtó szerverháttérrel.

További információt az IQSymposium – Operatív Informatíciotechnológia 2011 konferencián szerezhet. ■

IQsymposium
Operatív Informatíciotechnológia | 2011



Globális trendek - lokális stratégiák

2011. október 19. Regisztráció és további információ: www.iqsys.hu

- **Kollaboráció** – Az egymással együttműködő vállalati kulcsfolyamatok fejlesztése, javítása
- **Alkalmazások, alkalmazás platformok** – Egyedi és csomagalkalmazások helyben és felhőben
- **CRM** – Ügyfél Életút Modell, Ügyfélkapcsolatok
- **CLOUD** – Szolgáltatások a felhőből
- **Mobilitás** – Mobil vállalat

TÁMOGATÓINK

ORACLE®

FUJITSU

IBM

Microsoft

COMPUTERWORLD

VRH

EGYÜTTMŰKÖDŐ PARTNEREINK

ISH

KFKI

IQSYS

Systems

A nagy adat-készülék

San Franciscóban megrendezett konferenciáján az Oracle az integrált hardverből és szoftverből felépülő célrendszerek előnyeit igyekezett a résztvevők fejébe sulykolni, akik továbbra is elsősorban a gyártó adatbázisait, alkalmazásait és köztes szoftverét használják. Nagy appliance-bejelentések és kódos cloud-stratégia a múlt héten lezajlott OpenWorld 2011-en. Írta: Kis Endre

Mielőtt bejelentette volna az Exalytics Intelligence Machine készüléket, Larry Ellison, az Oracle vezérigazgatója az OpenWorld 2011 nyitó előadásában bő egy órát szentelt az appliance-stratégia ismertetésének, amely korábban két célrendszert – az Exadata adatbázis-készüléket és az Exalogic alkalmazásszervert – hozott piacra. Az Oracle eddig mintegy ezer Exadata gépet értékesített, és Ellison szerint idén további három ezer talál majd gazdára. A kedvező piaci fogadtatás annak köszönhető, hogy ezek a hardver- és szoftverelemeket integráló, párhuzamos architektúrára épülő rendszerek egy-egy alkalmazási területen nagyobb teljesítményt és megbízhatóságot adnak alacsonyabb áron, mint a megszokott módon kialakított rendszerek.

zélőfelülete pedig PC mellett az Apple iPadjén is elérhető.

Az Exalytics Intelligence Machine az Oracle válasza az SAP HANA in-memory platformjára, amelyet azonban a felhasználók több hardvergyártótól is beszerezhetnek célrendszerként. Az új appliance ára és a szállítás kezdési időpontja egyelőre nem ismert. Az Oracle viszont bejelentette, hogy E-Business Suite és PeopleSoft üzleti alkalmazásait is optimalizálta Exadata és Exalogic készülékekkel való használathoz.

PÓKERARC NOSQL-HEZ

A konferencia második napján az Oracle újabb célrendszert mutatott be. A Big Data Appliance az olyan masszív információmennyiségeken végzett lekérdezéseket hivatott felgyorsítani, amilyeneket például a weboldalak, az intelligens mérőórák és a szenzorok generálnak.

Az új készülék a nyílt forráskódú Apache Hadoop keretrendszer és az ugyancsak open-source R statisztikai elemző szoftver disztribúciója mellett az Oracle Data Integrator Application Adapter for Hadoop és Loader for Hadoop eszközeit, valamint a cég NoSQL adatbázisát tartalmazza. Segítségükkel a nagy, strukturálatlan adattömegből gyorsan kinyerhetők az üzleti szempontból értékes információk, és további elemzés céljából Exadata készülékre tölthetők. A Big Data Appliance bejelentése 180 fokok fordulatot hozott az Oracle eddigi stratégiájában. A cég idén májusban ugyanis még hosszú tanulmányban (*Debunking the NoSQL*

Hype) fejtegette, hogy a NoSQL adatbázis-technológiák nem elég kiforrottak nagyvállalati felhasználáshoz, kockázatot jelentenek az adatbiztonságra nézve, ezért a szervezetek okosabban teszik, ha a bevált (értsd: relációs) adatbázisok mellett maradnak.

Az Oracle stratégiájának gyors és markáns irányváltása egyértelműen a NoSQL növekvő piaci potenciáljára utal.

A NoSQL adatbázis-technológiák eltérő felépítésüknek köszönhetően éppen az olyan hagyományos, relációs adatbázis-kezelők méretezhetőségi korlátain lépnek túl, amelyet az Oracle is szállít. Nem felfelé, hanem horizontálisan méretezhető, így könnyebbé teszik a nagyon nagy, strukturált és nem strukturált adattömegek elemzését, és jobban illeszkednek a felhő környezetekhez.

Mindebből korántsem következik, hogy a relációs adatbázisoknak hamarosan bealkonyulna. Az Oracle stratégiájának gyors és markáns irányváltása azonban egyértelműen a NoSQL növekvő piaci potenciáljára utal. Elemzői várakozások szerint a Big Data Appliance bejelentése hamarosan az Oracle olyan versenytársait is, mint az EMC, az IBM, a Microsoft, az SAP és a Teradata arra fogja sarkallni, hogy erősítsenek a NoSQL adatbázis-technológiák terén.

BÁRÁNYFELHŐ

A harmadik napon az Oracle cloud computing stratégiájáról is áttekin-

tést adott, így kiderült, hogy SaaS (software as a service), PaaS (platform as a service) és IaaS (infrastructure as a service) felhő szolgáltatások bevezetését a következő pár évben tervezi. Az ütemezést az indokolja, hogy a cég az Oracle Database, Fusion Middleware, Enterprise Manager és VM alapjain a kategóriájukban legjobb technológiákat és szolgáltatásokat kívánja elérhetővé tenni magán- és nyilvános felhőkörnyezetekben. Az OAUG, az Oracle alkalmazások felhasználói csoportja által készített friss felmérés szerint a magán-felhőkörnyezetet építő vállalatok száma az elmúlt egy évben 28 százalékkal nőtt az Oracle ügyfélkörében, míg a nyilvános felhőalapú szolgáltatások használata ennél is nagyobb mértékű, több mint 50 százalékos növekedést mutatott. Így érthető, hogy a konferenciát megelőzően többen az említett cloud szolgáltatások bejelentésére számítottak, és az ütemterv felvázolását kevésnek találták. Akadt olyan elemzői vélemény is, amely szerint az OpenWorld számítási felhővel foglalkozó szekciója a kódösztétést szolgálta, annak leplezését, hogy az Oracle cloud stratégiája egyelőre kidolgozatlan. Az OpenWorld keretében megtartott JavaOne fejlesztői konferencián az Oracle többek között egy Apple iPaden és Google Android-alapú Samsung Galaxy táblagépen futó JavaFX kliensalkalmazást demózt, és bepillantást engedett az Avatar projekt keretében zajló munkába, amelynek célja, hogy a HTML5 segítségével ültessen Javát az iOS platformra. Az Apple hivatalosan nem engedélyezi harmadik féltől származó szoftvertechnológiák használatát iOS-alapú eszközein.

Az Oracle emellett azt is bejelentette, hogy a Java ME (Micro Edition) és SE (Standard Edition) változatát össze kívánja vonni, továbbá a Java SE 8 kibocsátását 2013-ra halasztja – az új verziónak eredetileg jövőre kellett volna megjelennie. A konferencia közönsége viszont működés közben láthatta az Oracle Solaris 11-es verzióját, amely a tervek szerint idén novemberől lesz elérhető. 



Larry Ellison nyitó előadása

Az Oracle készülécsaládjának legújabb tagja, az Exalytics Intelligence Machine a cég TenTimes memóriában futó (in-memory) adatbázisa, BI-alkalmazásai és az Essbase OLAP szerver mellett 40 processzormagot és 1 terabájt memóriát tartalmaz. A célrendszer Ellison szerint a gondolat sebességével futtatja le a lekérdezéseket, az alkalmazott tömörítés révén 5-10 terabájt adatmennyiségen is, a gyakori lekérdezések eredményét gyorsítótárazza, ke-

ribúciója mellett az Oracle Data Integrator Application Adapter for Hadoop és Loader for Hadoop eszközeit, valamint a cég NoSQL adatbázisát tartalmazza. Segítségükkel a nagy, strukturálatlan adattömegből gyorsan kinyerhetők az üzleti szempontból értékes információk, és további elemzés céljából Exadata készülékre tölthetők. A Big Data Appliance bejelentése 180 fokok fordulatot hozott az Oracle eddigi stratégiájában. A cég idén májusban ugyanis még hosszú tanulmányban (*Debunking the NoSQL*

Szoftvertesztelésről felsőfokon

A szoftvertesztelés legelismertebb szakembereinek részvételével rendezti meg a Computerworld és a Hungarian Testing Board kétnapos konferenciáját és workshopját október 13-14-én a Budapesten, a Hotel Heliában.

A magyar alkalmazástesztelési szakma ez évi legnagyobb szabású eseményére kerül sor október 13-14-én Budapesten. Lapunk és a szoftvertesztelők magyarországi szervezete, a Hungarian Testing Board (HTB) régiós fókuszú rendezvényt szervez, ugyanis a Hungarian Software Testing Forumra nemcsak a hazai szakembereket várja. A hallgatóság, ahogy az előadói gárda is, nemzetközi lesz. A környező országok társzervezeteinek tagjai, valamint a romániai, szlovákiai, csehországi és ausztriai szakemberek is szép számmal jelezték részvételi szándékukat.


Az érdeklődés nem véletlen: a konferencia hallgatói a szoftvertesztelés legnevesebb nemzetközi és magyar szakembere-

itől hallhatnak előadásokat, valamint a rendezvény második napján két workshopon vehetnek részt. A rendezvény egyik fő előadója *Rex Black* lesz, aki talán a legnagyobb név ma a szoftvertesztelési szakmában. A szakember, aki a közelmúltig az International Software Testing Qualifications Board (ISTQB), valamint az American Software Testing Qualifications Board (ASTQB) elnöki pozícióját is betöltötte, közel negyedszázados szoftver- és rendszermérnöki tapasztalattal rendelkezik. A program-, hardver- és rendszertesztelés mértékadó alakja, akinek szakírói munkássága is figyelemre méltó. Előadásában a menedzselt szoftvertesztelés vállalati bevezetésének előnyeiről és szükségességéről beszél. A má-

sodik napi workshop témája a rizikóalapú tesztelés eredményes alkalmazása a gyakorlatban.

Hasonló minőségűvel bír a rendezvény keynote előadója is, az angol *Lloyd Roden*, aki a '80-as évek óta dolgozik a szoftvertesztelési szakmában. Eddigi pályafutása jelentős részét az alkalmazástesztelésnek szentelte. Rangos szakmai konferenciáknak rendszeres előadója (EuroSTAR, AsiaSTAR, STAREast, STARWest, SQE Automation, Test Congress, Unicom konferencia). 2004-ben elnyerte az EuroSTAR Software Testing Excellence díjat Kölnben. Első napon előadásában a szoftvertesztelés jelentőségével és kihívásaival foglalkozik.

A két neves külföldi előadó mellett a legjobb hazai szoftver-

tesztelési szakemberek tartanak előadást. *Dr. Forgács István* (4DSOFT) az agilis tesztelés problémáiról beszél, *Tanács Lajos* (Alvicom) előadásának témája ugyancsak az agilis tesztelés, aminek egy banki területen történt gyakorlati megvalósítását ismerteti. *Csöndes Tibor* (Ericsson) a tesztautomatizálás lehetőségeit mutatja be, míg *Bozóki Ferenc* (Ericsson) a Titan/TitanSim keretrendszer működésére hoz gyakorlati példákat. *Száraz Tibor* (Mortoff) a tesztmenedzsment módszertanát és támogató eszközeit ismerteti. 

A konferenciára a <http://computerworld.hu/konferencia/55> oldalon lehet regisztrálni.



PCWorld

Október 6-tól keresse az újságárusoknál!

Ajándék

- Obit Advanced SystemCare PRO 4
- Xara Xtreme 5

A tartalomról

- **WINDOWS 8:** Csendes forradalom
- **FÓKUSZBAN:** kéttömbös NAS megoldások – dupla élvezet!
- **MEGFIZETHETŐ LUXUS:** középkategóriás okostelefonok körképe
- **BETÁBLÁZVA:** Motorola Xoom vs. Samsung Galaxy Tab 10.1



A forráskód közötti különbség

A költséghatékonyság és a gyors reakcióidő kényszerét sok esetben a szoftver tesztelésére fordított erőforrások sínylik meg. Pedig a hibák kijavításának költsége a szoftver kiadása után akár többszöröse is növekedhet, nem is beszélve a felhasználónál felszínre kerülő sérülékenységek és hibák miatti presztízaveszteségekről. Hogyan békíthetjük ki a költséghatékonyságot a minőséggel?

Írta: Forgács István, 4D Soft Kft. és Hajnal Ákos, MTA SZTAKI

A szoftveriparban is kiélezett verseny folyik.

A versenyképesség megköveteli a gyors, költséghatékony és – nem utolsósorban – jó minőségű programok fejlesztését. E kritériumok azonban ellentmondanak egymásnak: a rövid határidők nagyobb fejlesztői csapatot, korszerű fejlesztőeszközöket feltételeznek, míg a fejlesztésre, tesztelésre fordított források korlátozása nem kedvez a szoftver minőségének. A tesztelés a szoftverfejlesztés legköltségesebb fázisa. A fejlesztőcégek ezért nem ritkán az alapos tesztelés mellőzésével próbálnak pénzt és időt megtakarítani. Sokan elfelejtik azonban, hogy a hibák kijavításának költsége a szoftver életkorával párhuzamosan nő (a használat közben kiderülő hibák javításának költsége akár tízszerese is lehet a kódolás során felderített hibákénak), a karbantartás költsége a fejlesztési költségek duplája is lehet, továbbá **az eladott szoftverek minősége, megbízhatósága hosszú távon a fejlesztőcég iránti bizalmat befolyásolja.**

STATIKUS ANALIZÁTOROK

A szoftver minőségének javítása azonban nem feltétlenül

a tesztelésre fordított források növelésével érhető el, hatékonysága is növelhető megfelelő módszerek, a tesztelési folyamat részadatait automatizáló eszközök használatával. Ilyen eszközök az úgynevezett statikus analizátorok (static analyzer), amelyek túlnyomó többsége úgynevezett *forráskód-analizátor*. Ez a program forráskódjának elemzése révén futási időben jelentkező hibákat, metrikákat, kódrészletek és programutasítások közötti összefüggéseket, ún. hatásokat képes felderíteni.

Az automatikusan kinyert információk számos helyen használhatók. A legkorábbi használat a kód minőségének vizsgálata, a helyes kódolási stílus biztosítása. Itt nem konkrét hibákat talál az analizátor, hanem olyan programkonstrukciókat, amelyek használata könnyen eredményezhet hibákat. Ma már csak a C nyelvben kb. 700 ilyen konstrukciót képes felismerni egy jó statikus analizátor.

Lintnek hívták azt a programot, amely először volt képes gyanús, nem portolható (és valószínűleg hibát okozó) programkonstrukciókat megjelölni C nyelvű forráskódokban.

Ma már ezt a kifejezést használják minden olyan eszközre, bármilyen programozási nyelvről is van szó, amely ilyen gyanús kódrészletekre mutat rá. A Lint szoftver 1979-ben jelent meg, vagyis a statikus analizátorok használata több mint 30 éves múltra tekint vissza.

Ezek a statikus analizátorok, más néven kódellenőrök (code checker), a forráskódon, illetve a kód absztrakt szintaxisfáján (AST) alapultak. Mivel az AST előállítása már évtizedekkel ezelőtt is kiforrott technológián alapult, ezek az eszközök viszonylag hamar elterjedtek olyan cégeknél, ahol az egységességet és a kódolási biztonságot fontosnak tartották.

A hetvenes években szintén kifejlesztettek egy új technológiát, amelyet *adatfolyam-analízisnek* nevezünk. Ennek ellenére a technológia csak az utolsó néhány évben fejlődött olyan szintre, hogy a gyakorlatban is használni lehessen.

Manapság az adatfolyam-analízisen alapuló statikus analizátorok száma egyre nő. Mivel ezek készítéséhez kivételes szaktudás szükséges, nem meglepő tehát, hogy az adatfolyam alapú eszközök szinte kivétel nélkül fizetősek, míg a többiek

jórészt ingyenesek. Jelen pillanatban a piacon értékesített statikus analizátorok száma több mint hatvan.

Összefoglalva, az adatfolyam-analizátoroknak fontos szerepük van a szoftver életciklusának minden szakaszában. A programfejlesztés során használhatjuk a szoftverek megértésének megkönnyítésében (program comprehension), a hibák helyének megtalálásában (debugging), valamint a hibalehetőséget tartalmazó „veszélyes” kódrészletek megtalálásában. A tesztelés területén az említett futási idejű hibákat, a biztonsági rések jelentős részét célszerű statikus analízissel megkeresni. A szoftver karbantartása idején még nagyobb a jelentősége a program megértésének és a hibahelyek megtalálásának. Ezen kívül számos tanulmány jelent meg arról, hogyan használható az adatfolyam-analízis a szoftver-újratervezés és -visszafejtés (re-/reverse engineering), program-integráció és -migráció területén is.

FELHASZNÁLÁSI LEHETŐSÉGEK

Nézzük kissé részletesebben a lehetséges felhasználásokat. Az egyik leggyakoribb felhasz-

nálás a futási idejű hibák megtalálása. Ilyen például a memóriafolyás, a null pointer hibák, a nem elérhető kódrészek, az inicializálatlan változók stb. Ezeket a hibákat a hagyományos dinamikus analízis, vagyis a tesztelés segítségével nagyon nehéz megtalálni. Ennek oka, hogy a hiba csak egy speciális végrehajtási utat bejárva lép fel, amely tesztelésére kicsi az esély.

Növekvő felhasználási terület a biztonsági rések automatikus felderítése. Mivel napjainkban egyre erősödnek a hackerek okozta támadások, a piacvezető cégek figyelme mindinkább a biztonsági rések betömésére irányul. A felmérések szerint a statikus analizátorok a biztonsági problémák 70 százalékát megoldják. Az adatfolyam-analízis felhasználható a hibák helyének megtalálásában, valamint a kód megértésében. Ezekben az esetekben a kódon belüli hatásokat kell ismernünk.

HOGYAN MŰKÖDIK?

Minél pontosabbak az analizátorok, annál tovább tart az

héz kiválogatni. Ha csökkentjük a vaklármát, akkor az eredmény nem lesz biztonságos, vagyis tényleges hibákat nem talál meg az analizátorunk. Ebből látható, hogy a statikus analízis igen nehéz feladat, de napjainkban már elértünk egy olyan minőségi szintre, hogy érdemes a fejlesztés során analizátort használni.

Adatfolyam-analízis szempontjából egy számítógépes program változókra történő értékadások és hivatkozások halmaza. Az értékadások – vagy az adatfolyam-analízisben használt terminológia szerint: definíciók (definition) mindazon utasítások, amelyek valamely programváltozónak potenciálisan új értéket adnak (például inicializálás, olyan műveletek, amelyek operandusaik értékét megváltoztatják végrehajtásuk során). A változó hivatkozásokat tartalmazó utasításokat pedig felhasználásoknak (use) nevezük (például kiírató utasítások, feltételes utasítások predikátumai, számítási műveletek).

Egy utasítás lehet definíció és felhasználás is egyúttal, az adat-

nak az érintett változók felhasználásait tartalmazó utasításokra (definíciófelhasználás párok), és fordítva, a felhasználásokra hatással vannak a változók poten-

A statikus analízis a tesztelést egészíti ki: a hibákat könnyebben és gyorsabban javíthatjuk ki.

ciális definíciói. Egy értékadás hatása megszűnik, ha ugyanarra a változóra egy újabb értékadás történik (felüldefiníció). A hatások pontos megállapítását a vezérlési struktúrát reprezentáló ún. vezérlési gráfon (control flow graph) végzett útkeresések eredményeként kapjuk.

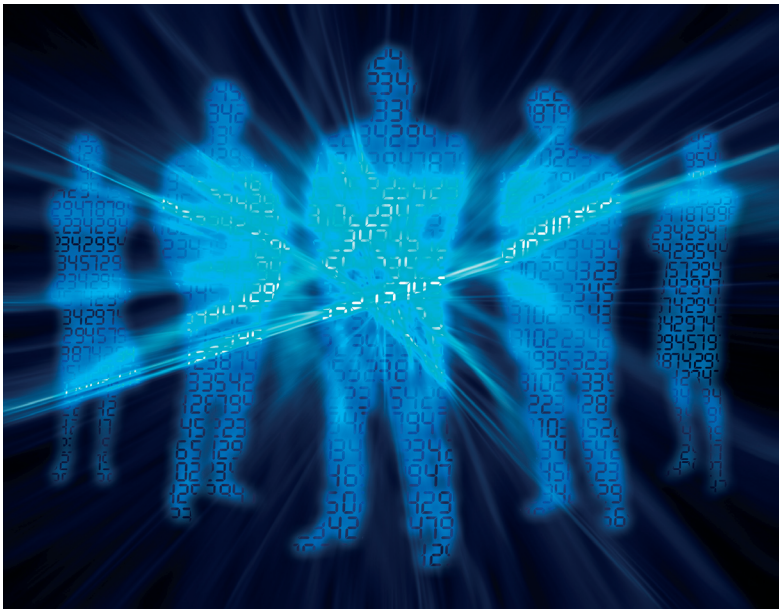
ÉS MIRE JÓ?

Vajon mire jó egy adatfolyam-analizátor? Képzeld el, hogy a tesztelés során programhibát tapasztalunk, ismerjük a hibás eredményre vezető inputot és a hibás eredményt tartalmazó utasítást. A tényleges hiba, azonban a futás során végrehajtott utasítások bármelyikében lehet. A klasszikus debugging során a programot a hibás inputra újra lefuttatjuk, és a lépésenkénti végrehajtás során újra ellenőrizzük az érintett műveleteket – esetleg a programozó intuíciója szerint irrelevánsnak ítélt bizonyos utasításokat átugrunk. Ha az intuíció tévesnek bizonyult, és átléptünk a hibát tartalmazó kódrészleten, a program újrafuttatása szükséges. Most tegyük fel, hogy rendelkezünk egy adatfolyam-analizátorral, amely képes a hibás eredményt adó utasításra közvetlenül vagy közvetetten ható utasítások meghatározására, azaz a program egy szeletének kiszámítására! Debugging során ekkor kizárólag a hibás kimenetre ható utasításokra fókuszálhatunk,

figyelman kívül hagyva a kimenet szempontjából lényegtelen utasításokat. Ha azt vesszük, hogy például az eredeti több száz soros végrehajtási történet (execution trace) így akár tíz releváns utasításra szűkülhet, beláthatjuk, hogy a hatások meghatározása lényegesen felgyorsítaná a hibakeresést.

Egy másik példa. Alapos tesztelés után a szoftvert kiadták, és egy bizonyos idő elteltével a program működésében mégis hibát fedeznek fel. Elkészül a hibajelentés, és a fejlesztő cég nekiáll a hiba kijavításának. Szerencsés esetben a kód érintett részeinek programozói még elérhetőek, rendelkezésre állnak, és emlékeznek a korábban alkalmazott megoldásokra. Kevésbé szerencsés esetben csak a program dokumentációja áll rendelkezésre, és egy másik programozónak kell a forráskódot megérteni, a hibát lokalizálni és javítani. Amíg a fejlesztés során detektált hibák esetén a programozók még pontosan tisztában vannak a program logikai struktúrájával, a karbantartás során kiderülő hibák esetén ennek visszafejtése nem egyszerű feladat. Még rosszabb a helyzet, ha a program számos javításon esett át, a kód kezd eltávolodni az eredeti tervektől, és a dokumentáció sem követte a módosításokat.

Az adatfolyam-analízis használata itt is segíthet. Az egyes változókra ható utasítások automatikus meghatározása révén könnyebben ismerhetjük fel az eredeti gondolatmenetet, az implementált részfunkciókat. A szeletek kizárólag az egymással kapcsolatban lévő, és a hiba szempontjából releváns programfunkciókhoz tartozó utasításokat mutatják. A program megértésén túl, ahogy az a korábbi példában is szerepel, az adatfolyam-analízis segít a hiba lokalizálásában. A javítás után nehéz kérdés annak eldöntése, hogy a program mely részeit szükséges újratestelni (regression testing). Amíg



analízis. A statikus analízis eredménye sohasem teljesen pontos, vagyis mindig kapunk vaklármát (false positive). Az analizátorok egy része olyan sok nem valódi hibát talál, hogy a valódi hibákat igen ne-

folyam-analízis szempontjából a konkrét művelet azonban lényegtelen (összeadás-e vagy szorzás), pusztán az utasításban hivatkozott változók egymásra hatását vesszük figyelembe. Az értékadások hatással van-

eddig a hátramenő szeleteket (mi van hatással erre?) használtuk, ezúttal az előremenő szeletek (mire van hatással?) lehetnek segítségünkre – ezek megmutatják, mely érintett modulokat, funkciókat kell újra tesztelnünk.

A TESZTELŐ SZEREPE

Most vizsgáljuk meg a statikus analízis és a tesztelés kapcsolatát! **A statikus analízis nem teszi szükségtelemmé a tesztelést, hiszen a statikus analízis során az egyes bemenetek hatására előállt output nem ismert.** A statikus analízis bizonyos hibák megtalálására jó – viszont ezekre sokkal hatékonyabb, mint a tesztelés.

Hogyan lehetséges ez? Nem tesztelhetünk minden lehetséges végrehajtási utat a programban, mert ezek száma elképzelhetetlenül nagy. A gyakorlatban annak is örülünk, ha majdnem

Vagyis egy pontos statikus analízátor elvileg minden null pointer hibát megtalál.

Tehát a gyakorlatban a tesztelés és a statikus analízis egymást kiegészítve használandó. Mindkettőnek megvan a maga szerepe és mindkettőre szükség van. A statikus analízis látzólag csak gépidőt igényel, a gyakorlatban sajnos az eredményeket meg kell vizsgálni és kiszűrni a téves riasztásokat. Ennek ellenére a jelentős cégeknél a statikus analízis használatát kötelezően előírják, mert lényegesen javítja a szoftver minőségét.

NEHÉZSÉGEK

Ha az adatfolyam-analízis ilyen hasznos, miért nincs még nekem ilyen eszközöm?! Egy pontos, gyors, azaz gyakorlatban is használható adatfolyam-analízátor megvalósítása nem egyszerű feladat, akármilyen prog-

valósításának nehézségéhez hasonlítható. Ezután a program további reprezentációit (vezérlési és hívási gráfok) is elő kell állítania a hatások további vizsgálata céljából.

Maga az analízis számos gyakorlati nehézséget rejt magában. Amíg tesztelni csak futtatható kódot lehet, addig a statikus analízátor használnak jogos igénye, hogy nem teljesen kész, sőt szintaktikusan hibás kódot is lehessen analízálni. A pontos eredményhez nagyon sok mindent figyelembe kell vennünk. A feldolgozás során csak úgynevezett realizálható programutakat járhatunk be, amelyek valamilyen futás során ténylegesen megvalósulhatnak.

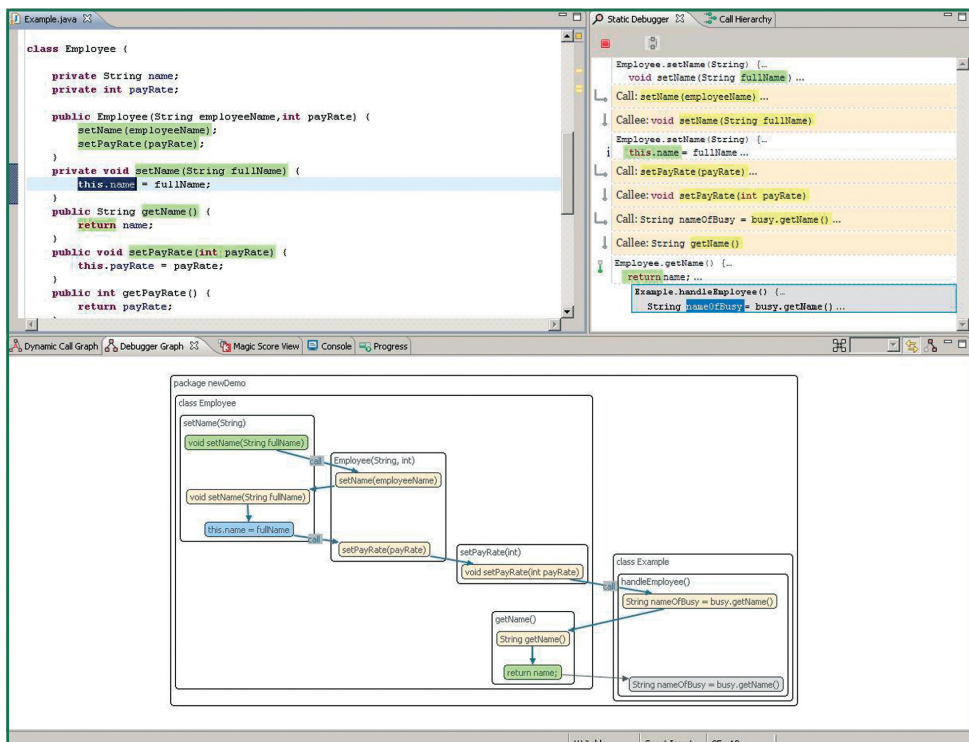
A statikus analízis egy másik alapvető problémája az ún. futásidejű kötések meghatározása (pointerek, függvénypointerek, objektumhivatkozások, polimorfizmus). Például, hogyan állapíthatjuk meg, hogy egy pointer mely változóra mutat, tetszőleges programfutást tekintve,

hiszen az algoritmus nem lineáris, mint a fordítás esetében. További jelentős problémákat okoz a kivételkezelés, tömbök, könyvtárak (például Java collections framework), natív kód, interfészek, absztrakt osztályok, generikus nyelvi elemek, többszálú kód stb. feldolgozása. Az említett problémákkal magyarázhatjuk, hogy miért is olyan drágák a piacvezető statikus analízátorok, noha ezek is távol állnak a tökéletességtől.

Korábban említettük, hogy a statikus analízátorok egyik felhasználási területe a programok megértése. A fejlesztői keretrendszerek támogatják a megértést, azonban a keresőmotorok egyszerű szöveg-, illetve AST-alapú megoldásokon alapulnak. Ezért ezek az eszközök hatásokat, illetve futási idejű hívási gráfokat nem képesek megadni. Ugyanakkor ezen eszközök hatalmas előnye az azonnali válaszidő (just in time, JIT). **Kérdés: lehetséges-e egyáltalán azonnali válaszidejű statikus analízátort készíteni, amely nagy programrendszerek esetén is azonnali eredményt szolgáltat?**

A válasz igen. Ma már létezik azonnali hatásanalízis elvégzésére képes program #, valamint Java nyelvi környezetben is. Egy változót kiválasztva néhány kattintással megtudhatjuk a kiindulási értékét, illetve, hogy a változó milyen hatást gyakorol a teljes szoftvert tekintve. A 4D Soft készülő Java analízátora a futási idejű hívásokat is képes figyelembe venni, így a vaklármák száma minimálisra csökken.

Összefoglalva, manapság a statikus analízis a minőségbiztosítás nagyon fontos eleme, amelyet komoly fejlesztőcég nem hagyhat figyelmen kívül.



Az ábrán a 4D Soft Kft. fejlesztés alatt álló forráskód analízátorának egy képernyője látható. A jobb oldali ablakban egy hatáslánc, a bal oldali ablakban a forráskód látható, megjelölve azok az utasítások, amelyekre a jobb oldali első utasítás hat. Az alsó ablakban a hatások grafikusan jelennek meg

minden programágot letesztelünk. Az adatfolyam-analízisen alapuló statikus analízis azonban olyan eredményt állít elő, mintha minden lehetséges útvonalon analizáltuk volna a kódot.

ramozási nyelvről is beszélünk. Az adatfolyam-analízátornak először is a kód nyelvi elemeinek értelmezését kell elvégeznie, amelynek komplexitása egy fordítóprogram (compiler) meg-

rete. Egy komolyabb szoftver mérete több százezer, sőt millió sor lehet. Ennek még a lefordítása is időigényes, ugyanakkor a pontos adatfolyam-analízis lényegesen tovább

A szoftvertesztelés oktatása Magyarországon

A szoftvertesztelés a minőségi szoftvertermékek előállításának kulcsa. De vajon hol és hogyan képzik azokat a szakembereket, akik a megfelelő kreativitással és szakértelemmel tölthetik be a ma még gyakran üresen álló pozíciókat? Ennek járt utána vendégszerzőnk, Beszédes Árpád egyetemi adjunktus.



Beszédes Árpád
egyetemi adjunktus

Softvertesztelői szakmai körökben általános vélemény, hogy a szoftvertesztelő mint szakma nem kellőképpen megbecsült diszciplína. Legalábbis, az infokommunikációs technológiák egyéb területeihez viszonyítva. Vegyük például azt, hogy számos olyan vállalattal találkozhatunk, amelyben a *szoftvertesztelő* munkahelyi pozíció alacsonyabb rangú (mind erkölcsi, mind anyagi megbecsülésben), mint például a *szoftverfejlesztő*, hasonló mértékű tapasztalatot alapul véve. „Ha nem vált be mint fejlesztő, jó lesz majd tesztelőnek” – vallják sokan. Nyugodtan állíthatjuk továbbá, hogy **jelenleg lényegében fejlesztőket képzünk a felsőoktatási formákban csakúgy, mint a különböző piaci képzési programok keretén belül, és nagyságrendekkel kevesebb figyelmet fordítanak a tesztelés oktatására.**

Jó tesztelőnek lenni azonban komplex, kihívásokkal teli, ugyanakkor megtisztelő és sok szakmai örömet magában hordozó feladat. Tesztelni nem csak annyit jelent, hogy a kérdéses szoftverrendszert sok ezerféle módon „próbálgatjuk” valamilyen soha véget nem érő specifikáció szerint, mindezt rendkívül unalmas és morálfomlító módon. A szakmában dolgozók nagyon jól tudják, hogy számos tesztelési fázis és feladatkör kreatív munkát rejt magában. További tévhit, hogy a tesztelő egyfajta destruktív munkát végez, és a fejlesztőknek mint „alkotóknak” az ellensége. Ez ellen megfelelő kommunikációval lehet védekezni, de számos egyéb tulajdonsággal

is rendelkeznie kell egy jó tesztelőnek; ilyen például az odafigyelés a részletekre, a jó intuíció és természetesen a komplex szakismeret. Mindezekhez először is rátermettség kell, de éppen ennyire fontos a megfelelő képzés biztosítása is.

Az, hogy egy adott szoftverfejlesztési vagy karbantartási projektben megfelelő tesztelési módszerek legyenek alkalmazva megfelelő tesztelő szakemberek által, mindenkinek érdeke. A legtöbb projekt esetében a szoftveréletciklus teljes idejének, illetve költségének jelentős része – akár fele vagy még több is, a kritikusságtól függően – tesztelésre, szoftverminőség-biztosításra fordítódik. De nincs ez az arány kissé ellentmondásban azal, amennyit áldozunk a fejlesztési kontra tesztelési kapacitások létrehozására, fenntartására? Jelenleg azokban a felsőoktatási programokban, amelyek valamilyen módon szoftverfejlesztéshez kötődnek, nem ritka, hogy mindössze néhány órányit foglalkoznak a teszteléssel, szemben a több szemeszteren és különböző tárgyakon átívelő programozó-, programtervező-, fejlesztőképzéssel.

Hogy az ipar a fentiekről hogyan vélekedik? Sokszor a felsőoktatási rendszert és a képzési programokat hibáztatja, tőlük várja el mindazt, amire szüksége van. Vagyis, naprakész tudást és tapasztalatot szinte a munkába állás első napján, és mindezekről valamiféle bizonyítékot is. Mindezt nem könnyű biztosítani, de nézzük meg, mit tehetünk az ügy érdekében mi, a felsőoktatás és az ipar közösen!

A JELENLEGI HELYZET

Szoftvertesztelő szakembereink jelenlegi képzettsége vegyes. Vannak magasan képzett, gyakorlatlaltal és tapasztalattal rendelkező „guruk”, sok a lelkes, de nem különösebben képzett kezdő, illetve fáradságos munkával és költségesen képzett, cégek által kinevelt, egyes szakterületekhez jól értő tesztelők. A képzési színtér jelenleg lényegében a felsőoktatást és a szabványos piaci képesítési rendszereket jelenti (ezek közül legfontosabb az ISTQB-konform képzési forma, *ld. keretes írás*). Jelenleg elvétve beszélhetünk az egyéb szintű képzési lehetőségekről, vagyis különleges középfokú képzésről vagy OKJ-s programokról.

Szinte minden hazai felsőoktatási intézményben, amely valamilyen formában szoftverfejlesztési programmal rendelkezik, találkozhatunk általános szoftverfejlesztési (*software engineering*) tárgyakkal. Ezek keretén belül jellemzően megemlítik a szoftverminőség-biztosítást és a tesztelést. Kissé differenciáltabb képzési móddal találkozhatunk néhány nagyobb egyetemünkön, de ki is emelhetünk néhány intézményt, amely már elindította szoftvertesztelés-képzéseinek jelentősebb átalakítását. A Budapesti Műszaki és Gazdaságtudományi Egyetemen több elméleti és gyakorlati tárgy is foglalkozik a teszteléssel, például a *Szoftvertesztelés, a Tesztelés és minőség laboratórium* a mesterképzésben, valamint a *Szoftververifikáció és -validáció* című, doktori képzésben adott

tárgy. Az Eötvös Loránd Tudományegyetem is több tárgyat tart fenn a tesztelés oktatásának: a *Szoftverfejlesztés minőségi aspektusai* és a *Tesztautomatizálást támogató eszközök a gyakorlatban*. Mindkettő mesterképzési tárgy, míg a doktori képzésben a *Szoftvertesztelés* áll rendelkezésre. A Széchenyi István Egyetemen az *Információs Rendszerek fejlesztése*, valamint a *Szoftverminőség-biztosítás* tárgyak foglalkoznak a teszteléssel. Végül, a Szegedi Tudományegyetemen a két speciálkollégium, a *Szoftvertesztelés alapjai* és a *Szoftvertesztelés gyakorlatán* kívül a *Tesztelési módszerek* mesterszakos tárgy foglalkozik behatóbban a tesztelés elméleti és gyakorlati aspektusaival. Közös jellemzőjük a fenti tárgyaknak, hogy tematikáikat az ISTQB ajánlásai szerint állították össze. [További részletek a témával kapcsolatban megtalálhatók az *Informatika a felsőoktatásban 2011* konferencia kiadványában, 1096–1103 oldalak (Debreceni Egyetem, Informatikai Kar, 2011)].

Ami az ipari gyakorlatot illeti, jellemzően külön tanfolyamok keretén belül készítik fel a vállalatok alkalmazottjait a munkavégzésre. Ez sokszor ad hoc jellegű egyedi képzéseket, különböző eszközökről, technológiákról szóló speciális kurzusokat jelent. **A vállalatok nagy része azonban nagyon fontosnak tartja a szabványos, nemzetközi ajánlásokat, hiszen így megbízható, ellenőrizhető tudáshoz juthatnak hozzá dolgozóik. Természetesen itt is az ISTQB emelhető ki követett módszertanként.**

A felsorolt, szervezett képzéseken kívül természetesen más fórumokon is elsajátíthatják, elmélyíthetik a szakmát a tesztlők, például különböző konferenciákon, szakmai vásárokon, illetve egyéb médiumok, például folyóiratok és internetes közösségek segítségével.

HOVA AKARUNK ELJUTNI?

Az elmondottak természetesen fontos elemei a hazai szakemberképzésnek, azonban még sok téren van fejleszteni való, de ehhez azt kell látni, hogy a probléma rendkívül sokrétű. Fontos a felsőoktatási képzés harmonizálása, továbbfejlesztése, de ugyanúgy az ipari képzéseket is tovább kell fejleszteni, szervezettebb mederbe terelni, esetleg egyéb kapcsolódó területekkel együtt, mint például a követelménykezelés vagy projektmenedzsment.

Elsődlegesen a felsőoktatásban lenne szükség jelentősebb fejlesztésekre, ugyanis a fentiek értelmében még eléggé kezdeti fázisban vagyunk, illetve jelenleg még viszonylag nagy a rés a kimeneti tudás és az ipari elvárás között. Természetesen az intézményektől nem várható el, hogy minden hallgatót naprakész tudással lássanak el, de a jelenleginél azért sokkal szervezettebben, összehangoltabban lehetne megtervezni a programokat.

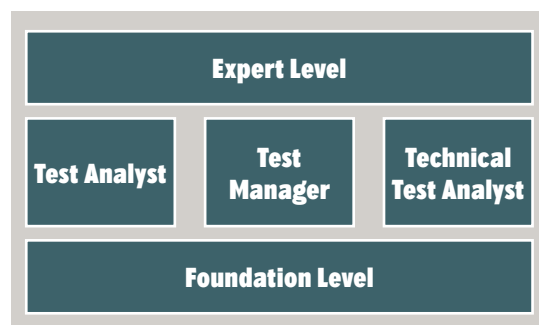
A jelenleg vezető intézmények kínálatát elemezve az alábbi fejlettségi szintek, illetve fázisok tapasztalhatók:

- ▶ **kezdeti fejlettség** – ez 1-2 általános törzstárgy keretén belül kínált néhány órás külön foglalkozást jelent a témával
- ▶ **fejlődő fázis** – 1-2 speciális kurzus, például speciálkollégium meghirdetését jelenti bizonyos szakokon, választható tárgyként. A tárgyak sokszor meghívott előadókkal tartottak
- ▶ **tervezett képzés**, amelyben már tudatosan összeállított képzési program keretében legalább 2-3 tárgy foglalkozik a tesztléssel, közöttük törzstárgy is van
- ▶ **speciális szakirány**
- ▶ **szoftvertesztlő szak** (országosan akkreditált).

Mint látható, jelenleg legjobb esetben is csak a 2. szinten helyez-

kednek el az intézményeink, míg a külön szakirány és szak egyelőre még csak távoli vízió. Azonban már most elkezdhetünk gondolkodni egy ilyen jövőbeli szak struktúrájáról azért, hogy ha a megfelelő feltételek rendelkezésre állnak, elkezdődhetnek az előkészületek. Egy lehetséges kompetencialista (amely a tapasztalat és az ajánlások alapján levezethető) az alábbiakat tartalmazhatná:

- ▶ programozási alapok, nyelvek
- ▶ operációs rendszerek
- ▶ szoftverfejlesztési ismeretek, software engineering
- ▶ követelmények kezelése, requirements engineering
- ▶ szoftverminőség általános megközelítése, ISO, CMMI, egyéb szabványok, software quality
- ▶ folyamatmenedzsment, modellek, IT-menedzsment: COBIT, ITIL, ...



Az ISTQB képzési struktúra

- ▶ **eszközhasználat**: konfigurációnedzsment, eseménykezelés, egyéb eszközök
- ▶ **üzleti aspektusok**, projektmenedzsment, gazdasági ismeretek, költségbecslés
- ▶ **végül**, de nem utolsósorban nemzetközi ajánlások alapján történő tesztlői képzés, legalább ISTQB Foundation és Advanced szintek.

A fentiek kiegészülnek a speciális, tesztléssel foglalkozó doktori képzéssel, valamint a tudományos kutatási programokkal.

Nagyon fontos a gyakorlati tudás megszerzése is, amihez hozzájárulhat az ipari kapcsolatok erősítése, ipari résztvevők bevonása a képzésbe. Ez elképzelhető lenne például kötelező szakmai gyakorlat keretében.

Az előbbieken vázoltak mind szükségesek kompetens szoftvertesztlő szakemberek „előállításá-

hoz”, és látható, hogy ezek teljes megvalósítása több, egymásra épülő kurzus elvégzését jelenti. Kisdolgozásuk sok erőfeszítést és időt igényel, de talán hamarosan külön, fejlesztőkkel egyenértékű szakon képezhetjük tesztlőinket...

RÖVID TÁVÚ REALITÁS, ÉS AMIT TEHETÜNK

De maradva a realitás talaján, először is fontos megteremteni és elmélyíteni a felsőoktatás és az ipar kommunikációját és együttműködését a témában. Az egyetemektől nem várható el, hogy naprakész szakembereket képezzenek, akik minden környezetben kis ráfordítással, szinte azonnal bevetethők. Az egyetemek és főiskolák képzési programjai természetüknél fogva nagy tehetetlenséggel bírnak, amit tudomásul kell venni, és amihez igazodni kell. Ugyanakkor alapvető kötelességük ezen intézményeknek a stabil alap biztosítása, megfelelő általános alapismeretek elmélyítése. A technológiai különlegességek később pótolhatók az adott munkakör el látásához szükséges mértékben.


Ennek egy lehetséges módja az, hogy az iparral konzultálva a felsőoktatási intézmények kidolgozzák az általános képzési programokat, ideális esetben egymással intenzíven együttműködve, így esetleg csereszabatos tematikákat eredményezve. Ezenfelül létre kell hozni a különböző szakterületekre vonatkozó speciális kurzusokat, amelyeket a hallga-

tók az igényeknek megfelelően választhatnának. E speciális képzéseknek viszont nagyon flexibiliseknek kell lenniük, az ipari igényekre való gyors reagálási móddal.

Ezt azonban csak az iparral szoros együttműködésben lehet megvalósítani, ami áldozatokkal jár mindkét fél részéről. Az egyetemi oktatóktól nem várható el az a fajta gyors tanulás, amit a képzések ellátása igényelne, ezért az ipar segítségével megkerülhetetlen. Szerencsére több lehetőség is adódik erre.

Az ipari képviselők támogathatják például a különböző laborok kiépítését, képzési anyagok előállítását, vendégoktatók biztosítását, különböző szakmai gyakorlatok szervezését, szakdolgozattémákat stb.

Rövid távon is több dolgot tehetünk a cél érdekében. Először is, fontos lenne a nemzetközi ajánlások minél elszántabb követése, például az ISTQB rendszer beépítése a felsőoktatási képzésekbe. Másodsorban, az egyes intézmények már most elkezdhetik saját tematikáik összehangolását és szorosabb kapcsolatok kiépítését. Ez jelentheti például az oktatók és diákok cserefoglalkoztatását, vendégelőadók meghívását, közös szakdolgozati és doktori témák meghirdetését, záróvizsga tételek közös kidolgozását stb.

Végül, de nem utolsósorban, rendkívül fontos a szakmai közösségek felélesztése és életben tartása, amely különböző rendezvények szervezésében, internetes fórumok üzemeltetésében és egyéb médiumok, hírlevelek, folyóiratok kiadásában nyilvánulhat meg. Ebben nagy segítség lehet az olyan szervezetek közreműködése, mint a HTB vagy az IDG. 

Az ISTQB Magyarországon

Az International Software Testing Qualifications Board (ISTQB, <http://www.istqb.com>) egy 47 tagot számláló világszervezet, amelynek célja a professzionális szoftvertesztlés képesítési kereteinek koordinálása; magában foglalja a képzés és vizsgáztatás szakmai és szervezési feltételeinek kidolgozását és azok ellenőrzését. 2008-ban alakult meg az anyaszervezet magyarországi tagszervezete, a Magyar Szoftvertesztlői Tanács Egyesület (Hungarian Testing Board, HTB, <http://www.hstqb.com>). Mára már Magyarországon is számos, de világszinten 180 ezer vizsga tanúsítja a rendszer sikerességét. Az ISTQB képzési séma három szintben határozza meg az elvárt ismerethalmazt: Alap (Foundation), Haladó (Advanced) és Szakértő (Expert) – *ld. ábra.*

Tesztelni az első pillanattól

Változóban van a szoftverfejlesztés metodológiája: az egyre gyorsabb reakcióidőt követelő piac hatására egyre több cég vált a különböző agilis fejlesztési módszertanok valamelyikére. De ugyanezek a követelmények a szoftvertesztelés számára is kihívást jelentenek. Cikkünkben egy olyan új módszertant mutatunk be, amely a szoftvertesztelést már a kezdetektől inkorporálja a szoftver-előállítás folyamataiba. Társkiadványunk, az infoworld.com szerzőjének, John Ferguson Smartnak „A better approach to software testing: Do it before you code” című cikke alapján a magyar változatot írta: **Odrovics Szonja**

A módszertan neve *acceptance-test-driven development* (ATDD), azaz magyarul kb. átvételi vizsgálatokon alapuló fejlesztés. Az acceptance tesztek, vagyis az átvételi vizsgálatok koncepciója természetesen nem újdonság; a hagyományos minőségbiztosítási eljárások is azon alapulnak, hogy mielőtt a terméket késznek nyilvánítanánk, tesztekkel ellenőrizzük, vajon megfelel-e bizonyos minőségi követelményeknek. **A módszer újdonsága abban rejlik, hogy ezekre a tesztekre és megtervezésükre nem a tervezési, majd a fejlesztési folyamatok lezárulta után, azoktól szinte függetlenül kerül sor, hanem e folyamatok szerves részeként, a kezdetektől végigkísérve azokat.**

A hagyományos waterfall-modell egyik legnagyobb hátránya, hogy ha egy hibára ilyen kései fázisban derül fény, az nagymértékben megnéhezíti a kijavítását még akkor is, ha a probléma viszonylag könnyen helyrehozható, kisebb kódrészletet érint. Ennek eredményeképpen növekednek a javítás költségei, a fejlesztőcsapat elvesztegetett ideje, és csúszik a leadás határideje.

Az ATDD teljesen más megközelítésből indul ki: ennek megfelelően az átvételi vizsgálatokat közös munka eredményeképpen, még a fejlesztési munkák megkezdése előtt határozzák meg. A tesztelés így már nem a QA (Quality Assurance, minőségbiztosítás) csapat felügye, hanem olyan közös feladat, amely a tesztelőkön kívül nemcsak a fejlesztőket, de a termékmenedzsereket és az üzleti elemzőket is bevonja. Ez lehetővé

teszi, hogy minden résztvevő már a tervezési fázisban is átlássa és megértse a tesztelés problémáinak jelentőségét.

Emellett az átvételi tesztek nem szeparálódnak és nem számúzetnek a fejlesztést követő időszakra, hanem automatikusan és a teljes fejlesztési folyamatba integrálva, azt végigkísérve valósulnak meg. Ennek következtében a hibák gyorsan a felszínre kerülnek, olcsóbban és gyorsabban javíthatók, a tesztelő munkaterhelése pedig egyenletesebben oszlik el, nem koncentrálódik a munkafolyamat végére. Összességében a csapat gyorsabb és jóval hatékonyabb reakciókra képes, ami különösen a kezdeti specifikációk változása esetén jelenthet óriási versenylőnyt.

ATDD A GYAKORLATBAN

Érdemes közelebbről is megnézni, hogyan működhet az ATDD egy agilis fejlesztési projektben. Egy szoftverprojekt alapvető célja, hogy bizonyos számú kiemelt feature-t, a felhasználó felé kommunikálható szolgáltatási értéket kínáljon a felhasználóknak. Ilyen lehet például egy ingatlanlizing-menedzsment applikáció esetében a karbantartás-menedzsment.

A feature-ök általában túl nagyok ahhoz, hogy egyszerre implementáljuk őket, ezért általában kisebb, kezelhetőbb egységekre bontjuk azokat. Agilis körökben ezeket az egységeket *ún. user storyk* formájában írjuk le, megfogalmazzuk egy-két rövid mondatban, hogy egy funkcionális egy bizonyos elemétől mit vár a felhasználó.

A karbantartás-menedzsment feature esetében ilyen lehet például a munkarend felállítása vagy a számla jóváhagyása.

Ezek a user storyk adják azoknak a beszélgetéseknek az alapját, amelyek segítenek tisztázni az igényeket, majd megszülethet belőlük az, amit végül implementálni kell: ezt célok és számonkérhető átvételi kritériumok formájában fogalmazzák meg, például így: „A felhasználó nem hagyhatja jóvá a számlát, ha annak összege túllépi a megállapított maximum árat.”

Az átvételi kritérium meghatározza, hogy egy bizonyos user story mikor áll készen a forgalomba helyezésre, de ez nem csak annyit jelent, hogy rögzíti, mit kell tesztelni az iteráció végén. Az átvételi kritérium közös feladatként jelenik meg már az iteráció kezdetén, a fejlesztők, a tesztelők és a termékmenedzserek bevonásával. Ennek eredményeképpen elősegíti azt is, hogy a csapat minden tagja konkrét és részletes vízióval rendelkezzen a megvalósítandó feature-ről, így csökkentve a hiányos erőforrás-felmérésből adódó konfliktusok és a belőlük eredő kisebb katasztrófák számát.

Fontos, hogy az átvételi kritériumoknak nem kell kimerítőnek lenniük; ez a technikai tesztek feladata. Az átvételi kritériumokat legalább annyira használjuk kommunikációs, mint verifikációs célokra: működő példák formáját öltik magukra, ezért is hivatkozunk gyakran mind az ATDD-re, mind a példákon keresztül való specifikációra.

Persze az ATDD-t nem csak agilis módszertant használó projektek esetében vethetjük be. **Azok a csapatok is profitálhatnak belőle, amelyek formálisabb és részletesebb use case-ekkel vagy olyan tradicionálisabb megközelítésekkel dolgoznak, mint például az SRS (szoftverkövetelmény-specifikáció), hiszen a lehető legkorábban juthatnak általa ellenőrizhető, automatizált átvételi kritériumokhoz.**

AZ ÁTVÉTELI VIZSGÁLATOK AUTOMATIZÁLÁSA

Az átvételi kritériumok kulcseleme, hogy automatizáltan képesek működni. Nem egyszerűen egy Word vagy Excel fájlban tároljuk őket: élő, végrehajtható tesztek. Az ATDD akkor igazán hatékony, ha az automatizált átvételi vizsgálatok lefutnak minden alkalommal, amikor változtatunk a forráskódon. Éppen ezért nagyon fontos, hogy rendelkezésünkre álljon egy eszköz, amely könnyen integrálható a fejlesztési folyamatainkba, és emberi beavatkozás nélkül fut a build servereinken.

Az automatizált átvételi vizsgálatok nem csupán azt a célt szolgálják, hogy teszteljék az applikációt, hanem egyben egy objektív mérőeszköz is adnak a kezünkbe a projekt előrehaladásának felméréséhez (az agilis projekteknél például nem is fogadunk el mást az előrehaladás mércéjeként, csak a működő szoftvert). A vizsgálatok ráadásul képet adhatnak az egyes feature-ök és storyk viszonylagos komplexitásáról is: azokat a funkciókat, amelyeket bonyolult és sokáig tart tesztelni, jó

eséllyel bonyolult és hosszadalmas fejleszteni is. Ez fontos segítség lehet a termékmenedzserek számára a prioritások felmérésében.

Bár természetesen a hagyományos teszteszközök – mint például a TestNG – is alkalmasak arra, hogy automatizált átvételi vizsgálatokat írjunk bennük, van néhány szoftver, amelyet kifejezetten az ATDD módszertan támogatására hoztak létre.

ATDD-ESZKÖZÖK

Az egyik legkorábbi ATDD-teszt-eszköz a *FitNesse*. Ebben a felhasználók táblázatos formában vihetik fel a követelményeket egy wikibe, majd a fejlesztők a kulisszák mögött írják meg a kódot, amely a wikiben tárolt tesztadatokkal lefuttatja a teszteket az applikáción. A tesztek végrehajtása után a táblázat színekkel jelzi, hogy melyek zártultak sikerrel, és melyeken nem ment át az alkalmazás. A *FitNesse* nagyon hasznos eszköz lehet, ha átvételi vizsgálatunk követelményei kifejezhetők adattáblákkal és elvárt eredményekkel, illetve használják még lépéssorozatban kifejezhető átvételi kritériumok esetében is.

Újabban több olyan eszköz is napvilágot látott, amely a BDD (behavior-driven development, azaz viselkedésvezérelt vagy viselkedésalapú szoftverfejlesztés) módszertant támogatja. Ez a technika arra sarkallja a fejlesztőket, hogy az applikációt annak viselkedése felől közelítsék meg, és az alacsony szintű technikai követelményeket is narratívába helyezve fogalmazzák meg. A *Cucumber* a Ruby közösség népszerű eszköze, amely lehetővé teszi, hogy az átvételi követelményeinket az agilis projekteknél gyakran használt *given-when-then* (*adva-amikor-akkor*) struktúrában fogalmazzuk meg. A *Cucumber* Javával használható.

A *JUnit* hasonló megközelítést használ: a sztorikat szövegfájlokban fejezi ki, a tesztek pedig Java osztályokban írhatók meg. Hasonló eszköz még a Groovy nyelven alapuló *Easy* is.

Még frissebb ATDD-eszköz a *Concordion*, amelyben az átvételi vizsgálatok táblázatokat és szö-

vegeket is tartalmazó HTML-oldalak formájában fejeződnek ki. Az eszköz Java osztályokat használ az oldalakon elhelyezett tagek (címkék) analizálására, az eredményeket pedig HTML-ben jelzi ki.

Ezek az eszközök mind nagy hangsúlyt fektetnek az olvashatóságra és a kommunikációra. Az alábbi kód jól illusztrálja, hogyan fejezhetünk ki egy átvételi kritériumot *Easy* segítségével:

```
scenario User can approve an
invoice for an amount less
than the agreed maximum"
{
given „the User has selected
an open invoice”,
and „the User has chosen to
approve the invoice”,
and „the invoice amount is
less than the agreed maximum
amount”,
when „the User completes the
action”,
then „the invoice should be
successfully approved”,
}
```

Amint ily módon meghatároztuk az átvételi kritériumot, a hozzá tartozó teszt kód már megírható a hagyományosabb programnyelvek valamelyikén, például Javában, Groovy-ban vagy Ruby-ban.

Az *Easy* bemutatása mellett ez a kód részlet az is jól megvilágítja, hogy a kommunikációnak kiemelt szerep jut az ATDD-eszközökben. Az automatizált átvételi kritériumokat olyan magas szintű kifejezésekkel fogalmazzák meg, amelyek az üzleti vezetők számára éppoly érthetőek, mint a szoftvermérnökök és a programozók számára. A legtöbb ATDD-eszköz még arra is alkalmas, hogy a teszteredményekből hétköznapi üzleti kifejezésekben megfogalmazott jelentéseket generáljon.

A még nem leködolt, de ily módon már megfogalmazott átvételi kritériumokat a program a „pending” (folyamatban) címkével látja el: egy iteráció kezdetekor

minden teszt állapota ez, a következő lépés pedig a tesztek implementálása. **Igy a riportok nem csupán a teszteredményekről, de a projekt, illetve az egyes részfeladatok állapotáról is tájékoztatnak.**

Kicsit távolabbról nézve az automatizált átvételi vizsgálatok olyanok, mint bármely más automatizált teszt: el kell tárolnunk őket a verziókontrollt biztosító rendszerünkben, és időről időre lefuttatni őket a CI- (continuous-integration, folyamatos integráció) szerverünkön (a gyors visszajelzés érdekében legalább minden éjjel, de lehetőleg minden forráskódváltozáskor). A CI-szerveren érdemes azt is beállítani, hogy az eredményeket olyan helyre is publikálja, ahol a nem fejlesztő munkatársak is könnyen elérhetik. Szerencsére az újabb CI-eszközök, mint például a Jenkins, nagyon könnyen integrálhatók szinte mindegyik BDD-alkalmazással. 

„Hagyományos” vs. közösségi

Nyílt forráskód és közösségi szoftverfejlesztés: e nem ok nélkül trendi fogalmakat gyakran használják egymás szinonimájaként. Pedig nem minden open source fejlesztés célja közösség építése és bevonása a fejlesztésbe, ahogy előfordul az is, hogy felhasználói vagy fejlesztői közösséget építenek egy nem nyílt forrású termék köré. Jelen cikkünkben a közösségekre helyezük a hangsúlyt, és kevésbé tartjuk fontosnak, hogy az adott projekt nyílt vagy zárt forrású-e. Írta: Kiss Attila

A hhoz, hogy megmutassuk a közösségi tesztelés előnyeit és hátrányait, vagyis megtaláljuk a helyét a mindennapi üzleti gyakorlatban, először vegyük górcső alá, hogyan működik a „hagyományos” szoftvertesztelés, amin itt az alkalmazott, hivatásos szoftvertesztelők munkáját értjük!

TESZTELŐK A VÍZESÉS ALJÁN

A tesztelés legalapvetőbb célja, hogy a termékfejlesztés minél korábbi fázisában fényt derítsen a hibákra, hogy azok minél alacsonyabb költségen javíthatók legyenek. Szoftverek, különösen komplex rendszerek esetében a gyakorlatban nem létezik hibamentes állapot, ezért a feladat a minőség mérése és meghatározott szintre emelése. Ez kiegészülhet a felhasználó kockázatainak becslésével és kezelésével. A tesztelés tehát a minőségbiztosítási folyamat része, hiszen egy elvárásnak való megfelelésről van itt szó. A problémát az jelenti már első körben is, hogy kinek milyen elvárásának kell megfelelni.

A szerző a BalaBit marketingvezetője

A fejlesztők természetesen egy specifikációnak való minél teljesebb megfelelést értik ezen. Viszont érthető módon mindez a felhasználókat teljes mértékben hidegen hagyja. A hibásan specifikált fejlesztéseket éppúgy minőségi hiányosságnak érzékelik, mint az azoktól való eltéréseket. Ebből következik, hogy egy szoftver minősége már a specifikációnál jelentősen meghatározott. Ennek ellenére a terméktervezést sokszor nem értik bele a fejlesztés folyamatába. A piacon általában azok a fejlesztővállalatok a sikeresek, amelyek ügyféllel a középpontban közelítenek a minőséghez, és nagy hangsúlyt fektetnek a felhasználói igények feltérképezésére és specifikálására. A következő minőségi értelmezési szint a tulajdonosi értelmezés – utóbbinak profitelvárásai vannak. Leegyszerűsítve, a tulajdonosok számára a termék azon tulajdonságai jelentik a minőséget, amiért a vevő magasabb árat hajlandó fizetni.

A hagyományos szoftvertesztelésen belül is van egy tradicionális vonal, amely kizárólag az elkészült szoftvert vizsgálja. A fejlesztők ilyenkor a kész terméket és annak funkcionális, valamint technikai



specifikációját átadják a tesztelő-csapatnak, amely megalkotja a tesztelési tervet, majd végig is viszi azt. A talált hibákat visszacsatolják a fejlesztőknek, akik nekiállnak kijavítani azokat, majd a folyamat kezdődik elölről. Ez a megközelítés viszonylag kevés tesztelőt igényel, viszont nem kedvez a gyors fejlesztéseknek, főleg dinamikusan változó specifikáció esetén, amit piacra készülő (nem egyedi fejlesztésű) szoftver esetében a turbulens piaci környezet generál.

TESZTELÉS AZ ELSŐ PILLANATTÓL

Ennél újabb, rohamosan terjedő fejlesztési metódus az *agilis* fejlesztési modell, amely a sikeres japán gyártók lean, folyamatalapú megközelítését igyekszik a korábban túlzottan projektszemléletű szoftverfejlesztésbe implementálni. A különbség nagyjából ahhoz hasonlítható, amikor az emberiség a manufaktúris termék-előállításról áttért a gyártásra.

Az agilis modellekben a tesztelők napi szinten együttműködnek a fejlesztőkkel a fejlesztés teljes folyamatában, annak legkorábbi fázisától kezdve. Ez azt jelenti, hogy sokkal több tesztelési időt kell tervezni, hiszen – látszólag feleslegesen – a szoftver minden porcióját rengetegszer fogják tesztelni, miközben tudják, hogy mind a kód, mind a specifikáció még sokat fog változni. Összességében mégis gyorsabb és rugalmasabb fejlesztést kapunk, ahol a problémák még a keletkezésük idején és helyén napvilágra kerülnek és javítják őket. A megközelítés főleg az alapvető, koncepcionális, architektúrais hibák esetében jelent hatalmas előnyt a tradicionális tesztelési folyamatokhoz képest. Az agilis módszer szerinti tesztelés filozófiája hasonló a TQM-éhez vagy a lean menedzsmenthez – ezek azt állítják, hogy a minőséget nem lehet a gyártás végén „beleellenőrizni” a termékbe, hanem a gyártás folyamatát kell úgy alakítani, hogy az minőséget adjon. Az agilis tesztelő napi szinten, kalákában dolgozik a termék egyazon részén a fejlesztővel, tulajdonképpen egyenrangú társaként, egy másik

szemléletet adva a folyamatba, így a problémák vagy hibás elgondolások akár már ötletfázisban kiszűrődhetnek.

Mivel a tesztelők már a termék igen korai, sőt embrionális fázisában is tesztelik az elkészült kódrészeket, nem tudnak olyan egyszerűen megfogalmazott elvek alapján dolgozni, hogy megfelel-e valami a specifikációnak vagy nem. Főleg, hogy a specifikáció az agilis fejlesztésben a fejlesztéssel együtt fokozatosan lesz egyre pontosabb és részletesebb. Ezért a tesztelők különböző szintű és részletességű teszteteket használnak. Lássuk ezeket:

– **Unit teszt.** A készülő szoftveregységek első körös tesztelése abból a célból, hogy a fejlesztésnek biztos, jó minőségű alapot szolgáltatassanak. Mivel a tesztelt egységek (unitok) funkcionálisan nehezen értelmezhetők, ezért a unit tesztesetek definiálásához nagymértékben szükséges a fejlesztők tevékeny hozzájárulása. Az együttműködés talán itt a legszorosabb fejlesztő és tesztelő között.

– **Modul (integrációs teszt).** Ebben a fázisban a szoftverelemek modulokká állnak össze, amelyek már jól értelmezhető funkciókat valósítanak meg. Mivel ekkor még nincs felhasználói felület (GUI), ezért itt szintén a programozási nyelvet jól ismerő tesztelők kerülnek előtérbe, akik saját scriptjeiket és parancsaikat használják. A tesztelésnek ez a fázisa a legjobban automatizálható, hiszen csak egy jól körülhatárolható rendszerünk van bemenetekkel és kimenetekkel. Ezen a szinten kell foglalkozni a modulok együttműködési képességével is, amely az interface-specifikáció ismeretében tesztelhető.

– **Rendszerteszt.** A szoftver tesztelése a normál felhasználói interface-eken keresztül. Ez az a fázis, amikor a specifikációnak való megfelelést első ízben vizsgálják a teljes termékre. Jellemzően jól automatizálható.

– **Elfogadási teszt.** Az a fázis, amikor a megrendelő átveszi a terméket. Itt a tesztesetek valós felhasználást szimulálnak, és azokat a megrendelő bevonásával készítik el. Egyedi fejlesztés esetén

a megrendelő a jövőbeli felhasználó, amíg piacra szánt termék esetén ez a termékmenedzser vagy a marketing.

A KÖZÖSSÉG SZÍNRE LÉP

Ha ezekhez a szoftvertesztelési módszerekhez és szintekhez hasonlítjuk a közösségi tesztelést, akkor a következő alapvető megállapításokat kell tennünk:

Hátrányok

– A közösségi tesztelői nem képesek unit és modulteszteteket végezni, vagyis agilis fejlesztésekbe csak korlátozottan vonhatók be. Még akkor is, ha gyakran adunk ki korai szoftververziókat.

Nem tesztelési tervet szerint haladnak, ezért nem lehetünk biztosak a teljes körű tesztelésben.

Előnyök

– A közösségi tesztelők elvileg ingyen dolgoznak nekünk. A gyakorlatban van némi költsége a közösségi infrastruktúrának, de ha a közösség kellő méretű, akkor ez bőven megtérül.


– Községi tesztelőink akár több tízszer vagy százszor többen lehetnek, mint a belső csapat.

– A közösségi tesztelők általában felhasználók is egyben, így olyan szemmel képesek tesztelni a ter-

méket, ahogy a belső tesztelők sosem lesznek képesek.

Míndezek alapján kimondhatjuk, hogy a közösségi tesztelés nem helyettesítheti a belső tesztelői kapacitást, de jól kiegészítheti azt. Ez a kijelentés azonban nem fejezi ki teljes mértékben a közösség jelentőségét, amely méreténél és összetételénél fogva iszonyatos versenylőnyt biztosíthat a fejlesztőcégek számára. Azt adják ugyanis a vállalatok számára, amiben azok leginkább szegények: információt a felhasználói igényekről, elvárásokról, szokásokról. A közösségi tesztelő szemében a nem megfelelő működés éppolyan hiba, mint egy hiányzó funkció. Az ő tesztelésükből tehát nemcsak a hibákat, de az elvárásokat is meg lehet ismerni.

És miről beszéltünk a cikk legelején? Egy jó specifikáció már fél siker!

A közösség segítségével rajta tarthatjuk ujjunkat a piac ütőerén, és idejekorán azonosíthatjuk azokat a felhasználói igényeket, amelyek a jövőben elvárásokként jelentkeznek majd. Egy jól menedzselte felhasználói közösséggel tulajdonképp beemeljük a felhasználót az értékteremtési láncba, hiszen az ő segítségével fogjuk specifikálni a termékeinket. 



Felhasználókhöz igazodva

A fejlesztők újabb és újabb adaptív felületekkel bővítik az EuroOffice irodai programcsomagot. A jövőben kiemelt szerepet kapnak a netbookok és a mobilkészülékek használatát segítő megoldások.

Több irányban is folyik az OpenOffice forráskódra épülő EuroOffice irodai szoftvercsomag továbbfejlesztése. Az egyik fő irány a jobb használhatóságot segítő bővítmények, ezek közül is kiemelten az adaptív felületek létrehozása. De mit is takar az adaptív felület, illetve mi adta az ötletet, amelynek megvalósításán a Multiráció Kft. az ELTE bevonásával dolgozik?

EMLÉKEZŐ SZOFTVER-KÖNYVTÁRAK

Induljunk ki abból, hogy egy irodai alkalmazásgyűjtemény funkcionalitásának még a legprofesszionálisabb felhasználók is csak a töredékét, legfeljebb az 5 százalékát használják ki. Mindazonáltal még a legritkábban használt funkciókat is el kell érni a menüből, ugyanakkor olyan menüszervezetet kell kialakítani, amiben a kezdő felhasználók is eligazodnak. A feladat nem egyszerű, de a megoldása korántsem lehetetlen. A Multiráció – az ELTE-vel karöltve – olyan változtatásokat hajtott végre a forráskódban, amelyek hatásai mind az alap EuroOffice programban, mind annak bővítményeiben észlelhetők. A hatás lényege, hogy a szoftver valósággal kitalálja a felhasználó gondolatait, ezáltal növeli a hatékonyságát, ráadásul mindenféle tolokodás nélkül.

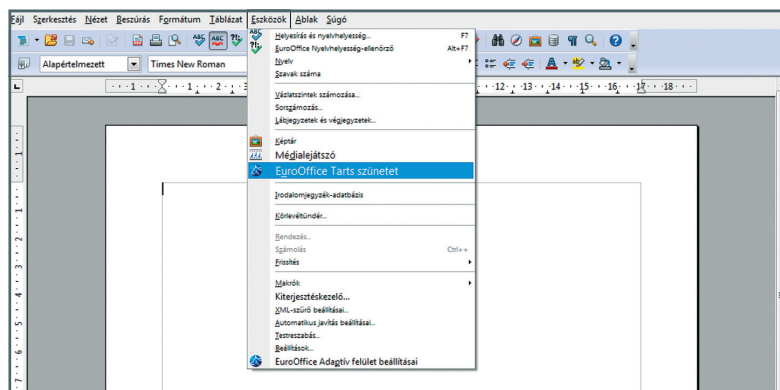
„Megoldásunk alapja, hogy magában az alkalmazásban és a bővítményekben is létrehoztunk egy-egy olyan szoftverkönyvtárat, amely figyeli a felhasználó által használt funkciókat, emlékszik ezekre, majd hozzájuk igazítja a felhasználói

felületet. Az így létrejövő interface-t nevezzük adaptív felületnek” – mutat rá *Banai Miklós*, a Multiráció ügyvezető igazgatója.

NETBOOKHOZ IGAZÍTVA

Az újszerű, adaptív menürendszerek iránti növekvő igény egyik alapvető oka a netbookok terjedése. A netbooknak fizikailag kicsi a kijelzője, azon tehát csak viszonylag kis méretű betűket lehet megjeleníteni. Ez különösen komoly problémát okozhat olyan helyszíneken, ahol gyér a külső megvilágítás, vagy például utazáskor, az eszköz állandó mozgásakor, rázkódásakor. Ilyenkor nincs más választás: nagyobb betűméretet kell választani. Ez azonban a kis képernyőn számos, rendkívül zavaró megjelenítési problémához vezet – itt juthat szerephez az adaptivitás. Ha a program megismeri a felhasználó szokásait, akkor például a leggyakrabban használt menüpontokat eleve nagyobb betűmérettel jeleníti meg, miközben a többinél meghagyja a kis méretet. De elképzelhető olyan meg-

donságok. Az EuroOffice-ban már megjelentek ezek a funkciók, maga a szoftver tartalmazza a szóban forgó jellemzőket. Az igazi újdonság azonban az, hogy az adaptív tulajdonságokat átvezettük a bővítményekébe is, így már a bővítmények is tudnak emlékezni” – hívja fel a figyelmet Banai Miklós.



A fejlesztőcsapat most azon dolgozik, hogy a teljes felhasználói felületet újra lehessen szabni, tehát a kis, dokkolható menüsorokat a felhasználó a mostaninál sokkal rugalmasabban használhatja, illetve igény esetén akár át is tudja programozni.

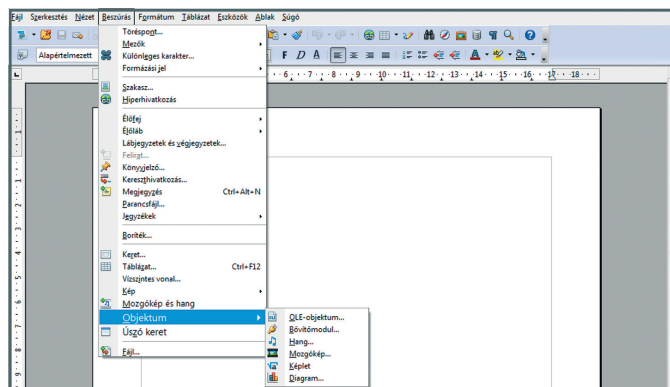
tárolt komplex dokumentumból – a kliens-szerver kapcsolat révén – mindig csak az éppen szükséges kis rész jelenik meg a mobilkészülék kijelzőjén. A másik lehetőség, amikor – megtartva az OpenDocument formátumot – eleve csökkentett funkcionalitását a mobilkészüléken tárolt dokumentum.

PREVENTÍV HIBAÉSZLELÉS

Banai Miklós egy további, szintén a bővítményekhez kapcsolódó fejlesztési irányvonalra is felhívja a figyelmet. A Szege-di Tudományegyetemmel kar-

öltve folyamatosan dolgoznak a forráskód minőségellenőrzésén, -javításán. A munka során nagyban támaszkodnak a Szege-di Egyetem különböző forráskódelemző eszközeire, amelyek ki tudják szűrni az adatokból a gyanús kódreszeket. Ezen hardvereket a Multiráció szakemberei továbbfejlesztették: behangolták az OpenOffice kódjára, illetve a bővítmények scriptjét is hozzátették a kódminőség-biztosító rendszerhez. A kódelemzések eredménye kettős: egyrészt kijavítják az OpenOffice kritikus, gyanús kódreszeit (a kódjavítások száma az ezres nagyságrendbe esik), másrészt a fejlesztések során nem követik el a már feltárt hibákat.

A forráskódelemző eszközt eredetileg a felhasználók által jelzett, ismert hibák felismerésére hangolták be, de a program a folyamatos továbbfejlesztéseknek köszönhetően ma már számos egyéb kódhiba preventív észlelésére is alkalmas. ■



ODF DOKUMENTUMOK MOBILESKÖZÖN

A szakember szerint az OpenDocument formátumú (ODF) dokumentumok mobil- és egyéb kis képernyős eszközökkel való kezelése az elkövetkező egy-két évben központi kérdéssé válik. A Multiráció

ilyen projektekkel is foglalkozik, a munka prototípus szintig jutott el.

A probléma jellemzően két irányból közelíthető meg. Az egyik lehetőség, amikor a mobilkészüléken csak egy egyszerű kliensprogram fut, és a szerveren

oldás is, hogy mindig azt a menüpontot nagyítja ki, ahol az egér mozog. Ezen funkciók révén a felület felhasználóbaráttá, a munka hatékonyabbá válik.

„Néhány éven belül várhatóan az összes mobilkészüléknél fontosak lesznek az adaptív tulaj-

Kérdések kulcsszavak helyett

Tim Berners-Lee két évtizede mutatta be a széles nyilvánosságnak a World Wide Web projektet. A világháló azóta mindennapjaink részévé vált, elképzelhetetlen nélküle élni, átalakította az emberi kommunikációt, társadalom- és gazdaságformáló szerepe felbecsülhetetlen. Írta: Kömlödi Ferenc

Mélyére ásní, bugyraiban tényleg intelligensen kutakodni a Google sikere ellenére is komoly problémákat okoz még. A keresés ugyan hatékony, de az információhoz való hozzáférés hiába tűnik gyorsnak, a relevancia egyre kevésbé felel meg a csúcstechnológiai környezet lehetőségeinek. **Kulcsszavak, feleslegesen hosszú dokumentumlista kezelése helyett célszerűbb és pontosabb lenne kérdéseket feltenni, és közvetlen, egyértelmű választ kapni rájuk.**

Ehhez azonban a keresőtornak szövegek végigfutása helyett alapentitásokat – embereket, helyeket, tárgyakat – kell azonosítani és feltárni a közöttük fennálló kapcsolatokat.

AZ ONLINE KERESÉS JELENE

„Megfelelő tudományos és anyagi befektetéssel a keresés mai formájára rövid időn belül ugyanolyan nosztalgiával tekinthetnénk vissza, mint a múlt tovatűnt technológiáira, például az elektromos írógépre és a bakelitlemeze” – véli a témakörrel hosszú évek óta foglalkozó *Oren Etzioni*, a Washington Egyetem (Seattle) Turing Központjának vezetője.

Az átalakulás azonban késik. Egyrészt a kutatói közösségek viszonylag kevés energiát fordítottak bonyolult válaszokat színtetizáló megoldásokra, másrészt, mert inkább a klasszikus, kulcsszavas keresés fokozatos fejlődésére fókuszáltak. Pedig referenciaanyagok végeláthatatlan katalógusa helyett célszerűbb lenne, ha természetes nyelven kapnánk választ kérdéseinkre. A fejlesztők hiába találnak ki a korábbiaknál praktikusabb módszereket a használható találatok rangsorolására (*ranking*), az átvizsgált honlapok száma napról napra

nő, a kereső milliárdnyi dokumentumot indexel, és akár a leg-egyszerűbb kutakodásra is képes milliányi találattal előállni. A jelen és a közeljövő adatáradataiban a felhasználó kérdését értelmező, az információból tényeket kinyerő, majd a helyes választ kiválasztó programra lenne szükség.

Az ismert keresőmotorok csak kis lépéseket tettek ebben az irányban. A Google hiába áll elő például speciális esetekben időjárás-előrejelzéssel, a megoldás egyáltalán nem általános, minden egyes kérdésre (egyelőre) lehetetlen egzakt felelettel szolgálni. A magát inkább „döntésmotorként” pozicionáló Bing a repülőjegy-kalkulátor kivételével többé-kevésbé ugyanúgy működik, mint a nagy rivális.

A Microsoft cambridge-i kutatórészlege (Egyesült Királyság) *Antonio Criminisi* vezetésével az emberi testet ábrázoló orvosi képeket néhány másodperc alatt indexelő (értelemszerűen speciális) keresőt fejlesztett, amely CT-röntgennél automatikusan megtalálja a szerveket és más szerkezeteket, így segítve a 3D-s fényképet tanulmányozó orvost. A kereső könnyebbé teszi röntgenképek kezelését és feldolgozását, sokat javítva a betegellátás minőségén. Csakhogy a szervek mellett a betegségek vizuális jeleit is célszerű lenne indexelni.

A 2009-ben indult és nagy médiafelhajtással beharangozott Wolfram Alpha elvileg az információ első olyan „globális tárháza”, amely ugyanúgy érti és válaszolja meg a tényszerű kérdéseket, mint ahogy feltettük neki. Irreleváns honlapokat és dokumentumokat is tartalmazó listák helyett strukturált és szakértők által hitelesített adatokból összeálló korpuszok feldolgozásán alapuló konkrét válaszokat ad,

emberi nyelven. Beütjük a szöveget, a gép számításokat végez, releváns szöveges és képi feleleteket generál.

Működése hasonlít *Douglas Lenat* 1980-as években elindított Cycjéhez. A Cyc folyamatosan bővülő adatbázist használva

Általában gyors választ, elsődleges forrásokat akarunk. Ambiciózus cél bonyolult kérdések azonnali megválaszolásához

von le következtetéseket. A mesterségesintelligencia-kutatás brit fenegyereke, *Stephen Wolfram* tálmánya ugyanígy következtet, ráadásul viszonylag kevés adatból is képes pontosan dolgozni. Másként, mint a sok választ indexelő, azokat a kutakodás tárgyával összevető, számukat – optimális esetben – egyre (remélhetőleg a helyesre) csökkentő szemantikus keresők.

A „józan paraszti ész”-ből – common sense – kiinduló Cyc rendszer számos szakmabeli véleménye szerint ez idáig a legelőrehaladottabb mesterségesintelligencia-kivitelezés.

A Wolfram Alphát (még béta-változatban) egyébként maga Lenat is tesztelte. Egyes kérdésekre valóban meglepően pontos választ ad. Például erre: „Túl részeg vagyok a vezetéshez?” A program javaslata: „Súlyunk és az elfogyasztott ital mennyisége alapján számoljuk ki a véralkoholszintet.” A megközelítés hátránya, hogy csak előzetesen nagyon pontosan definiált, kor-

látozott kérdéssorozatokra alkalmazható.

„A Cycnél szélesebb, a Google-nál szűkebb területre vonatkozó kutakodást kezel – állapította meg Lenat. – Az általa megjelenítették közül egyiket-másikat érti is. De csak az egyiket-másikat. Végeredményben, nagy mennyiségű kérdést nem tud elemezni és nagy mennyiségű elemzett kérdésre képtelen választ adni.”

A Microsoft és a Google egyaránt próbálkozik alternatív megoldásokkal. Előbbi 2008-ban vásárolta fel a természetes nyelven történő keresésre specializálódott Powerset startupot, utóbbi a filmekről, éttermekről és más helyi szolgáltatásokról érdeklődő felhasználónak válaszoló iPhone-alkalmazást kidolgozó, szintén startup Siri 2010-es akvizíciójával igyekezett újítani. Az igazság azonban az, hogy a két gigacég ilyen irányú kiadásai is csekély összegek a kulcsszavas keresésre fordított jelenlegi befektetésekhez képest.

A Google, a Bing és a Wolfram Alpha egyértelműen példazzák, hogy a keresők csak akkor tekinthetők intelligensnek, ha általános rendeltetésű „kérdés-válasz” képességekkel rendelkeznek. Jelenleg még a legfejlettebbek sincsenek ezen a szinten. Etzioni szerint mind a fejlesztői, mind a befektetői oldalon invenciózusabb megközelítésre, több kockázat felvállalására lenne szükség.

INFORMÁCIÓKIVONATOLÓ RENDSZEREK

Tényszerű információ automatikus kivonatolásával szövegekből az 1970-es évek óta foglalkozik a számítástudomány. A korai rendszereket nagyon jól körülírható, speciális és szűk témakörökre dolgozták ki, például az 1980-as években a Jasper Reuters-hírekből próbált pénzügyi információkat értelmezni. A vizsgálódási terület bővítése munkaintenzív folyamatnak bizonyult, ráadásul a hibázás lehetősége is nagyobb lett. Az újabb fontos változás a követ-

kező évtizedben érezte hatását: a régebbieknél sokkal automatizáltabb információkivonatoló rendszerek megjelenésével a számítógépek már nem manuálisan „barkácsolt” szabályok alapján nyertek ki mondatokból tényeket, hanem – gondosan összeválogatott példamondat-gyűjtemény alapján – maga a rendszer generálta automatikusan a szabályokat. **Az automatizált megközelítés modernebb, kisebb a hibaszázalék, viszont manuális beavatkozás nélkül még mindig nem elég hatékony, nem tud jól működő példagyűjteményt összeállítani minden egyes témakörre.**

Etzioni laboratóriuma 2007-ben bevezette a bármilyen topikra méretezhető, tetszőleges mennyiségű angol mondatra alkalmazható (nyílt forráskódú) nyílt információkivonatoló módszert, az entitások közti kapcsolatokat webes dokumentumok alapján „feltérképező” ReVerb programot.

Az alapötlet felettébb egyszerű: a legtöbb mondat megbízható szintaktikai fogódzókat ad ahhoz, hogy mit jelent. Az angol nyelv általában igékkel vagy igékkel és prepozíciókkal feje-

zi ki a kapcsolatokat. A számítógép sokszor teljesen egyértelműen megállapítja, hogy az adott mondatban hol az ige, majd azonosítja a hozzá kapcsolódó entitásokat, és felhasználva ezeket az ismereteket, az adott tényre vonatkozó állításokat hoz létre. Természetesen nem mindig hibátlanul, mert például a *Kentucky Fried Chicken* esetében úgy is következtethet, hogy „Kentucky állam megsütött néhány csirkét.” Viszont a szöveggyűjtemények, korpuszok, a nagy mennyiségű honlap egyértelműsítik, hogy ugyanazon állítás többféleképpen kifejezhető. Ha a rendszer különböző szerzők egymástól független azonos állítását sokszor kivonatolja, a kikövetkeztett jelentés valószínűsége exponenciálisan nő.

A nyílt információkivonatolás példamondatok témaspecifikus gyűjteménye helyett az internet szerteágazó topikjait lefedő általános modellen alapul, azon keresztül vizsgálja az információ (angol) nyelven keresztül való kommunikálását.

Más megközelítéseket tesztelve is születtek fontos eredmények az utóbbi években. A szakterületi fejlesztések – a Google, a Microsoft és számos startup, például a felhasználót a kinézett termék megfelelő időben történő megvásárlásában segítő Decide.

com mellett – híres amerikai felsőoktatási intézmények nevéhez fűződnek: Carnegie Mellon (Pittsburgh), Stanford (Palo Alto), New York Egyetem. A nyílt információkivonatolással ellentétben, ezek a megoldások egyelőre nem működnek automatikusan a teljes világhálón.

Egyik-másik projekt hiába talált ki mondatokat, jelentéstani mélységeikben is értő szupergyors rendszereket, azok csak speciális területeken (leginkább a pénzügyekben) vagy meghatározott műfajokban (Wikipédia szócikkek) működőképesek. Tudományos cikkek-

ben kutakodó, új kapcsolatokat és hipotéziseket feltételező masszív programokkal szintén kísérleteznek. Többségük egyelőre még nem teljesen automatizált, ráadásul olyan kihívásokkal is szembe kell nézniük, mint a leszűkített vizsgálódási kör (például Etzioni nyílt információkivonatoló módszerével kombinálva eredményesen kivitelezhető) bővítése.

A jelenlegi rendszerek távolról sem tökéletesek, amellyel, hogy igével kifejezett kapcsolatokra következtetnek, főneveket és melléneveket is jobban el kellene boldogulniuk. Mivel az információt forrása, szándéka és szöveggörnyezete határozza meg, célszerű lenne, ha ezeket és a nüánszokat is beépítenék az elemzésbe, valamint a rendszereket minél több nyelvre adaptálnák.

Úgy tűnik, az ambíció és a képzelőerő hiánya is akadályozza az információkinyerésről a kérdésfelelet megközelítésre történő váltást. A természetesnyelv-feldolgozással kapcsolatos kutatások jelentős része korlátozott feladatokra, mondatok jelentésének megfejtése helyett szintaktikai szerkezetekre, masszív korpuszokra nem méretezhető módszerekre vagy (a manuálisan annotált adatoktól való függés miatt) tetszőlegesen kiválasztott topikokra, esetleg bevált, de a szövegek mennyiségi növekedése miatt egyre bonyolultabb és hosszadalmasabb számítási műveletekre kényszerülő algoritmusokra összpontosít.

WATSON NYOMÁBAN

A DARPA 2009-ben indult *Gépi Olvasás* programja figyelemre méltó kezdeményezés: a résztvevő kutatóközösségek a jelenlegi módszereket a web méretére és sokszínűségére igyekeznek adaptálni, és értelemszerűen előbb-utóbb szembenéznek a „kérdésfelelet” jellegű keresés szemantikai és egyéb kihívásaival.


Etzioni az IBM Watsonját hozza fel pozitív példaként, a jövőt előrevetítő kivételként. Február egyik technológiai szenzációjaként az amerikai *Mindent*

vagy semmit (Jeopardy) vetélkedősorozat speciális fordulójában a szuperszámítógép legyőzte a show addigi két örökös bajnokát. A hardver sok szerver összességében, amelyeket a kvízmsorozat alatt a színpad mögé rejtettek egy megfelelő hűtéssel ellátott óriási helyiségben. Tíz darab 100 IBM Power 750 szervert rejtő rackben található, kapacitása mintegy 2800 feletti masszív számítógépnek felel meg, belső memóriáját például 15 terabájtra méretezték.

Elektronikus szöveggént érkező feladványnál bonyolult algoritmusokkal elemzi a szövegben fellelhető kulcsszavakat, viszonyait, a mondatok felépítését. A következő lépésben lehetséges válaszokat állít fel, és elvet más ötleteket. A válaszokhoz több mint egymillió könyvnek megfelelő szöveggörnyezetet és adatbázisokat használ. Az IBM mérnökei azt tanulmányozzák, hogyan lehetne Watson képességeit általános rendeltetésűvé (és érvényűvé) tenni, hogy tevékenysége túlmutasson a Jeopardy keretein.

Az általános rendeltetésű kérdés-válasz rendszerek pozitív hatással lesznek a webes keresésre. Amellett, hogy például a tudósok alaposabban átfésülhetik az online szakirodalmat, a tömör válaszok miatt a világháló mobil-eszközökön használók is eredményesen kutakodhatnak. **Mivel egyre többen neteznek parányi kijelzőkkel, és a mobil lehetőségeket választók száma drasztikusan nő, kereséseik eredményétől nem megijedni, hanem megörülni fognak.** A kulcsszavas módszer néha végtelennek tűnő találatlistája helyett azt és semmivel sem többet vagy kevesebbet kapnak annál, mint amire kíváncsiak voltak.

Például, ha a Barcelona mai meccsének eredményére kérdeznek rá, megtudják, hogy a csapat 5-0-ra nyert, ha a legközelebbi thai étteremre, pontos címet kapnak. Általában gyors választ, elsődleges forrásokat akarunk.

Ambiciózus cél, ám a bonyolult kérdéseket azonnal megválaszoló Watson sikere optimizmusra ad okot. 



Húsvéti nyúlnak öltözött rendszergazdák

Noha az önkéntes feladatvégzés eredménye nem váltható készpénzre, mégis egyre többen élnek a lehetőséggel az IT-szakmában is. A vállalatoknak pedig érdemes támogatni a jelenséget, mivel profitálhatnak belőle. **Írta: Szilágyi Szabolcs**

Képzelve el, hogy a napközben mogorvának tűnő rendszergazdát délután húsvéti nyúlnak öltözve látja viszont, az IT-projektmenedzserrel a helyi idősek otthonában találkozunk, ahol kertészkedéssel foglalkozik, az informatikai részleg vezetője pedig gyakran felbukkan az év végi időszakban, rögtönzött karácsonyi kórusokban dalokat énekelve. Furcsa lenne? Nos, az Egyesült Államokban közel sem az. A fentiek élő példák, ráadásul nemcsak a munkavállalók önkéntességére világítanak rá, hanem munkaadójuk támogatására is. Mindhárman cégük beleegyezésével végzik tevékenységüket, amelyre egyébként nem is olyan ritkán fizetett munkaidejükben kerítenek sort.

De miért éri ez meg a vállalat számára? Hiszen az alkalmazott közvetlenül a céges erőforrásokat csapolja meg azzal, hogy munkában töltött idejének egy részét „saját” céljaira fordítja. Nos, a helyzetnek több hozadéka van a vállalat számára, mint amennyibe kerül: egyrészt a munkavállaló valóban olyan célokért tehet, amelyekkel egyetért, ám az a helyi közösség érdekeit is szolgálja – többnyire annak a közösségnek az érdekeit, amely a vállalat ügyfélkörének jelentős részét adja. **Ezáltal a cég reputációja is növekedhet abban a környezet-**

ben, ahol az önkéntes munka folyik. Ez akár ahhoz is hozzájárulhat, hogy a vállalat nagyobb érdeklődést váltson ki a fiatalabb, közösségi életre fogékonyabb és technikához jobban értő munkavállalók körében.

GÖRDÜLEKENYEBB EGYÜTTMŰKÖDÉS

Az önkéntesség jó hatással van a „hivatalos” munkára is. Már számos gazdálkodó szervezet felismerte, hogy közvetlen kapcsolat van a közösség érdekében tett erőfeszítések és a munkavégzés termelékenységére, illetve a csapatmunka fejlődésére között. Hasonló előnyökkel szolgál a gondosan megválasztott önkéntes feladat-



vállalás, mint egy csapatépítő tréning. Utóbbi pedig a legtöbb cégnél valamilyen formában létező jelenség.

„A cégen kívül végzett önkéntes tevékenység lehetővé teszi az alkalmazottak számára, hogy saját kollégáikat egy másik szemüvegen keresztül, más szemszögéből nézhessék” – állította a jelenséget bátorító amerikai Reed Technology vállalat informatikai igazgatója, *David Ballai*: „A megszokottól eltérő környezetben kell helytállniuk, egy közösség érdekeit szem előtt tartva. Miután visszatérnek munkájukhoz, sokkal holisztikusabb szemléletmódot vallanak kollégáikkal és ügyfeleikkel szemben is, jobban megértik, hogyan működik a világ. Nagyon jó csapatépítő!” (az önkéntes feladatvállalás – a szerk.).

Ballai állítását erősítette meg *Marcia Riley*, az ESI Internationalnél emberi erőforrásokkal foglalkozó alelnök is. Elmondta: a 30 alattiak hajlamosak igazán önkéntes feladatvállalásra, gyakran rá is kérdeznak az ilyen lehetőségekre egy-egy állásinterjúban. Riley elárulta, hogy 20 évvel ezelőtt nem találkozott ilyen igényekkel a felvételi elbeszélgetések során, vagyis egy relatíve új jelenségről van szó. „A fiatalabbak igénylik ezeket az előnyöket, a jó munkaadó pedig válaszol rá-

juk” – foglalta össze véleményét az ESI alelnöke.

MI KÖZE AZ IT-NEK AZ ÖNKÉNTESSEGHEZ?

A szektorban dolgozók nemcsak húsvéti nyúlnak öltözve válhatnak önkéntesen a társadalom hasznára, hanem akár saját szakértelmüket kihasználva is segítségére lehetnek másoknak, például olyan gazdálkodó szervezeteknek, amelyek ugyan rászorulnának erre a tudásra, de megfizetni nem tudják azt. Jó példa erre az USA-beli Comerica esete: a bank vezetői mindig úgy gondolták, hogy a pénzügyi tevékenység a bizalomról szól, ezért teret adtak az alkalmazottaiknak önkéntes feladatvégzésre. Ennek során aktívan közreműködtek olyan csapatmunkákban, ahol szakértelmükre volt szükség. Így kaphatott a detroiti állatkert informatikai rendszere teljes átvilágítást és a szegényeket segítő helyi ételmisszer-elosztó központ osztályozó-rendszert a konzerv élelmiszerek katalógizálásához – ingyen.

Az ilyen tevékenységek során az alkalmazottak a megszo-

kott környezetben tapasztalható nyomás, bizonyítási kényszer nélkül használhatják képességeiket. Anélkül tehetnek valamit a rászorulókat informatikai igényeinek kielégítéséért, hogy közben az „általános” munkavégzéskor megszokott és átélt stresszel kellene szembesülniük. Így egyfajta lazításra nyílik lehetőségük amellett, hogy használják tudásukat.

A csoportmunkát erősítő tényezőről számolt be az egészségügyben tevékenykedő Texas Health IT-részlege is. A *Healing Hands, Caring Hearts* („Segítő kezek, gondoskodó szívek”) motót alkalmazó vállalat szintén felismerte az önkéntességben rejlő erőt, ezért havonta 8 órányi fizetett, de a közösség javára felhasználható munkának biztosít teret. Ennek pozitív hozadékáról *Ed Marx*, a vállalat informatikai szakértője számolt be.

„Nemcsak a közösztudatosság tekintetében voltak nagy hatással ránk a feladataink, de az önkéntesen vállalt tevékenységek szemléletformálónak is bizonyultak. Felfrissülve láttunk neki munkánknak, amit nagyban elősegített az is, hogy tisztába kerültünk azzal: cégünk nem »csak« egy munkahely. Már tudjuk, mi a szerepünk az egészségügyben, és azt is, hogy ezen belül mi az IT feladata” – lelkesedett az informatikai vezető.

Emellett **olyan képességekre derült fény, amelyek a hétköznapi munka során rejtve maradtak. Egyes alkalmazottakból természetes módon tört elő a vezető, holott a mindennapok feladatvégzése során beosztottként tevékenykedtek. Marx fontosnak tartotta azt is kiemelni, hogy az informatikai részleg vitálsababá vált.** Normál esetben a rendszergazdák egész nap csak ülnek a monitor előtt, kezelik a rendszer hibáit, rendellenességeit, hallgatják a felhasználók panaszait – vagyis nem igazán mozgatják meg magukat. Ezzel szemben az önkéntes feladatvégzés (jelen esetben közösségi házépítés) akár alapos tornának is fel-fogható, így az alkalmazottak élete kiegyensúlyozottabbá vált.

A tapasztalatok szerint a mérnöki tevékenységgel és informatikai feladatokkal foglalkozókat nem közvetlenül szakírányú tudásuk miatt is szívesen látják az önkéntes feladatvégzésben. Ezek az emberek ugyanis jók problémafelismerésben és -feltekerkezésben. **Az IT-részleg dolgozóknak gyakran szigorú határidőknek kell megfelelniük, ráadásul sok tennivalójuk akad, amit viszonylag kevés pénzből kell megvalósítaniuk. Így képesek meglátni a lehetőségeket az önkéntes munkavégzés során is, illetve feladatukat hatékonyan tudják elvégezni.**

HAZAI HELYZET

Ugyan 2011 az Önkéntesség Európai Éve, itthon ez mégis kevésbé jellemző munkavállalói oldalról, inkább vállalati szinten jelentkezik. Utóbbira rengeteg példát láttunk az elmúlt időszakban. *Christopher Mattheisen*, a Mindentudás Egyetemét alapító Magyar Telekom vezérigazgatója például idén januárban arról szólt, hogy az ME 2.0 a távközlési cég olyan CSR- (Corporate Social Responsibility – vállalati társadalmi felelősségvállalás) tevékenysége, amivel felelősséget vállalnak a jövő társalmáért.

Úgy gondolták, hogy a gazdasági krízis ellenére is fenn kell tartani az efféle társadalmi felelősségvállalási tevékenységet, sőt tovább kell bővíteni azt, mert a válságban talán még nagyobb szükség van az innovációra. Az ME 2.0 csökkenti a digitális szakadékot, népszerűsíti az e-learninget, javítja a társadalmi mobilitást, és végeredményben az ország versenyképességét.

Egyéni szinten is működik az önkéntesség a kommunikációs vállalat hazai leányánál. Erre szolgált példát az a 18 játszótér, amelynek felújításához a cég alkalmazottai is hozzájárultak az ezredfordulón, többek között azal, hogy a játszótér alapjainak kiásását saját maguk végezték el. 2005-ben a „Vigyázz a madárral!” program keretében három nemzeti parkban építettek fel túrapontokat és madármegfigyelő

tornyokat. Ezt követte a 2007-es Ócsai Madárvárta Egyesületnek eljuttatott 4 millió forintos adomány, a leégett oktatóház újjáépítéséhez a dolgozók kétkézi munkájukkal is hozzájárultak.

A Samsung Electronics Magyar Zrt. képviselőjében pedig *Si Ho Jang*, a jászfényzarui Samsung-gyár elnöke önköltségen 10 millió forint értékű adományt adott át szeptember 30-án a 2010-es vörösiszap-katasztrófa során legnagyobb kárt szenvedett településeknek. Az adományt képező 189 Samsung televíziókészülékét és monitort a Magyar Kármentő Alapon keresztül Devecser, Somlósárhely és Kolontár kapja. A dél-koreai vállalat számára nem volt ismeretlen korábban sem az ilyen te-


és családjaik részére nyújt rövid kilépést a kórral való küzdelemből. Noha számos céges támogató segíti munkájukat, néhány vállalat alkalmazottai „cimboraként” is részt vállaltak a gyerekek felvidításában. Például a gyógyszeripari GlaxoSmithKline Kft. alkalmazottai feladatot vállaltak a táborkban, az üzletviteli és vezetési tanácsadással (is) foglalkozó Deloitte Magyarország munkavállalói pedig saját maguk vettek részt a hatvani tábor felújításában.

A hazai kezdeményezések javát a B&P Braun & Partners által életre hívott, idén negyedik évébe lépő Good CSR program fogja össze. A kutatás a hazai vállalatok körében térképezi fel a felelős vállalati működés ál-



vékenység, hiszen 2010-ben 20 millió forint értékű adományt juttattak el az árvízkarosultaknak, emellett platina fokozatú támogatói a magyarországi SOS Gyermekfalvaknak és arany fokozatú támogatói a Magyar Olimpiai Bizottságnak immár 15 éve. A tavalyi vörösiszap-katasztrófa károsultjainak megsegítése egyébként nem „fentről” indult kezdeményezés, hanem azt az alkalmazottak vetették fel, amelyet felkarolt a vezetőség.

Említést érdemel a Bátor Tábor kezdeményezés, amely daganatos, cukorbeteg, krónikus ízületi gyulladással, valamint haemophiliával kezelt gyermekek

lapotát, a társadalmi felelősségvállalás különböző formáinak elterjedtségét. Szerencsére a felmérés pozitív tendenciákat mutatott ki hazánkban. Amint írja, jellemző az átgondolt stratégiai tervezés a társadalmi felelősségvállalással kapcsolatos programok megvalósításakor is. Ennek egy példája a vállalati önkéntesség, amelyre a február 1. és március 31. között zajló felmérésben részt vevő 53 cég kétharmadánál van jelenleg lehetőség. A Good CSR program megállapítása szerint a közösségi jellegű önkéntesség valamivel elterjedtebbnek tekinthető Magyarországon, mint a szakértői típusú. 



COMPUTERWORLD ONLINE

VIDEÓK: emberek, események, termékek

CIO.HU: az informatikai vezetők fóruma

Mobilon is!

HÍREK ÉS ESEMÉNYEK az IKT-piacról

Olvassa el, ami történt

Tudja meg, ami történni fog

WHITEPAPER: a tudásbázis

CÉGINFÓ: az IKT-adatbázis

www.computerworld.hu

HUNGARIAN SOFTWARE TESTING FORUM

IDŐPONT: 2011. október 13–14.

HELYSZÍN: Budapest, Danubius Hotel Helia (1133 Budapest, Kárpát utca 62–64.)

A konferencia angol nyelvű, de megfelelő számú igény esetén tolmácsot biztosítunk.

A TERVEZETT PROGRAM

2011. OKTÓBER 13. CSÜTÖRTÖK

- **Lloyd Roden:** A szoftvertesztelés jelentősége és kihívásai (keynote)
 - **Esettanulmányok**
 - **Rex Black:** A menedzselt szoftvertesztelés vállalati bevezetésének előnyei és szükségessége
 - **Moderált kerekasztal-beszélgetés**
- Este kötetlen beszélgetési lehetőség az előadókkal

2011. OKTÓBER 14. PÉNTEK

- Párhuzamos workshopok
- **Lloyd Roden:** Becléstechnikák a szoftvertesztelésben: hogyan is végezzük sikeresen? A sikeres tesztmenedzser eszköztára – amire feltétlenül szükség van, amit mindenképpen ismerni kell
 - **Rex Black:** Rizikóalapú tesztelés eredményes alkalmazása a gyakorlatban

A részletes program a <http://computerworld.hu/konferencia/55> weboldalon található.

Jelentkezni a konferencia@idg.hu e-mail címen lehet.

További információk:

Bíró Ilona

Telefon: +36-1/577-4374

Fax: +36-1/266-4274

E-mail: ilbiro@idg.hu

Héjjas Ágnes

Telefon: +36-1/577-4314

Fax: +36-1/266-4274

E-mail: ahejjas@idg.hu



SZÁMÍTÁSTECHNIKA
COMPUTERWORLD

KIKNEK SZÓL?

- Szoftvertesztelőknek
- Tesztmenedzsereknek
- Tesztkoordinátoroknak
- Tesztelési vezetőknek
- Szoftvertervezőknek
- Szoftverfejlesztőknek
- Rendszeresztelőknek
- Tesztmérnököknek
- IT-igazgatóknak, -vezetőknek
- IT-projekt-vezetőknek
- IT-fejlesztési vezetőknek, menedzsereknek
- Informatikai szakembereknek

PARTNEREINK

GOLD PARTNER



ERICSSON



mortoff

SILVER PARTNER



KIÁLLÍTÓ PARTNER



IT Services



Lufthansa Systems

IT that makes your life easier



BUSINESS TRAVELLER HUNGARY

Az üzleti utazás hazai iránytűje

A magazin, amely bepillantást enged a céges utaztatás kulisszatitkaiba és hasznos tanácsokkal, praktikus ötletekkel segíti az utazó üzletembereket.



COMPUTERWORLD
CW BUSINESS

Mit akar a CFO az IT-tól?

COMPUTERWORLD – A világ IT-szemmel

A lap, amely megmutatja, hogyan lesz az információtechnológiából üzlet!

Cégvezetőknek, pénzügyi vezetőknek, informatikai vezetőknek

**40%
KEDVEZMÉNY
+ AJÁNDÉK
CORPORATE CLUB
KÁRTYA***

SZÁMÍTÁSTECHNIKA
COMPUTERWORLD

**1 ÉVES COMPUTERWORLD-ELŐFIZETÉS
1 ÉVES BUSINESS TRAVELLER HUNGARY ELŐFIZETÉSSEL**

Fizessen elő vagy hosszabbítsa meg előfizetését!

26940 Ft helyett **most csak 16200 Ft**

Hívja a **06-1/577-4301**-es telefonszámot vagy kattintson a **piacter.idg.hu** oldalra!

*A kártya névre szól, egy évig érvényes. Egyedülálló kedvezményekre jogosít hazai és külföldi turisztikai szolgáltatóknál. A kedvezmények magán- és üzleti utakhoz is felhasználhatók.

Az akció az IDG Hungary Kft.-nél 2011. december 15-ig megrendelt és befizetett előfizetésekre vonatkozik. Az előfizetés időtartama alatt az előfizetés nem mondható fel. További információért hívja a **06-1/577-4301**, nem emelt díjas telefonszámot vagy írjon a **terjesztes@idg.hu** e-mail címre. Megrendelése egyben önkéntes adatközlés is. Az adatközlő hozzájárul, hogy megadott adatait a kiadó előfizetői adatbázisában nyilvántartsa és az előfizetői akcióban szereplő másik kiadvány kiadójának átadja. A megrendelő megrendelésével továbbá hozzájárul, hogy a kiadó tájékoztató- és reklámanyagot küldjön marketingcélből. A hozzájárulás visszavonásig él, a kiadó címére (IDG Hungary Kft. 1075 Budapest, Madách I. út 13-14. A.ép. IV.em.) írt levélben bármikor visszavonható. Minden jog fenntartva!