

Az új népiesség

A Unix építőmesterei

A lakályos jelző a forráskód esetén azt jelenti, hogy a programozó több évvel később is előveheti az anyagot és képes átlátni, illetve biztonságosan módosítani.

Richard Gabriel

Nem beszélhetünk a programokról anélkül, hogy át ne vennénk a programgyártás nyelvezetét. Fejlesztőknek, tervezőknek és mérnököknek nevezzük magunkat. Szerkezetekről, terekről, objektumokról, keretrendszerekről, szintekről, rétegekről, összetevőkről és felületekről beszélünk. Az elkészült munkát azután úgy kezeljük, mintha valami ingatlan lenne, hiszen különböző címekhez kapcsolódó (web)helyeket építünk köré.

Mi folyik ebben a kusza nyelvezetű világban? Már jó ideje az a véleményem, hogy a nyelvhasználat azt jelzi, hogy a programipar kezd éretté válni – olyan iparág alakul ki belőle, amelyet képzett és elismert szakemberek határoznak meg, nem pedig az egyforrásos, úgynevezett „felületszolgáltatók”. Az érett programiparban a Microsoft semmivel sem több vagy kevesebb, mint mondjuk a Georgia Pacific vagy a Kaufman & Broad.

Az építőiparban a fejlesztés teljesen nyilvánosan zajlik. Az építéshez használt anyagok és eljárások nem titkosak, éppen ezért a fejlesztéshez nagyon sok ember tapasztalatait fel lehet használni.

A programipar mindössze néhány évtizednyi múltat tekint vissza, ezzel szemben az építészet egyidős a kultúrával. Ez azt sugallja, hogy talán elleshetnének néhány dolgot a régebbi iparágtól. Úgy tűnik, hogy *Gabriel Richard* is egyetért ezzel, amikor *Patterns of Software* című könyvében kihangsúlyozza annak fontosságát, hogy programjainkat lakályosra tervezzük. Az eszményi szerkezetet egy New England-i tanyaházhoz hasonítja: a végeredmény egy kissé kusza, de minden egyes részlet jól megfelel az igényeknek, és jól illeszkedik a többihez. A lakók könnyen megváltoztathatják a saját környezetüket, mivel azt olyan sablonok szerint építették, amely megfelel a családnak. Ezek a sablonok magukban hordozzák a kis darabokból építkezés lehetőségét. *Stewart Brand* a *How Buildings Learn* (Hogyan tanulnak az épületek?) című könyvében a New England-i tanyaházat a népiesség építkezés tökéletes példájának nevezi, és a következőt írja: „Az, hogy az építők és a lakók mit visznek át az egyik épületből a másikba, semmiféle szabályt nem követ, teljesen esetleges, az eljárás mégis mindig kellőképpen körültekintő és előrelátó.”

Az építésztörténeiszek az 1850-es években vették át a nyelvészekről a „népiesség” kifejezést. A nyelvészetben ez a kifejezés egy adott terület saját nyelvjárására vonatkozik. Mindennapit jelent a szó mindhárom értelmében: elterjedt, általános és észrevétlen.

Az építészetben a népiesség épületeket az jellemzi, hogy szöges ellentétben állnak mindennel, ami tudományos, magasos és művészi. Népiességnek nevezünk minden olyan épületet, amelyet nem profi építészek terveztek – más szóval a legtöbb épületet. A népiesség építészeti hagyományok első sorban arra összpontosítanak, hogy miként lehet a több nemzedék során összegyűjtött tudást felhasználni az olyan hosszú távú gondok megoldásánál, mint amilyen például egy épület folyamatos karbantartása és fejlesztése. Ezzel szemben a profi építészet előszeretettel keres új megoldásokat a régi nehézségekre, s ez sokszor szerencsétlenséghez vezet. A népiesség épületek folyamatosan fejlődnek. Miközben az új épületek



nemzedékei egyre érettebbek és kifinomultabbak lesznek, megőrzik egyszerűségüket.

Használhatjuk-e a „népiesség” jelzőt a Unixra? *Neal Stephenson* az *In the Beginning was the Command Line* (Kezdetben volt a parancssor) című könyvében a következőt írja: „A Unix gépek fájlrendszerei mind ugyanazt az általános szerkezetet követik. Egy gyatra operációs rendszer alatt könyvtárakat bármilyen eszement névvel elláthatjuk, és oda tehetjük őket, ahová csak akarjuk. Unix alatt azonban a fájlrendszer legmagasabb szintjét (a főkönyvtárat) mindig ugyanaz a karakter, a / jelöli, és ez a könyvtár mindig ugyanazokat a felsőszintű könyvtárakat tartalmazza: /usr, /etc, /var, /bin, /proc, /boot, /home, /root, /sbin, /dev, /lib és /tmp.

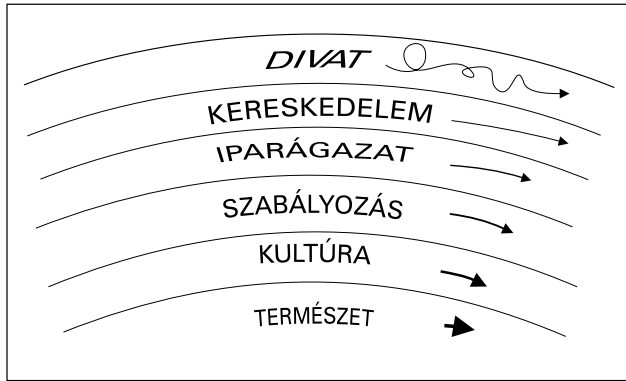
Általában ezeknek a könyvtáraknak a belső felépítése is meglehetősen szabályos. A könyvtárnevek szinte mindig rövidítésekkel állnak, és szinte sehol sem találkozunk nagybetűkkel. Ugyanis ezt a rendszert olyan emberek találták ki, akik képtelenek elviselni a rendtelenséget. A hosszú nevek hárombetűs rövidítésekkel koptak le, pontosan úgy, mint a folyómederben csiszolódkövek.”

Ez a fajta kulturális hatás az, amelynek köszönhetően a Unix-hívők annyira bíznak a rendszerükben, és ez az, amely nyugodt, megrendíthetetlen, idegesítő felsőbbrendűséget kölcsönöz számukra. A Windows 95 és a MacOS olyan termékek, amelyeket a mérnökök egy-egy vállalat berkein belül fejlesztettek ki. Ezzel szemben a Unix nem nevezhető a szó szoros értelmében terméknek, mivel pontos összeállítása tulajdonképpen szájhagyomány útján alakult ki.

A népiesség építészet pontosan az ellenkezője annak, amit *Brand* „magazin építészetnek” nevez, mely szerinte az építészetet művészetként, nem pedig mesterségként kezeli. *Henry Glass* így fogalmazza meg a különbséget: „Ha egy kényelmi szolgáltatás uralkodik, akkor művészetről beszélünk, ha egy gyakorlatias szolgáltatás veszi át az uralmat, akkor pedig mesterségről.” Nehéz gyakorlatiasabb dolgot elképzelni, mint egy parancssor.

Stewart Brand szerint rengeteg a szakértelem az általa „Low Road” jelzővel illetett épületekben. Olyan építményekről van szó, amelyek nem feltűnőek, alacsonyak a költségeik, és egyáltalán nincs stílusuk. *Brand* szerint: „A világban a legtöbb munka Low Road épületekben zajlik... és még a gazdag társadalmakban is a legötletesebb alkotóerőt, főleg a fiatalos alkotószellemet, ilyen épületekben találhatjuk meg.” Valószínűleg nem véletlen egybeesés tehát, hogy a mintaszerű Low Road épület példaként a MIT 20-as épületét említi, ahol a harmadik emeleten lévő Vasútmodell klub adott otthont az 1960-as évek elején, a számítógépes betyárok első nemzedékének, akik egy sor forradalmi módszertani újítást indítottak útjukra.

Hajlamosak vagyunk azt gondolni, hogy a forradalmak gyorsan történnek, az a forradalom azonban, amely a MIT 20-as épületében kezdődött, megelőzte Moore törvényét, amelyet elsőként 1965-ben tettek közzé. Ezenkívül megelőzi magát a számítógépipart is majd egy teljes korosztállyal. Bárhogyan nézzük is, a betyárvilág hosszú ideig a számítástechnika alapjaként szolgált, hiszen a folyton változó divatok és forradalmak ellenére is viszonylag állandó maradt. Nem állítom, hogy a Unixnak nincsen kereskedelmi oldala, mindössze azt mondom, hogy kultúrális alapjai mélyebben gyökereznek, mint a kereskedelmiek. Az *In The Clock of the Long Now* című könyvében *Stewart Brand* a művelt világot hat rétegre osztja fel, amelyek az idő múlásával eltérő mértékben változnak. A felosztást ekképpen ábrázolja:



A alsó rétegek lassabban, a magasabban lévő pedig gyorsabban módosulnak. A gyors rétegek újításokat, a lassúak pedig megbízhatóságot hoznak. A tanulás így a folytonossággal társul. – állítja Brand.

A betyárvilágot és a programkereskedelmet egyaránt új és igen érdekes szemszögbe helyezi. A betyárvilág, amely egy szinttel a természet felett helyezkedik el, gondosan őrzi a programok mesterkéletlen értékeit. A GNU kiáltvány szerint például az ingyenes program olyan, mint a levegő. Egy szinttel feljebb megtaláljuk a szabályozást, a betyárvilág állandó kínlását a felhasználási szerződésekkel. Ezek a szerződések egy réteggel feljebb ipari ágazatot alkotnak: olyan programok és protokollok ezek, melyek annál értékesebbek, minél több helyen vannak jelen. Talán nem túlzás azt mondani, hogy a betyárvilág természetes módon ráértett arra, hogy mi tesz egy programot értékessé. Ezt az elgondolást láthatjuk megtestesülni az Internetben is, amely legalább három természetes értéket mondhat magáénak: senki sem birtokolja, mindenki használhatja és bárki továbbfejlesztheti.

Míndezek a lehetőségek beépültek az Internetbe (és az ingyenesen elérhető programfejlesztő eszközökbe), elsősorban anyagi okból.

Ez az oka annak, hogy a GNU projekt azt mondja: „az embereknek lehetőséget kell adni arra, hogy szabadon használhassák a programokat minden olyan célra, amely a társadalom számára hasznos”. A GNU eszközök és a többi ingyenes program mellett ezt az elvet követi a Linux, a Bind, a TCP/IP, a sendmail, az Apache, a Jabber és a SOAP is. Ezeket a programokat (javarészt) senki sem birtokolja, mindenki használhatja és bárki továbbfejlesztheti.

Ezek azok az értékek, amelyek nem az üzleti életből jönnek (és nem is jöhetnek onnan). Ezek nem kereskedelmi értékek. A program kultúrarétege azonban – amit kereskedelemnek nevezünk – attól a társadalmi háttértől függ, amely a betyárvilágból és általában véve a Unixból fejlődött ki. Nyilvánvalóan ide tartozik az Internet, és ezenkívül még sok más célszerű és igen elterjedt eszköz. Ezek ingyen vannak, mint a levegő.

Nagyon fontos, hogy megértsük azt, hogy milyen viszonyban állnak egymással ezek a rétegek, mivel számtalan félreértés és rossz döntés születet úgy, ha például a kereskedelmi érdekek saját javukat szem előtt tartva próbálnak meg kialakítani egy iparágat, vagy amikor a szabályozás megpróbálja átalakítani a közösség igényei szerint a kereskedelmet.

Michael Polany azt mondja, hogy az átfogó tudatok (ilyen például a társadalom) a valóság különböző szintjeinek logikus együttese, és léteznek olyan elvek – úgynevezett kerületi feltételek –, amelyek révén az egyes szintek biztosítani tudják azokat a feltételeket, amelyekről a felsőbb szintek függenek. *Phil Mullins* ezt a következő módon fogalmazza meg: az alacsonyabb szint olyan korlátokat szab, amelyeket a felsőbb rétegek nem tudnak áthágni. Az alacsonyabb réteg meghúzza a határokat, de szabadon hagyja a lehetőségeket. Egy alacsonyabban lévő szint nem határozza meg pontosan a magasabb szinteket... egyetlen szintnek sincs hatalma a saját kerületi feltételei

főllött, így nem is létezhet egy magasabb szinten. Ennek megfelelően tehát, az elektronikus kereskedelmet nem az Internet hozta létre, mégis igaz az, hogy az elektronikus kereskedelem teljes mértékben az Internetre van utalva.

Miközben tehát az ingyenes programokat támogató megmozdulások a programipar kereskedelmi övezete számára kereskedelemellenesnek tűnhetnek, ezek a kereskedelmi és a divatrétgtől teljesen függetlenül történnek. Ezek sokkal tartósabb, tisztán műveltségi, természetes dolgok. Ha megnézzük a piaci ágazatot szegélyező kerületi feltételeket, láthatjuk azt is, miért ütözköznék gondokba a programkészítő cégek olyankor, amikor az iparágukat felülről lefelé próbálják meg beilleszteni a rendszerbe. A Microsoft talán sikeres volt ebben, de csak átmenetileg.

Manapság mindannyian (beleértve a Microsoftot is) azt érezhetjük, hogy a program alulról felfelé kezd növekedni, és ez elsősorban az Internetnek köszönhető, illetve annak a kultúrának, amely azt létrehozta. Láthatjuk, hogyan fejlődnek az olyan protokollok, mint a SOAP vagy az XMP/RCP, amelyeket a Világhálón történő közlés elősegítésére fejlesztettek ki. Ezek a módszerek teljesen nyitottak és bárki számára hozzáférhetőek. *Dave Winer* és vállalata, az Userland, mindkét fejlesztésben részt vett. Ugyanez igaz a Microsoftra és a Developmentorra is. Amikor megkérdeztem Dave-től, hogy hogyan zajlott a fejlesztés, azt mondta, hogy a Microsofttal és a Developmentorral végzett közös munka volt a legeredményesebb azok közül, amelyekben cégével részt vettek.

A Microsoft a személyi számítástechnikából érkezett, és örökre ott is fog maradni. Egy olyan világban is boldogulnia kell azonban, ahol a számítástechnika egyre inkább a szűkebb-tágabb környezetéhez csiszolódik. A Unix már a kezdetek óta fel volt készítve erre.

Craig Burton arra hívja fel a programkészítő cégek figyelmét, hogy „az igazi kihívás az, hogy mindenütt jelen lévő iparágazatot építsünk ki, miközben folyamatosan növeljük részvényeink értékét”. A feladat egyáltalán nem könnyű, a megoldásban azonban két dolog is segíthet. Az egyik, hogy összhangba kerüljünk azokkal a fejlesztő-mémókeinkkel, akik az ágazatot készítik, a másik pedig az, hogy képesek legyünk felismerni, hogy mi válik be hosszú távon is.

A Long Now alapítvány hét alapvető fogalmat meg azon cégek számára, amelyek hosszú életűek és értékálló szeretnének maradni:

- Szolgálja a hosszú távú tervek.
- Vegye kézbe a felelősséget.
- Jutalmazza a türelmet.
- Figyeljen a mélyebb értelemre.
- Lépjön szövetségre a versenytársakkal.
- Ne kösse magát semelyik oldalhoz.
- Használja ki a hosszú távú lét előnyeit.

A cégek állandóan tanulnak – még azok is, amelyeket elsodort a divat. Vegyük például az Apple-t. A legjelentősebb változás, amely az Apple berkein belül történt az elmúlt évben, nagyon távol áll a divattól, talán éppen ez az oka annak, hogy keveset is hallhattunk róla. A változás a szabályozás szintjén történt. Miután hetven ezer szaki együtt látott neki az Apple BSD-alapú Darwin operációs rendszerének továbbfejlesztéséhez, az Apple válaszul a „természetes” irány felé mozdította el a forráskódra vonatkozó felhasználási szerződést. Az új szerződés természetesen még nem olyan, mint a GPL, de sokkal közelebb került ahhoz. *Chris Bourdon*, az OS X termékmenedzsere azt mondta: „Mégfoghatod a Darwint, és azt tehetsz vele, amit akarsz. Bárki hozzáférhet.”

Még szép, hiszen ez is Unix.



Doc Searls

a Linux Journal szerkesztője és a Cluetrain Manifesto társszerzője.

- www.cluetrain.com
- www.longnow.org