

IT-ARCHITEKTEK A PORONDON: FEJLESZTÉSEK AZ ÜZLETI IGÉNYEK FÉNYÉBEN

Összehangolt működés az üzletfolytonosságot támogató fejlesztésekért

Az IT-fejlesztések nemcsak az informatika, hanem az üzlet és a vállalat hatékonyságát, gyors reakciókészségét és a bitekbe kódolt piaci világban való érvényesülését is meghatározzák. Épp ezért nem mindegy, hogy az adott innováció mikor mutat túl a belső fejlesztési csapaton, vagy mikor van szükség arra, hogy az IT-architektek értő kezei közé kerüljön. Egy CIO-t és egy IT-architekt vezetőt kérdeztünk a no-code, low-code dilemmáról, a kihívásokról és arról, hogy miként lehet priorizálni az üzlet igényeit fejlesztési szemszögből.

A stratégiák összhangban működve teszik flottá az üzleti működést és az üzletfolytonosságot, azonban az ezt támogató fejlesztésekig néha hosszú és szövevényes út vezet, az üzleti prioritások szűrésétől kezdve a költségvetési keret elfogadásán át a megvalósításig.

Nagy jövő vár a low-code programozásra az üzleti alkalmazások fejlesztésében

Rentabilitás mindenekelőtt

„Le kell szögezni, hogy bármilyen fejlesztés hatékonysága akkor lesz ideális, ha már az elejétől, a tervezési szakasztól kezdve bevonjuk a felhasználót és az adott szakmai területeket. Tapasztalataim szerint a fejlesztéseket nem az IT kezdeményezi első sorban, hanem az üzlet, ugyanakkor a fejlesztés minden esetben egy megvalósítási tervvel



TANKA LÁSZLÓ,
BM ORSZÁGOS KATASZTRÓFAVÉDELMI FŐIGAZGATÓSÁG

kezdődik, ahol az erőforrásokat és az időráfordítást mérlegeljük, illetve azt, hogy szükséges-e a projekt megvalósításához külsős szakember. A prioritások megvitatása az IT-vezető és az üzleti döntéshozók között zajlik, ezt követően születik meg a projektre szánt költségvetés”, mondta *Tanka László*, a BM Országos Katasztrófavédelmi Főigazgatóság CIO-ja. Cége válogatja, de a kritikus, azonnali megoldásra szoruló igényeket a pénzügyi terv terhére is megvalósítják a szervezetek, amit a rövid távon kisebb ráfordítással járó projektek követnek. A költségesebb, de nem annyira sürgető fejlesztések esetében előfordul, hogy akár egy év is lehet a kifizetési idejük.

„Gyakran előfordul, hogy nagyobb a projektigény, mint a leszállítói kapacitás, ezért mi egy úgynevezett »long listtel« kezdjük a prioritizálást. Ezen a bővített listán minden igényt felsorakoztatunk, amit az IT-üzemeltetést, az üzletet és az egyéb területeket képviselő felső vezetők megvizsgálhatnak, és kiválasztják a legfontosabbakat. Ezekkel haladunk tovább a fejlesztők-ből, üzleti döntéshozókból és architektékból álló szakmai csapattal, akik elkészítik az üzleti specifikációt, megnézik a piacon elérhető technológiát, az abból származó esetleges hasznot. Így készül el a »short list«, amely a fejlesztés alapjául szolgál”, mondta *Tóth István*, a GRÁNIT Bank IT-Architektúra vezetője.

Ami a fejlesztések delegálását illeti, a cégek leginkább a belső fejlesztő-csapat felé nyitottak, azonban ha kompetenciahiány lép fel, egy-egy alkalom erejéig külső fejlesztőt „bérelnek”.

A prioritások megvitatása az IT-vezető és az üzleti döntéshozók között zajlik, ezt követően születik meg a projektre szánt költségvetés



TÓTH ISTVÁN,
GRÁNIT BANK

FORRÁS: ITB

A szomszéd füve mindig zöldebb, avagy belső vagy külső erőforrásból fejlesszünk?

„Olyan dolgokat fejlesztünk, amelyeket házon belül, end-to-end meg tudunk csinálni, legalábbis ez a cél. Egyrésztől azért, mert így a teljes folyamatot kézben tudjuk tartani, ha pedig nehézség vagy probléma adódik, akkor azonnal képesek vagyunk reagálni és közbeavatkozni. Másrésztől pedig azért, mert általában olyan innovációkat alkalmazunk, amik nincsenek jelen dobozos megoldásként a piacon. Ha pedig az adott tudás nincs meg a belső fejlesztői vagy IT-architekt csapatban, akkor adott időre igénybe vesszük egy külsős szakember kompetenciáit”, mondta *Tóth István*. „Egyébként IT-architekt csapatunk tudása széles körű: front-end, back-end, mobilos architektúra és field architekt egyaránt megtalálható benne.”

Ugyanakkor nem minden szervezet teheti meg azt, hogy saját IT-architekt csapatot tart fenn. Ilyen esetekben a rendszerek közötti kapcsolatok átláthatóságáért általában a business analyst felel, vagy az üzleti oldal és az architektékek között álló szervezők.

„Az államigazgatásban is szükség van IT-architektre a belső fejlesztői csapatban, azonban ritkán vannak jelen valójában. Ezt a típusú kompetenciahiányt azzal küszöböljük ki, hogy a fejlesztők univerzális tudásúak”, mondta *Tanka László*.

No-code vagy low-code?

A kódolás kapcsán egybehangzó válaszok érkeztek: a két szakember egyaránt a low-code útját látja a fejlesztés egyik nagy várományosának. A no-code ellen szóló érvek közül tucatnyit hallani, azonban a leghangsúlyosabb mind közül, hogy a modulok az esetek többségében nem képesek kielégíteni, lefedni az összes szükséges funkciót, az üresen maradó hézagok kitöltése így a programozókra hárul. Ezzel szemben a low-code-dal alacsony beruházási költség mellett, gyorsan, minimális kézi kódolással lehet készíteni és bevezetni üzleti alkalmazásokat.

„Azt tervezzük, hogy a közeljövőben modernizáljuk a dolgozók számára készült belső egységes front-endünket, aminek kapcsán megvizsgáljuk a low-code platformot is”, mondta *Tóth István*, aki azt is hozzátette, hogy használnak saját publikus könyvtárakat, de csak nagyon szigorú biztonsági előírások mellett, hogy az ügyfelek adatvagyonra semmiképp ne sérülhessen. A közigazgatásban is hasonló a helyzet az állami adatvagyon tekintetében. „Ezt erősíti a létrehozott Állami Alkalmazás Fejlesztési Környezet (ÁAFK), melynek egyik kiemelt célja, hogy az egyes alkalmazások minimális konfigurációs beállítások mellett könnyen implementálhatók legyenek más állami szervezetek számára is. Mivel az alkalmazás-nyilvántartás az államigazgatásban központosított, ezért saját forráskódkönyvtárat kizárólag biztonsági célból üzemeltetünk”, mondta *Tanka László*.

Az IT-fejlesztések kapcsán a legfontosabb a technológiák és kódok alkalmazásán túl az, hogy az IT és a felhasználói igény közeledjen egymás felé, és jól definiálható követelményekkel, közösen haladjanak a projekt megvalósítása útján.

Kiss Franciska