

Modern szoftverfejlesztési folyamatok: előnyök és új technikák

Az Agile, a DevOps és a CI/CD a modern szoftverfejlesztési folyamatok alappillérei, a gyorsan változó környezetben való boldoguláshoz nélkülözhetetlen alapelvek. A különböző iparágakban való bevezetésük sikertörténetekkel járt, ami meghozta a kedvet további alkalmazásukra, 2023 végére már a szervezetek 75%-a alkalmazza ezeket a gyakorlatokat. A Deloitte szakértői azokat a módszereket ismertetik, amelyek segíthetnek az agilis tesztelési célok, a tesztelés ütemezésének korlátjai és a szigorú teljesítménykövetelmények jobb kezelésében.

A Gartner előrejelzése szerint idén már a szervezetek 75%-a alkalmazza az agilis gyakorlatokat a termékfejlesztésben, a válaszadók az Agile-t és a DevOps-ot jelölték meg a következő 1-2 évben a beruházási fókuszban lévő legfontosabb minőségfejlesztési trendként.

Ez egyértelműen jelzi az Agile és a DevOps javára történő elmozdulást az olyan minőségmérnöki gyakorlatok iránti igényel együtt, mint a shift-left megközelítések és az automatizált folyamatos tesztelés. Elengedhetetlen, hogy a szervezetek újraértékeljék Agile és DevOps folyamataikat a tesztelés hatékonyságának maximalizálása érdekében, és finomítsák automatizálási stratégiáikat a valódi folyamatos tesztelés lehetővé tétele érdekében. – mondta Bakó Béla, a Deloitte Magyarország Technológiai tanácsadás üzletágának menedzsere.

Mi a DevOps, az Agile és a CI/CD és hova fejlődhetnek?

A DevOps egyesíti a szoftverfejlesztést (Development), és az üzemeltetést (Operations), ezzel meggyorsítva és hatékonyabbá téve a munkát. A DevOps és az Agile üzleti vezetőket, fejlesztőket és IT-üzemeltetési munkatársakat egyesít többfunkciós csapatokban, így a magas szintű technológiai automatizálás és együttműködés racionalizálásával alakítja át a szoftverszállítást. A Deloitte felmérésének válaszadói számos, a DevOps által realizált előnyt soroltak fel (1. ábra), köztük első helyen a problémák gyorsabb megoldását (74%). A rövidebb piacra jutási idő (62%), a folyamatos szoftverszállítás (60%) és a stabilabb működési környezet (58%) szintén magasan végzett, és a válaszadók 99%-a jelezte, hogy használja a DevOps-ot. Nagy léptékben használva a DevOps segít a vállalatoknak abban, hogy jelentősen csökkentsék a szoftverhibák számát és gyorsabban piacra tudják dobni a termékeket. Ez fenntartható versenyelőnyhöz vezet.

20 évvel az Agilis Kiáltvány megalkotása után az agilis gyakorlatok tovább fejlődnek és terjednek a további funkcióik az iparágak között. A megvalósítási modell a szoftverfejlesztési és informatikai részlegeken túl olyan területekre is kiterjedt, mint az agilis audit, az agilis költségvetés-tervezés vagy az agilis HR. A gyártóipartól kezdve a kiskereskedelmen át az energiaiparig az ágazatok aktívan befektetnek.

A CI/CD olyan módszertan a szoftverfejlesztésben, amely a folyamatos integrációt és a folyamatos szállítást egyesíti, (Continuous Integration CI, Continuous Delivery, CD) de ma már több, mint a szoftverfejlesztés és -szállítás egyszerű megközelítése. A felhő-natív alkalmazások telepítésének szerves és elválaszthatatlan összetevőjévé vált, amelyek robusztusabbá válnak, ha fejlesztésüket CI/CD módszerekkel integrálják. Az olyan vállalatok, mint például az Amazon és a Netflix, minden szakaszban alkalmazzák a DevOpsot, a CI/CD folyamatokat és az automatizálást a kód napi több ezer alkalommal történő telepítéséhez. A hagyományos szoftverszállítási módszerek nem alkalmasak a

felhősebességű szállításra. A szervezetek ma már olyan skálázható alkalmazásokat akarnak, amelyek a konténerek, mikroszolgáltatások, szolgáltatáshálóok és virtuális gépek felhasználásával magán, nyilvános és hibrid felhőinfrastruktúrán futnak. Az Agile, a DevOps és a CI/CD számára optimalizált tehetségmodellek továbbképzése és frissítése kritikus lépés (2. ábra).

Fontos még az ún. 'pod' modell is, amely egyre nagyobb teret nyer a szervezetek körében. A podok olyan önálló, önszerveződő, különböző kompetenciákkal felszerelt csapatok, amelyek teljes mértékben magukénak érzik és irányítják a termék iteratív szállítását. Ha a podok speciális és egymást kiegészítő készségekkel rendelkező emberekből állnak, jelentősen javíthatják a döntéshozatalt, segíthetik a termékfókuszálást, valamint a termék gyorsabb építését, tesztelését és kiadását. – mondta Szász Beáta, a Deloitte technológiai tanácsadás üzletágának szenior tanácsadója.

A minőségfejlesztés és az agilitás művészete, a tesztelés legfontosabb technikái

A minőségfejlesztő csapatoknak folyamatosan kihívást jelent, hogy kevesebb idő alatt, kevesebb emberi erőforrással végezzék el a tesztelési tevékenységeket. Amikor a projekt ütemtervei összeomlanak, gyakran a tesztelési tevékenységek terjedelme és időtartama az első, ami hatással lehet a minőségre. A nagy teljesítményű, skálázható, biztonságos és funkcionális rendszerekkel kapcsolatos elvárások miatt azonban a magas minőség nem lehet kérdés. Az agilis és a DevOps gyakorlatok segíthetnek enyhíteni ezt a feszültséget, előnyeiket látva azonban nem szabad azt hinni, hogy könnyű eljutni alkalmazásukig. A Deloitte felmérésének válaszadói jelezték, hogy még mindig számos, teszteléssel kapcsolatos kihívással szembesülnek az agilis és devOps gyakorlatok bevezetése során (3. ábra). Ezek közül a szervezeti kultúrával (56%), a technikailag kompetens erőforrások szükségességével (46%), valamint a folyamatos tesztelési megközelítésekkel kapcsolatos kihívások (46%) a leggyakoribbak.

Számos olyan technika létezik, amely segíthet az agilis tesztelési célok, a tesztelés ütemezésének korlátjai, és a szigorú teljesítménykövetelmények jobb kezelésében.

A sprint közbeni automatizálás az agilis szoftverfejlesztési folyamat szerves része, amely segíthet a rövid fejlesztési ciklusok, illetve a követelmények gyakori változásai esetén, valamint akkor, ha a tesztek kis időablakon belüli megtervezése és végrehajtása szükséges. A cél itt az automatizálási tesztszkriptek alkalmazásfejlesztéssel párhuzamosan, ugyanazon sprinten belül történő fejlesztése és végrehajtása. A sprint befejezésekor az automatizált tesztek a regresszióteszt-készlet részévé válnak.

A TestOps („shift-right”) módszertan az alkalmazás életciklusának kiadás előtti és utáni szakaszaiban történő nagyobb arányú tesztelést (azaz a gyártás közbeni tesztelést) támogatja. A lényege, hogy a vállalkozások elfogadható minőségi szinteket határoznak meg a gyorsabb piacra jutás érdekében, majd gyors javítási módszereket alkalmaznak a hibák kezelésére vagy a változások visszaállítására, amennyiben problémákat észlelnek.

A tesztvezérelt fejlesztés (TDD) a fejlesztés evolúciós megközelítése, amely a tényleges szoftver fejlesztése előtt egységtesztek írását foglalja magában. Ez a megközelítés segíti a fejlesztőket abban, hogy a kódot úgy írják át, hogy az megfeleljen a teszteknek, ami minimális fejlesztési erőfeszítéssel jár, amely elsősorban a tesztkövetelmények teljesítésére összpontosít. Mivel a tesztek fejlesztése előre történik, a regressziós tesztelésre szánt idő is jelentősen csökken.

A *viselkedésvezérelt fejlesztés (BDD)* ösztönzi a projektben érdekelt személyek közötti előzetes kommunikációt, így a fejlesztési folyamat előtt minden tag megérti az egyes funkciókat. A BDD-ben a tesztelők, a fejlesztők és a terméktulajdonosok "forgatókönyveket" hoznak létre, amelyek megkönnyítik a példákra összpontosító kommunikációt. A BDD elősegíti a szoros együttműködést az üzleti és a műszaki csapatok között, és javítja a projekt előrehaladásának átláthatóságát. A BDD használatával az érdekeltek kiküszöbölhetik a kétértelműségeket a termékfejlesztés teljes életciklusa során, kezdve a felhasználói történetek létrehozásától a szoftver sikeres tesztelésén és telepítésén át.

Az *A/B tesztelés (más néven osztott tesztelés)* során a weboldal látogatóinak különböző szegmensei számára egyszerre mutatják meg ugyanazon weboldal vagy mobil képernyő két változatát, majd összehasonlítják, hogy melyik változat eredményez több konverziót. Az A/B tesztelés során általában az a változat nyer, amelyik nagyobb konverziót eredményez, és ez a változat segíthet az alkalmazások optimalizálásában a jobb eredmények érdekében. Az A/B tesztelés lehetővé teszi a korai visszajelzést, hogy a végfelhasználók által preferált, minimálisan életképes termékeket tudjunk létrehozni. Ez a tesztelési módszer lehetővé teszi a szervezetek számára, hogy hipotéziseket állítsanak fel, majd menet közben tanuljanak és finomítsák bizonyos felhasználói élményelemekre vonatkozó telepítési stratégiájukat.

Az *API-tesztelés* segít az API-k funkcionalitásának, megbízhatóságának, teljesítményének és biztonságának validálásában. Ahelyett, hogy a szabványos felhasználói bemenetekre és kimenetekre támaszkodna az alkalmazás logikájának teszteléséhez, az API-tesztek az üzenetrétegen, grafikus felhasználói felület (GUI) használata nélkül történnek. Ez segít az alkalmazás logikájának validálásában olyan esetekben, ahol a gyakori változások vagy a rövid kiadási ciklusok miatt nem lehetséges a GUI-tesztelés. Az API-tesztelés a tesztelés számos aspektusát lefedi, beleértve a funkcionális tesztelést, a biztonsági tesztelést és a terheléses tesztelést.

A „*dogfooding*” egy termék belső bétatesztelése a termék bevezetése előtti iteráció és finomítás érdekében. A módszer lényege, hogy a termék létrehozói és a vállalat alkalmazottai a felhasználók bőrébe bújnak, hogy első kézből tapasztalják meg azokat a problémákat vagy fájdalmas pontokat, amelyekkel az ügyfelek a való életben szembesülnének. Ez a technika javítja a minőséget, és segít megelőzni az idő előtt kiadott termékre adott kedvezőtlen piaci reakciókat. A szoftverfejlesztésben, a gyártásban és más iparágakban részt vevő szervezeteknél ez mára bevett gyakorlattá vált. Segít növelni az általános UX-tudatosságot a szervezeten belül, és nagy költségeket takaríthat meg (pl. lehetséges garanciák).

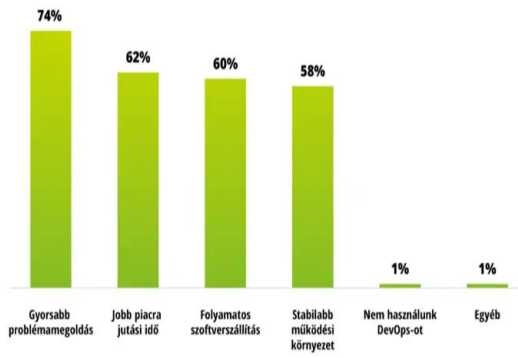
Az agilis, a DevOps és a CI/CD módszertanok elősegítik a fejlesztési, tesztelési és üzleti csapatok közötti együttműködést, felgyorsítják a szoftverfejlesztést, javítják a termékminőséget és automatizálják a szoftverszállítási folyamatot. Mindez segíti a szervezeteket abban, hogy jobban reagáljanak a gyorsan változó piaci körülményekre, versenyelőnyre tegyenek szert, és növekedést érjenek el. – foglalta össze Bakó Béla, a Deloitte Magyarország Technológiai tanácsadás üzletágának menedzsere.

Sajtókapcsolat:

- Bakó Béla, menedzser
- bebako@deloittece.com

1. ÁBRA

A DevOps bevezetésével elért előnyök a szervezetek számára

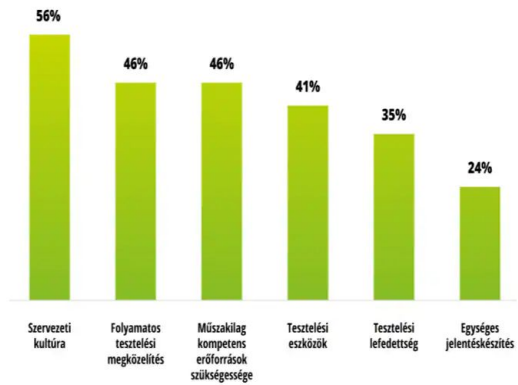


Megjegyzés: A válaszadók több választ is megjelölhettek.

© Deloitte Magyarország

3. ÁBRA

Tesztelési kihívások, amelyekkel a szervezetek szembesültek a Dev Ops használata során



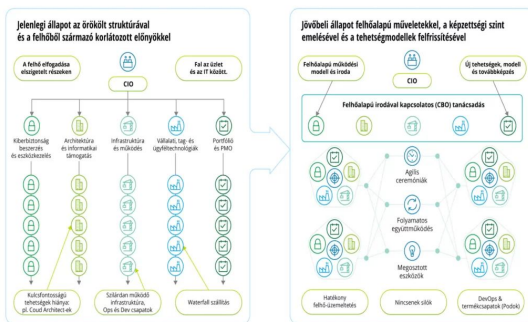
Megjegyzés: A válaszadók több választ is megjelölhettek.

© Deloitte Magyarország

© Deloitte Magyarország

2. ÁBRA

Az öröklött rendszerrel való átérés a felhő-képes szervezetre



Forrás: Deloitte elemzés

Eredeti tartalom: Deloitte Magyarország

Továbbította: Helló Sajtó! Üzleti Sajtószolgálat

Ez a sajtóközlemény a következő linken érhető el:

<https://hellosajto.hu/4501/modern-szoftverfejlesztési-folyamatok-elonyok-es-uj-technikak/>