

## **Az egyedi fejlesztés vagy keretrendszerek, melyiket szeressem?**

Hogy egy megrendelőnek megfelelő lesz-e egy keretrendszeres fejlesztés, az olyan döntés, aminek a meghozatalához szükséges ismeretanyag általában nem áll az ügyfél rendelkezésére, ezért szakértő segítségre szorul. A döntéssel járó kockázatokat és következményeket rendszerint nem ismertetik az ügyféllel, az ezek megoldásával járó lappangó extra költségeket pedig az ajánlatok nem tartalmazzák, ezáltal nagyságrendekkel kedvezőbb ár illúzióját keltve.

Így nem csoda, hogy gyakorta éri az a kritika az egyedi fejlesztésű rendszereket készítő fejlesztőcégeket, hogy díjaik túl magasak, mert mások töredék áron kínálják ugyanazt a megoldást. Egyrészt az ugyanaz ritkán ugyanaz, másrészt pusztán ár alapon versenyeztetve az ajánlatokat figyelmen kívül hagyunk sok olyan tényezőt, melyek potenciális problémákat és fenntartási többletköltségeket rejtenek.

### ***A keretrendszerekről***

A keretrendszeren alapuló fejlesztések lényege, hogy a weboldalt készítő fejlesztő a rendszer alapjait, és a fejlesztések oroslánrészét kitevő szerkesztőségi rendszert készen kapja, és ezt idomítja, bővíti valamelyest, hogy a kész keret közelítsen a megrendelő igényeihez.

Ezzel nyilván időt lehet spórolni, az idő pedig pénz, mint a mondás tartja, de egy másik mondás szerint semmi sincs ingyen, így van ez a spórolással is. Kinek is

spórol ez valójában időt és pénzt? Kevés kivételtől eltekintve a fejlesztőnek. Három, hosszú távon fontos, de rendszerint alulértékelt problémaforrást vethet fel a keretrendszerekre épülő kivitelezés, melyek:

#### a) **Rugalmasság a fejlesztés során**

Ha szeretnénk elkerülni az "azt úgy nem lehet" típusú fejlesztői válaszokat, akkor nem a keretrendszerek a legjobb választás. A témázott, modulokkal megtűzdelt keretrendszer pont annyit tud és pont úgy, ahogy azt az eredeti fejlesztők és a modulfejlesztők megálmodták. Nem a megrendelő, és nem a weboldalt fejlesztők, hanem vadidegenek. Ha ez a megoldás kevés, akkor rendszerint vagy elutasítás, vagy kompromisszumos látszatsmegoldás a vége.

Panelmondatok:

*"az nem megoldható, hogy ott inkább három kép legyen"*

*"megoldottam, hogy három kép legyen, de a két extra képet nem ugyanott lehet feltölteni, hanem van neki egy külön kis felület"*

*"a következő verzió már tudni fogja"* (bocs, azt hittem programozói szakértelemért fizetek)

#### b) **Jövőállóság a fejlesztést követően**

Ha az a fajta weboldal tulajdonosok vagyunk, aki két hónap elteltével veszi észre, ha nem működik az oldala, akkor nyugodtan válasszuk a legolcsóbb keretrendszeres megoldást, édesmindegy, hogy milyen módon készül el

oldalunk, ha csak porfogónak van az interneten.

### **Passzív jövőállóság**

Az alap keretrendszer szolgáltatásokon túlmutató funkcionalitás kialakításakor kritikus szempont, hogy azok a keretrendszer belső szabványainak figyelembe vételével készüljenek el. Mind a téma (design), mind az extra funkcionalitás (modulok) esetében fontos ez, ha nem ezek szerint készül el a fejlesztés, akkor a keretrendszeres oldalunk kvázi megragad azon a verzión, amin készítették, a későbbiekben azt frissíteni nem tudjuk, vagy az új verziókban is újra és újra bele kell patkolni a korábbi megoldásokat, melynek nyilván meg lesz a maga díja.

Egy keretrendszer alapú megoldásban az egyedi elemeknek a keretrendszer alapjaira kell épülniük, és nem bele azokba. Egy böngészőből nézve weboldalunkat ez sajnos nem derül ki, hogy a kivitelezők az igényes, de több munkával járó szabványos fejlesztés, vagy a rövidebb idő alatt nyélbe üthető gányolás mellett döntöttek mikor oldalunk készült.

### **Aktív jövőállóság**

Ha vállalkozásunk napi működésének online meghosszabbítása lesz a készülő weboldal, neadjisten a cég dinamikusan változó vagy bővülő igényeinek, stratégiáinak követését várjuk el tőle, akkor fontos, hogy a választott fejlesztőcégünk bármit (megismétlem, mert fontos: BÁRMIT) el tudjon készíteni, és a meglévő weboldalba integrálni, záros határidőn belül. Ennek az a garanciája, ha weboldalunk az utolsó szögig a fejlesztőcég szellemi terméke, nem pedig egy harmadik fél által fejlesztett megoldás

alkalmazása.

Jó példa erre, mikor a keresőcégek előrukkoltak a schema.org-os közös szemantikus markup megoldásukkal. Ennek a támogatása egy keretrendszerben nem triviális, a rendszer legalapvetőbb funkcióit érintő módosításokat igényel, ami nem egy jó ötlet, mert a következő kiadott frissítés leradírozza az egészet, mintha ott sem lett volna. Vagy, amennyiben a funkció fontos számunkra, le kell mondanunk a frissítőcsomag által nyújtott javításokról, ami kifejezetten problémás, ha az biztonsági javításokat is tartalmaz.

### c) **Biztonság**

A keretrendszerek népszerűségük és publikus forrásuk miatt állandó nemkívánatos figyelem középpontjában állnak, rendszeresen adnak ki hozzájuk biztonsági frissítéseket, amiket magára valamit is adó weboldal tulajdonosnak a lehető legrövidebb időn belül telepítenie kell. Az elterjedtebb rendszerek esetén évi 4-6 biztonsági frissítésre biztosan számíthatunk. Ha a keretrendszert használó fejlesztőnk erre nem hívja fel a figyelmünket, kerüljük el messziről. Győződjünk meg róla, hogy milyen áron vállalja a verziókövetést, egyáltalán vállalja-e. Ha nem vállal ilyesmit, akkor szándékosan hibás portékával kereskedik, ezért kerüljük el messziről. A modulokat akkor még nem is említettük, jó eséllyel mindegyik más fejlesztő keze alól kikerült alkotás, a saját potenciális biztonsági hibáival. minden egyes extra telepített modul tovább növeli a hibalehetőségek számát és a frissítések követéséhez és telepítéséhez szükséges időt. Célszerű ezt

komolyan venni, hatalmas pénz van a be nem foltozott biztonsági lukas keretrendszerek kiaknázásában, megállás nélkül járnak a programok a weboldalakat ismert hibákra vadászva.

Példák a legelterjedtebb támadásokra:

**Kártékony tartalom beékelése:** trójai telepítő + internet explorer + windows = spamküldő zombigépek.

**SPAM tartalom:** vadidegen oldalakra mutató linkekkel ellátott semmitmondó kommentek, esetenként teljesen pofátlanul csak ömlesztett linkek.

**Felhasználói adatok kinyerése:** a valódi email cím a legértékesebb spam célpont, a regisztrált felhasználó e-mail címe pedig jó eséllyel az.

**E-mail spam állomás:** brutálisan kritikus biztonsági rések esetén weboldalunk könnyen spamküldő gócponttá is válhat.

## Worst case scenario

Friss Wordpress weboldalunkat megspékelte a fejlesztő "A" és "B" modulokkal, kicsit igényeinkhez is igazította őket. Egy hónappal később kijön egy biztonsági frissítés "B" modulhoz, amit nem tudunk telepíteni, mert elveszítjük a kis extra funkcionalitást, amit beletett a fejlesztő. "B" modul nem annyira fontos, ezért kikapcsoljuk, hogy megszűnjön a biztonsági probléma.

Egy hónappal később kijön egy frissítés "A" modulhoz, amit szintén nem tudunk frissíteni. "A" modul fontos feladatot végez az oldalon, ezért nem tudunk mit

Cikkeink, tanulmányaink és értekezéseink célja a tájékoztatás, az eredeti szerző és forrás megjelölésével (Hostonic Online Bt, [www.hostonic.hu](http://www.hostonic.hu)) egészükben vagy a kontextust megőrző részleteikben szabadon újraközölhetőek.

tenni, hagyjuk működni és imádkozunk, hogy ne akadjon rá senki.

Egy hónappal később kijön egy Wordpress frissítés, ami olyan kritikus hibákat javít, ami miatt muszáj feltelepíteni, szomorúan elbúcsúzunk hát "A" modultól.

Egy hónappal később kijön egy új Wordpress verzió, amiben kicsit megváltozik a témakezelés, és újabb fontos biztonsági problémákat orvosol. Sajnos telepíteni nem tudjuk, mert oda a saját design, így örökre megragadtunk egy hibás verzióban.

Két hét múlva a google találati listájában oldalunk mellett óriási figyelmeztetés: "kártékony weboldal". Újabb két hét múlva tárhelyszolgáltatónk felmondja szerződésünket, mert a weboldalunk titokban fusi munkát vállalt egy spamküldő cég zombihálózatának dicspécsereként.

Természetesen nem mindenki vel történik ez meg, de potenciálisan bárkivel megeshet, aki keretrendszeres megoldást választ anélkül, hogy meggyőződne az alkalmazott megoldások jövőállóságáról.

### ***Az egyedi fejlesztésekről***

Az évek óta a piacon lévő egyedi fejlesztéseket kínáló cégek nagy valószínűség szerint kifejlesztették saját házi szerkesztőségi rendszerüket, mely az évek során egy homogén, kiforrott és valós megrendelők tüzeiben megedzett megoldássá nőtte ki magát, így kezelhetőség szempontjából nem marad mögötte a keretrendszerek csilli-villi felületeinek.

Cikkeink, tanulmányaink és értekezéseink célja a tájékoztatás, az eredeti szerző és forrás megjelölésével (Hostonic Online Bt, [www.hostonic.hu](http://www.hostonic.hu)) egészükben vagy a kontextust megőrző részleteikben szabadon újraközölhetőek.

Maga a tudás, ami egy ilyen rendszer elkészítéséhez kell, áll a rendelkezésünkre, mikor egy egyedi megoldást választunk, és a helyismeret, ami a fejlesztőnek a hazai pálya előnyét adja, mikor egyéni megoldásokat készít számunkra.

Keretrendszerek esetében ez a helyismeret általában nem áll rendelkezésre, ritkán találni olyan keretrendszeres fejlesztőt, aki eléggé elmélyedt az adott rendszerben ahhoz, hogy megüsse azt a szintet, mint akik az adott rendszert fejlesztik. Az ismeretanyag általában sajnos megáll azon a szinten, hogy a publikus forrásokból kész modulokat begyűjtenek egy-egy problémára, testreszabás címén összeraknak egy témát és ha nagyon meg kell változtatni valamit, akkor az adott keretrendszer szabályait figyelembe nem véve belepiszkálnak itt-ott, előszeretettel csemegézve a kész megoldásokat szállító online scriptgyűjteményekből. Az otromba huszárvágások és a mértéktelen gányolás melegágya sajnos ez a terep, tisztelet a kivételnek.

Természetesen maga az egyedi fejlesztés ténye nem garancia arra, hogy egy átgondolt és megfelelő színvonalon kivitelezett megoldást kapunk pénzünkért, sok otromba és elavult házi-barkács megoldás lelhető fel egyedi fejlesztés címén. Fontos tehát, hogy meggyőződjünk az egyedi fejlesztést kínáló cég szerkesztőségi rendszerének milyenségéről, kérjünk egy bemutatót vagy egy demó hozzáférést, ha lehetséges, de minimum egy leírást. Ebben a rendszerben kell az elkövetkező néhány évben dolgozni az oldalunk tartalmán, nagyon nem mindegy, hogy az admin felületen eltöltött időt élvezni, vagy átkozni fogjuk.

## ***Kivételek a keretrendszerek javára***

Az igények pontos és hosszú távon pontos ismeretében, annak tudatában, hogy a keretrendszer maradéktalanul kiszolgálja igényeinket, lehetőleg kizárólag az alap rendszer által nyújtott lehetőségekkel, vagy csak hivatalos fejlesztésű modulokkal, teljesen indokolt lehet egy keretrendszer használata. Egyedi elképzelések, vagy a jövőt illető bizonytalanság, tarsolyban tartott további ötletek esetén viszont valószínűleg zsákbamacskát vásárolunk.

## ***Vicces tévhitek***

A keretrendszer jó, mert könnyebb hozzá értő fejlesztőt találni.

Totál hamis. Ha van valami, amibe a más által fejlesztett rendszernél jobban utálnak a fejlesztők hozzányúlni, az a más által összegányolt keretrendszerek.

A keretrendszer alapú fejlesztés olcsóbb.

Hamis. A keretrendszerbe gányolás olcsóbb. A szomszéd néni kandós unokájának a szobatársa, aki nem ad számlát, sem garanciát, az olcsóbb. Ha minőségi keretrendszer alapú megoldást akarunk, ami teljesen az igényeinkhez van szabva, az pontosan olyan igényes munkát és felkészült fejlesztőt követel, mint egy egyedi megoldás, ennek pedig ára van.

Az egyedi fejlesztés rossz, mert nem fogja más folytatni később.

Cikkeink, tanulmányaink és értekezéseink célja a tájékoztatás, az eredeti szerző és forrás megjelölésével (Hostonic Online Bt, [www.hostonic.hu](http://www.hostonic.hu)) egészükben vagy a kontextust megőrző részleteikben szabadon újraközölhetőek.



Válasszunk fejlesztőt úgy, mint fodrászt, borbélyt, nőgyógyászt vagy háziórvost. Válasszunk úgy, hogy ne kelljen másnak folytatni. Győződjünk meg a múltjáról, kérdezzük meg ismerőseink tapasztalatait, és keressünk olyan céget, aki vélhetően 2-3 év múlva is a helyén lesz. Ha évente fejlesztőt váltunk, akkor újra és újra ugyanazt a munkát kell nekünk is ismételtelen elvégezni, borul a fáradságos munkával felépített online jelenlétünk, a gyűjtőoldalakon elhelyezett linkjeink elkezdnek nem létező oldalakra mutatni, pagerankünk zuhanni kezd, és fizethetünk vagyonokat ismét egy SEO szakembernek, hogy rázza gatyába az oldalunkat.

Minden fejlesztő csak az elődje munkáját szapulja.

Egy weboldal legkomplexebb eleme a szerkesztőségi rendszer, a továbbfejleszthetőség is általában ezen áll vagy bukik. Ha több munka belerágni magunkat, hogy hogyan is működik az előző rendszer, mint megoldani ugyanazt kibővítve a saját házi kedvencünkkel, akkor a logikus lépés az előző rendszer kukázása. Ez általában azzal a felkiáltással zajlik, hogy az előző egy nagy szar volt, és többnyire tényleg az is, de a legtöbbször a rendszer idegensége és az igényes módosítások készítéséhez szükséges tanulási görbe az, ami a kegyelemdőfést megadja.

## **Konklúzió**

Ha vonzó alternatívát nyújt egy keretrendszeres megoldáson alapuló ajánlat, akkor mindenképp kérjünk választ és árat a következő kérdésekre, és ezek

Cikkeink, tanulmányaink és értekezéseink célja a tájékoztatás, az eredeti szerző és forrás megjelölésével (Hostonic Online Bt, [www.hostonic.hu](http://www.hostonic.hu)) egészükben vagy a kontextust megőrző részleteikben szabadon újraközölhetőek.

alapján hasonlítsuk össze az ajánlatokat:

- A keretrendszer alap szolgáltatásain túli funkcionalitást saját fejlesztésű modulokkal oldják-e meg, vagy csak összefésülnek néhány harmadik fél által készített modult (történik-e effektíve fejlesztés).
- API level kompatibilitást vállalnak-e (frissíthető lesz-e az oldal a hivatalosan kiadott frissítésekkel), ha igen milyen szintig, vagy mennyi időre.
- Verziókövetést vállalnak-e a saját fejlesztésű/módosítású modulra és mennyiért (nem mindig rajtuk múlik persze, nagyot változhat egy keretrendszer 1-1 verzió között).
- API konform kódot gyártanak-e, vagy gányolnak (kompatibilis modulként készülnek-e el a szükséges modulok).
- Keretrendszer frissítést vállalnak-e, ha igen milyen díjon (évi 4-6 frissítéssel szinte biztos számolhatunk)
- Fenn vannak-e a vonatkozó levelezési listákon, elvégzik-e a frissítéseket maguktól, vagy valakivel külön figyeltetni kell a frissítések megjelenését.
- Érdeemes átnézni korábbi projektjeiket, hogy milyen régiiek a használatos verziók, hányas verzióval dolgoznak jelenleg és melyik a legfrissebb.
- Betörési vészforgatókönyvük és mentési stratégiájuk van-e.

Bármilyen megoldást is választunk végül, fontos, hogy legyen:

- Napi mentésünk (adatbázis és webtárhely egyaránt);
- Helyreállítási forgatókönyvünk, és emberünk, aki mindezt gyorsan el is tudja végezni.