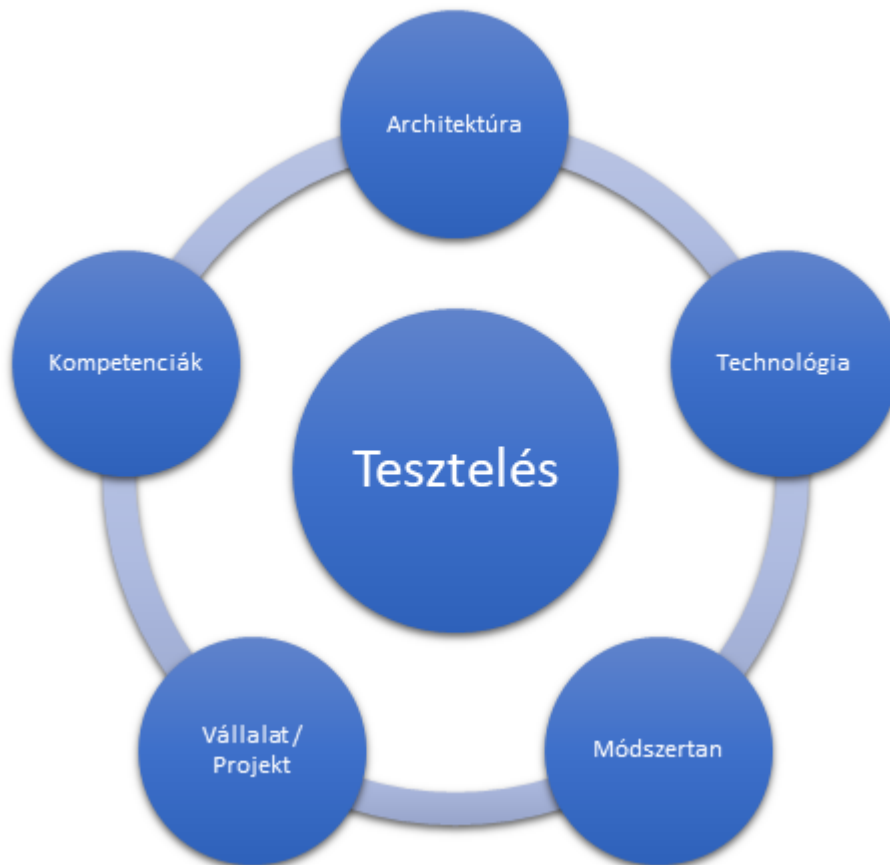


## Privát felhőre áttérés hatása a tesztelésre

A cikk azt kívánja bemutatni, hogy a privát felhő, azon belül is a konténerizációs platformok kialakítása és kapcsolódó alkalmazások fejlesztése saját tapasztalataim alapján hogyan hat a tesztelésre.

Előbb azonban szeretném kontextusba helyezni a témát. Meglátásom szerint a szoftver tesztelésnek nagyon adaptívnak kell lennie. A tesztelésnek képesnek kell lennie alkalmazkodni a megváltozó technológiákhoz, módszertanokhoz és egyéb tényezőkhöz. Erre utal a tesztelés 6. alapelve is, mely szerint a tesztelés függ a körülményektől.

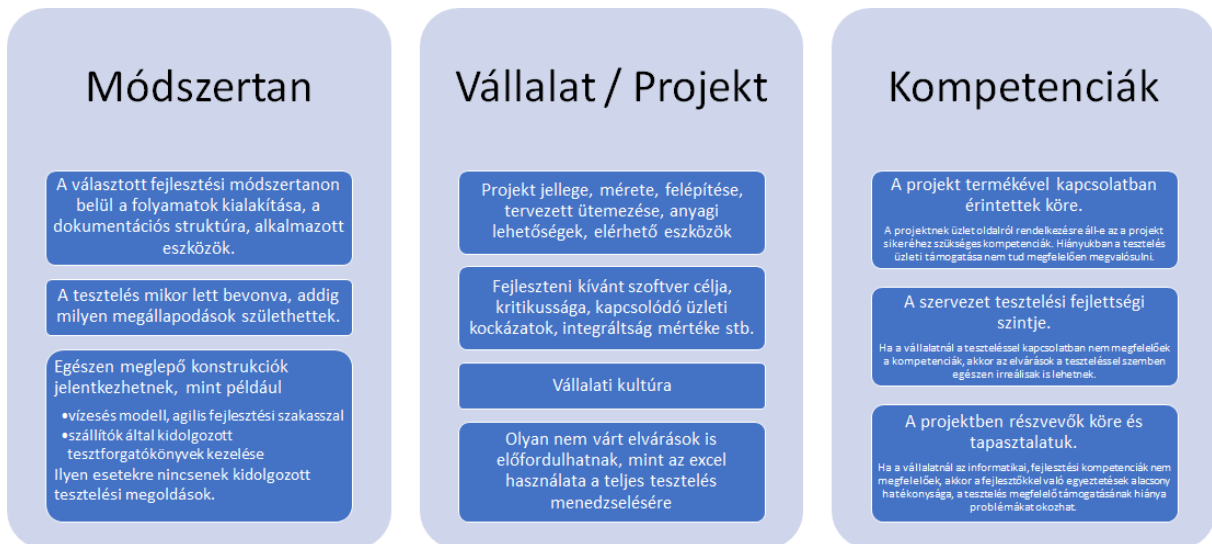
Mik azok a fő körülmények, amelyektől függ a tesztelés? Önkényesen az alábbi ábrán látható öt tényezőt jelöltem ki, mint szerintem legfontosabb tényezőket. Mielőtt a tesztelés tervezésének nekiállnék, az alább kifejtett tényezőkhöz hasonló listán megyek végig és mindegyik témát a lehető legrészletesebben tisztázom. Az összegyűjtött információkat és azok hatását a tesztelésre (pl. lehetőségeket, kötelező elemeket, korlátokat) a tesztelési stratégiában foglalom össze.



1. ábra A tesztelésre ható legfőbb tényezők.

Természetesen a fenti ábrán a kompetenciák és a módszertan tekinthető akár a Vállalat és/vagy a projekt részének is, de fontosságuk miatt kiemeltem őket.

Nézzünk néhány példát erre a három kategóriára.



2. ábra Példák a Vállalat/Projekt, a módszertanok és a kompetenciák jellemzőire, melyek hatással vannak a tesztelésre.

A módszertan esetén egyértelműen a választott fejlesztési módszertan a legfontosabb, hiszen annak megfelelő tesztelési módszertant kell alkalmazni. Továbbá fontos a projekt módszertan is, hiszen további kereteket adhat a tesztelés számára.

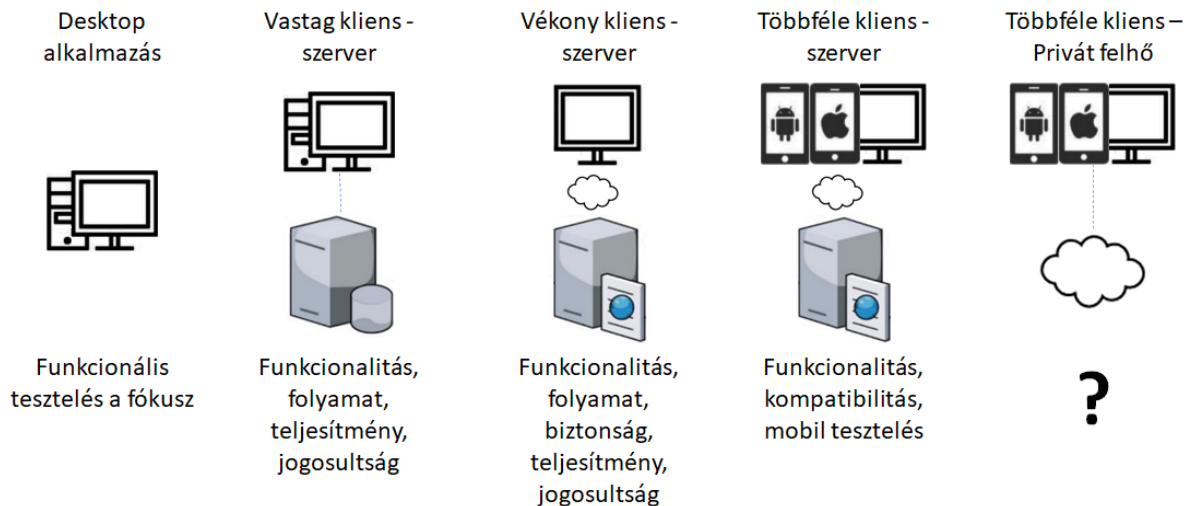
Fontos, hogy a tesztelés időben kerüljön bevonásra már az előkészítő fázisban, hogy az ott készülő anyagok, szerződések tervezetnek a tesztelés számára is véleményezhetőek legyenek.

A projekt jellege (pl. új szoftver fejlesztése, meglévő szoftver továbbfejlesztése, szoftver bevezetése), a feladat mérete, az ütemezés és még számos tényező befolyásolja, hogy milyen tesztelés típusokat kell elvégezni, mi a szükséges erőforrások száma, típusa, milyen tesztelési folyamatok indokoltak, milyen tesztelési riportokra lesz szükség.

A szoftver célja, kritikussága jelentősen befolyásolja, hogy milyen alapos tesztelésnek kell alávetni a rendszert, ami kihat a módszerekre és az erőforrásokra, de a kár a dokumentáltságra is.

A kompetenciák megléte mind üzleti, tesztelési és informatikai oldalon fontos. Annak hiányában arra kell készülni, hogy a scope sokkal többet fog mozogni, a döntéshozatal időigényes lesz és a tesztelés támogatása sem lesz megfelelő. Így ez az időbecsléseknél mindenképpen figyelembe kell venni.

Az architektúrák és technológiák hatását a tesztelésre az alábbi egyszerű ábrával szeretném érzékeltetni.



### 3. ábra Példák az architektúrák és technológiák jellemzőire, melyek hatással vannak a tesztelésre

Kezdetben voltak a desktop alkalmazások. Ezek egy-egy számítógépen lokálisan működő programok, melyek mind a mai napig megtalálhatóak. A tesztelés legfőbb fókusza a funkcionális tesztelés volt. Természetesen kisebb mértékben megtalálhatóak voltak a tesztelés egyéb elemei is.

Ezt követően megjelentek a központi adatbázissal jellemzően belső hálózaton működő, vastag klienses alkalmazások és a tesztelésnek alkalmazkodnia kellett. Ekkor már megjelentek a funkcionális tesztekben is új elemek, mint a pl. a konkurens használat tesztelése, de erősödtek olyan elemek is, mint a folyamat tesztek, performancia tesztek, jogosultság tesztek.

Az internet és a webes technológiák fejlődése lehetővé tette, hogy böngészőkben futó alkalmazások jelenjenek meg. Ezek az alkalmazások már sokkal jobban kitétek voltak a hackerek támadásainak, így előtérbe került a biztonság kérdése. Az általam vezetett tesztelésekben ma már standard, hogy a felületi validációkat, felületen jogosultság miatt el nem érhető funkciókat, adatköröket közvetlen service hívással próbáljuk meg a szerverrel végrehajtani meggyőződve arról, hogy szerver oldalon is léteznek ugyanazok a validációk, mint kliens oldalon és minden válaszüzenetet ellenőrzünk, hogy abban nincs-e felesleges, a felhasználóra nem tartozó adat.

A telefonok fejlődése és a mobil sávszélességek növekedése miatt megjelent a mobil technológia mely szintén nagy hatást gyakorolt a tesztelésre. Funkcionalitásban az eltérő felületi működések (pl. nincs hover over) az esetek többféle kidolgozása lett szükséges, a sokféle méret és felbontás miatt a rezponzivitás tesztelése fontosabbá vált, a sokféle készüléktípus és operációs rendszer miatt a kompatibilitás tesztelése is nagyobb hangsúlyt kapott, a gps, a giroszkóp és a kamera miatt megjelenő újfajta alkalmazások pedig új jellegű tesztelési feladatokat jelentettek.

Bár az ábrán nem került megjelölésre, de mindenképpen érdemes megemlíteni a vékony kliensek „vastagodását”, ahogy az újabb html szabványok és a böngészők, fejlesztési keretrendszerek fejlődése egyre több funkció kliens oldalra történő áthelyezését teszi lehetővé.

Az internet sebességének további növekedése és megbízhatósága, az üzembiztonság igénye és a költségmegtakarítás igénye miatt megjelentek a felhőszolgáltatók, melyek lehetővé teszik az adatbázisok és/vagy szolgáltatások felhőbe költöztetését. Vannak azonban vállalatok, amelyek valamilyen okból nem szeretnék adataikat és rendszereiket a szervereiken kívül látni, de nekik is megvan az igényük a felhő nyújtotta előnyökre. Azonban itt nem lehet alkalmazni a klasszikus előnyöket felsoroló listákat, mint pl. azt, hogy nem igényel beruházást. Éppen ellenkezőleg, mivel a

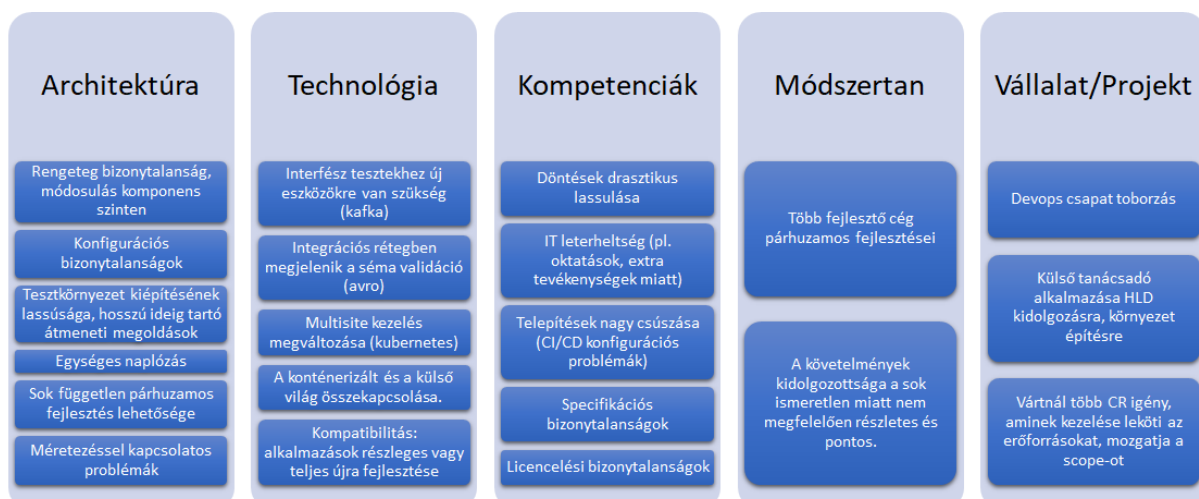
vállalatnak ilyenkor magának kell biztosítania az infrastruktúrát, a hardware és licence költségek, az egyedi tervezés és kivitelezésbe bevont tanácsadók, az IT terület képzése, új munkatársak toborzása, a meglévő szoftverek átalakítása vagy platform kompatibilis technológiával való újra fejleszttetése mind nagy befektetéseket igényelnek.



4. ábra A privát felhőre jellemző és nem jellemző előnyök összefoglalása

A privát felhő megoldások közül az alábbiakban kifejezetten a konténerizációs platformokkal foglalkozom. Azt gyűjtöttem össze, hogy milyen tesztelésre ható tényezők jellemezték azt a projektet, mely párhuzamosan célul tűzte ki a platform bevezetését, ahhoz kapcsolódó CI/CD folyamat kialakítását és a platformra modulok fejlesztését.

Az alábbi ábra összefoglalja az öt fő tesztelésre ható terület szerint csoportosítva azokat a tényezőket, amelyek leginkább érintették a tesztelést.



5. ábra A konténer alapú privát felhő bevezetési folyamat során a tesztelést ért hatások

Mivel az architektúra párhuzamosan fejlődött, ezért az alkalmazás oldalról érkező igények és egyéb előre nem látott tényezők miatt a változások és bizonytalanságok, a döntéshozatal lassúsága sok esetben hátráltatták a specifikációk kidolgozását és a tesztelés felkészülését.

A microservice architektúra a monolit webapplikációkkal szemben lehetőséget nyújt arra, hogy egy rendszert egyszerre több fejlesztőcég is fejlesszen. Ilyen helyzetben elő kell írni a stratégiában, hogy a teszterv azonosítsa be azokat a pontokat, amelyek esetén szükséges fejlesztőcégenként elágazásokat létrehozni a folyamatokban, módszerekben.

Az egységes naplózás egy pozitív hatása az egységes platformnak, melynek segítségével a tesztelőknek csak egy alkalmazást használva elemezhetik a naplóban rögzített bejegyzéseket.

Technológia szempontjából a tesztelésnek új feladatai keletkeztek és új eszközökre is szüksége van a tesztelés feladatok elvégzéséhez, a tesztelés alatt a rendszer monitorozásához.

Az új technológia velejárója, hogy számos régi alkalmazás nem vihető át a platformra, mert például nincsenek felkészítve a több példányban való futtatásra, vagy olyan megoldásokat tartalmaznak, amelyek nem kompatibilisek (pl. websocket) és/vagy IT biztonság által nem támogatottak. Az ilyen alkalmazásokat vagy át kell írni vagy teljesen újra fejleszteni.

Kompetencia terén egy ilyen változásnál jellemző, hogy mindenki leterhelt, hiszen az új technológia számos extra feladatot ad, és a szükséges oktatásokon is részt kell venni. Ezért a tesztelés támogatása IT oldalról így nem tud olyan szinten megvalósulni, ahogy azt a tesztelők szeretnék. A tesztkörnyezetek késnek, a verziótelepítésekkel számos probléma adódik, így a tesztelések csúsznak. A technológia ismeretének hiányában a specifikációkban is számos kérdés, bizonytalanság található, melyek mind nehezítik a tesztelés felkészülését.

Módszertan szempontjából a több fejlesztő céggel való együttműködés a legkiemelkedőbb. Minden cégnek mások a módszerei, az eszközei, a kommunikációja, a folyamatai és néhány esetben a fogalmai is. A tesztelésnek több cég esetén mindezekhez mind alkalmazkodnia kell és sok esetben többféle folyamatot, módszert fenntartani.

A projekt erőforrásait és így a tesztelést is leterhelte az a nagy számú változáskezelési igény kezelése, amely a technológia, a scope és az üzleti specifikációk bizonytalanságából adódott.

Számos egyéb hatása lehet privát felhő alapú fejlesztéseknek, melyek közül alábbiakban felsorolom az általam beazonosítottakat.

- A számos új technológia, komponens és fogalom a teszteléssel szemben is komoly tanulási elvárást támaszt, hogy a tesztelés azok specifikumainak megfelelően legyen megtervezve. Ennek erőforrás igényét mindenképpen jelezni kell a tesztelési tervben.
- A platform összetettsége és konfigurálhatósága miatt előtérbe kerülnek a konfigurációs tesztelések. A teszteknek fokozottabban ki kell térnie arra, hogy a platform egyes konfigurációs paramétereinek megváltozása miként hat a futtatott szoftverek működésére, illetve fontos a specifikációkban rögzített konfigurációs paraméterek beállítását ellenőrizni.
- A könnyebb menedzselhetőség miatt lehetőség nyílik több tesztkörnyezet kialakítására, amely jelentős lehetőséget biztosít tesztkoordináció oldalon párhuzamos tesztelések lebonyolítására.
- A Kafka kommunikáció aszinkronitása miatt a megoldások eltérnek a webservice hívásoknál megszokottól, mely a tesztelőknek is új feladatokat generál és új eszközöket kell alkalmaznia (pl. Konduktor).

- Az architektúra változásával párhuzamosan a vállalat megnövekedett elvárásai (pl. 99,99% rendelkezésre állás, az alkalmazások szolgáltatás kiesés mentes frissítése) az üzemeltetési tesztekben további változásokat hoznak.
- A microservice architektúra miatt az egyes részek sérülékenységvizsgálata egymástól függetlenül nagyrészt elvégezhetőek.
- Microservicek viszonylagos függetlensége fejlesztésben, verzióadásban, hibajavításokban egyrészt elvárást támaszt a teszteléssel szemben, hogy ezeket az eltérő ütemezéseket kezelni tudja, másrésztől lehetőséget teremt arra, hogy a munka ütemezésében nagyobb mozgástere legyen.

*Szerző: Szakály Béla - Passed Informatikai Kft.*