# Diszruptív technológiák, üzleti modellek

1995-ben, a populáció-genetikából kölcsönött fogalom. Olyan innováció, amely során kifejlesztett eszköz vagy **szolgáltatás jelentős mértékben átrendezi** a piaci erőviszonyokat, a sorrendet.

**A strtatégi lényege, hogy nem a piacvezető, sikeres vetélytárs termékvonalát fejlesztjük tovább, hanem egy gyökeresen másik fejlődési vonalat indítunk el.**

A szelekciós nyomás hatására vagy kiszelektálódik, marginalizálódik az új feltörekvő, vagy lenyomja az addigi piacvezető vetélytársát.

Példák

1. Mobil telefon vs. vezetékes telefon
2. Hagyományos (szoftver) fejlesztési ökoszisztéma szereplői (SVN szerver vs. Git)
3. Hagyományos rendszermenedzsment eszközök

A stratégia még egy cégen belüli ember-ember vetélkedésben is zajlanak. A régi öreg tudásbázisát utolérni nem tudom, mert esélytelen, akkor hozok egy gyökeresen újat, diszruptív módon és beelőzöm ezzel.

*„Az agilitásra való igény és szándék azonban önmagában nem elég. Az agilitást nem lehet csak úgy bevezetni. Agilitás akkor van egy szervezetben, ha szabad teret biztosítunk a működéséhez. Ehhez le kell bontanunk az útjában álló akadályokat, megfelelő környezetet, infrastruktúrát kell biztosítsunk az agilis működéshez.”*

Másrészt, egyáltalán nem biztos, hogy egy módszertan MINDEN környezetben optimális.

Egy banki környezet IT fejlesztése NEM összemérhető egy általános szoftver terméket kifejelesztő cég környezetével.

Vajon egy atomerőmű vezérlő szoftvereit miért nem agilis módszertannal fejlesztik?

Ha egy könyvelő program egy funkciója nem tökéletes, az nem ugyan az, mintha napokig nem tud valaki a pénzéhez hozzáférni. (3 napig nem megy az IT, akkor egy bank összeomlik.)

Kockázat/haszon arányok lényegesen mások.

# DevelopmentOperation

Lényegében egy olyan fejlesztés-módszertan, amely a hagyományos ITIL (Information Technology Infrastructure Library) és a hozzá kötődő metódusokat kívánja újra cserélni.

Az ITIL-t 5 nagy terület definiálja, ezek a következők:

|  |  |
| --- | --- |
| Serviec Strategy | Új szolgáltatások tervezésének, kivitelezésének és beüzemelésének dokumentációja. |
| Service Design | Az előbbi stratégia alapján készülő projektterv amely a bevezetésről, a bevezetés támogatásáról és az üzembiztonségról szól. |
| Service Transition | Ez a változásmenedzsment, a verziókezelés folyamatát, azok szabályait rögzítő doksi. Szigorú release és deplymenedzsment. |
| Service Operation | Adott SLA-nek megfelelő üzemeltetés-biztonsági szint és rendelkezésre-állás követelményeit kielégítő, üzemeltetési rend kialakítása. |
| Continual Service Improvement | Üzemeltetés-minőség folyamatos emelése, monitoring, reporting. Montauk, etc. |

Itt egy szoftver csak szigorúan tesztelt módon, több szűrőn átmenve kerülhet production kategóriába. Ez nyilván idő és energia.

A rendszeradmin és a fejlesztői szerep- és felelősségi körök szigoruan szétvállnak. Az admin nem feltétlenül ismeri az üzemeltetett szoftver funkcionalitásait, más a dolga, és hasonlóan, a fejlesztő dolga a fejelsztés maga, és sok esetben nem ismeri a majdani futtató környezet paramétereit, kapacitásait. (A saját dev környezetén lefejleszti, ahol fut …, és innentől nem nagyon érdekli a dolog.).

SaaS Software as a Service  
IaaS Infrastructure as a Service  
PaaS Platform as a Service 🡪 OpenShift!!

A fejlesztési paradigma a vízesés-modell, ami, egy iteratív, de szigorúan szabályozott folyamat.

**Új paradigmák**

**agilis módszertan**  
futamokban tervezett, felelősségmegosztáson (felhasználó-tervező) alapuló ejlesztés-módszertan

**lean opitmalizáció**   
(értéket nem teremtő komponensek eltávolítása) Ez az IT-ben az adott szoftver felesleges funkcionalitásainak ki nem fejlesztését jelenti. Pl. a word menüpontjainak 10%-át használja a felhasználók 90%-a.

**Gyors build-elhetőség**

**Gyors deployment**, akár többszáz szerverre (virtualiáció, dockerizáció)

**A fejlesztési, rendszerbeállítási, üzemeltetési folyamatok automatizálásának az igénye**

DevOps feladatok

|  |  |
| --- | --- |
| CI Continous Integration | Folyamatos, bevezetés, integrálása a változásoknak. (GitLab, de ilyen az SVN-integráció is.) |
| Build | Maven, ANT, rendszer-könyvtárak és függőségek biztosítása |

**Nincs tesztelés igazándiból, csak kipróbálás.**