

---

HOLTAI ANDRÁS EZEDEDES –  
DR. MAGYAR SÁNDOR ALEZEDEDES –  
PUSKÁS BÉLA ALEZEDEDES

## **AZ INFORMATIKAI FEJLESZTÉS ÉS ÜZEMELTETÉS HATÁRVONALAI**

---

Az utóbbi évtizedek egyik leggyorsabban fejlődő technológiai ágazata az informatika, beleértve a hardver- és a szoftvergyártókat, valamint az alkalmazói rendszereket egyaránt. Ez a dinamikus változás jelentős hatással van mind a versenyszféra, mind az államigazgatási szektor informatikai rendszereire is. Az üzemeltetés közép- és hosszú távon nem tud állandó maradni, mivel követnie kell a folyamatosan megjelenő új informatikai technológiákat és az azokra épülő hardveres és szoftveres megoldásokat.

Az informatikai rendszerek tekintetében a fejlesztést és az üzemeltetést – hagyományosan – egymásra hatással lévő, de jól elkülönített folyamatokként kell kezelnünk. Kérdés azonban, hogy egy ilyen dinamikusan fejlődő területen van-e értelme hagyományos, jól bevált módszerekről beszélni? A fejlesztői tevékenységnek számos területe van. A kisebb szervezetek életében gyakran az üzemeltetéssel együtt közös feladatként kezelik, melyet ugyanazon kiszolgálói személyzet lát el, sok esetben ugyanazon az IT-rendszeren. De bonyolultabb, nagyobb informatikai rendszerek esetében – ahol a jogszabályi és a tulajdonosi előírásoknak is meg kell felelni – már nem lehet egyben kezelni a két területet.

Az informatikai rendszerek fejlesztése és üzemeltetése különböző szintű szakmai tudást és tapasztalatot igényel az informatikai kiszolgáló személyzet részéről. Mindkét területnek megvannak a sajátos elvárásai, továbbá mindkét terület rendkívül fontos a rendszer szempontjából. Míg a fejlesztők esetében a legfontosabb a nyitottság az új technológiákra, addig az üzemeltetőknél a felhalmozott és rendszerezett üzemeltetési tapasztalatok a leginkább meghatározók.

Mindkét eltérő terület esetén sajátos okmányrendszernek szükséges támogatni a tevékenységeket. A fejlesztést – vagyis a modernizálást, az új képesség teremtését – jelentős tervezéssel, kockázatelemzéssel kell támogatni. Az üzemeltetés során pedig nélkülözhetetlen a rendszer állapotának, konfigurációjának a pontos ismerete, valamint a változáskövetés.

Fontos, hogy míg a két területnek szoros együttműködésben kell lennie, az okmányrendszer mellett az erőforrások is különböznek. Ilyen például a szabályzó környezet, a hardver- és a szoftverkörnyezet, a humán erőforrás, de ilyen a pénzügyi erőforrás is. Így például egy fejlesztői szoftvert nem érdemes összehasonlítani egy rendszerfelügyeleti megoldással.

A publikációban a fejlesztés és az üzemeltetés fogalomtisztázásán túl a nemzetközi szabványok, ajánlások iránymutatásai alapján definiáljuk a határterületeket, az elhelyezkedést egy szervezeten belül, valamint a működési feltételeket, melyek a hatékonyabb szervezeti működést és az üzleti célok elérését segítik elő.

### Fejlesztés

Az informatikai rendszerek létrehozása alapvetően három fázisra bontható. A **definíciós** fázis – melyet jelen tanulmány nem taglal részletesebben – a rendszer funkcióit rögzíti, a megrendelői vagy a felhasználói igények alapján. A **fejlesztési** fázis magában foglalja a tervezést, a megvalósítást, az ellenőrzést, a tesztelést és a dokumentálást. A **támogatási** fázis a megrendelőnek történő átadással kezdődik, karbantartással és továbbfejlesztéssel folytatódik, majd a termék kivezetésével zárul.<sup>1</sup>

Tovább bontva az informatikai rendszerek életciklusának három szakaszát, a működés során hét fázist definiálunk:

- tervezési fázis;
- beállási fázis;
- normál üzemállapot;
- készütségi állapot;
- katasztrófaállapot;
- visszaállási állapot;
- a rendszer kivezetési fázisa.<sup>2</sup>

A tervezési és a beállási fázis egyértelműen fejlesztői hatáskör, az üzemeltetésnek azonban mindegyik állapotot figyelemmel kell kísérnie, hogy elegendő információ és tapasztalat álljon rendelkezésre a szolgáltatás teljes életciklusa során.

A fejlesztés elsődleges célja az innováció, új ötletek és megoldások eljuttatása a felhasználókhoz, a lehető legrövidebb idő alatt. Ezzel ellentmond az üzemeltetőknek az az igénye, hogy a felhasználók birtokába stabil, megbízható és gyors megoldások kerüljenek.

### Üzemeltetés

Az üzemeltetés szükséges vagy végrehajtott cselekmények sorozata, amelyek célja, hogy egy eszköz vagy rendszer eredeti állapotát konzerválják, vagy a lehető leghosszabb ideig fenntartsák – az ésszerűség és a költséghatékonyság szempontjait szem előtt tartva.

---

<sup>1</sup> OBUCH László: Informatikai rendszerek tervezése.  
[http://aktivitas-tiszk.hu/elearning/Obuch\\_Laszlo/Tervezesi\\_alapismetek.pdf](http://aktivitas-tiszk.hu/elearning/Obuch_Laszlo/Tervezesi_alapismetek.pdf); letöltés: 2016.02.03.

<sup>2</sup> HOLTAI András –MAGYAR Sándor – PUSKÁS Béla: Az informatikai üzemeltetés általános kérdései. In: Felderítő Szemle, 2005. november. 14. évf. 4. sz.  
<http://knbsz.gov.hu/hu/letoltes/fsz/2015-4.pdf>; letöltés: 2016.02.22.

Hardverelemek tekintetében az üzemeltetés szigorúan véve tesztelést, állapotellenőrzést és tisztán tartást jelent. Szoftverek tekintetében elsősorban a frissítést értjük üzemeltetés alatt, történjen az hibajavítás vagy új funkciók hozzáadása miatt. A gyorsan változó informatikai eszközrendszer jellegzetes üzemeltetési feladata a meglévő szoftverek telepítése új vagy kibővített hardverkörnyezetben. Információs rendszerek esetében ez a feladatkör kibővül például a felhasználói fiókok kezelésével, az adathordozók és a fájlrendszerek karbantartásával. Egyre kevésbé elhanyagolható a biztonság fokozásának szerepe a folyamatos karbantartással.

Ez utóbbi abból a szempontból is kiemelkedő jelentőségű, hogy Magyarország Nemzeti Biztonsági Stratégiája is kimondja: „Egyre sürgetőbb és összetettebb kihívásokkal kell számolnunk az informatikai és a telekommunikációs hálózatok, valamint a kapcsolódó kritikus infrastruktúra fizikai és virtuális terében. Fokozott veszélyt jelent, hogy a tudományos és technológiai fejlődés szinte mindenki számára elérhetővé vált eredményeit egyes államok vagy nem állami – akár terrorista – csoportok arra használhatják, hogy megzavarják az információs és kommunikációs rendszerek, kormányzati gerinchálózatok rendeltetésszerű működését.”<sup>3</sup>

Összegezve, az üzemeltetés olyan műveletek sorozata, amelyek megőrzik vagy helyreállítják egy informatikai eszköz vagy rendszer rendeltetésszerű funkcióit azok hasznos élettartamának végéig.

Az elmúlt évtizedek gyakorlatát vizsgálva számos üzemeltetési/karbantartási stratégiát definiálhatunk. Hibaelhárító (Failure Based Corrective Maintenance, FBCM) karbantartás esetén a rendszer minden esetben a meghibásodásig üzemel, így a javítás előre nem tervezhető, és túl hosszú időt vehet igénybe. A tervszerű megelőző karbantartás (Preventive/Planned Maintenance, PM) számol a meghibásodások várható idejével, ezáltal növeli a tervezhetőséget, de rugalmatlansága miatt a költségeket is. Az elvégzett munka szerinti fenntartás (Parameter Condition Based Maintenance, PCBM) nem előre rögzített, hanem az adott rendszerelemek egyedi, jellemző paraméterei (pl. üzemidő, futásteljesítmény, felhasznált anyagok stb.) alapján határozza meg a karbantartási periódusokat. Továbbfejlesztett változata az állapotfüggő karbantartási stratégia (Condition Based Maintenance, CBM), ahol a rendszerek bizonyos paramétereit időszakosan vagy folyamatosan mérik, a rögzített értékekből pedig következtetéseket vonnak le a meghibásodások várható bekövetkeztének időpontjára.

A megelőző vagy proaktív karbantartás (Proactive Maintenance) már nem csupán a meglévő vagy a várható meghibásodások elhárításával számol: célja a hibák eredendő okának (root cause) feltárása, a tervezési hibák kijavítása. Az egyre bonyolultabb rendszerek, kritikus infrastruktúrák megjelenésével a megbízhatóság a középpontba került. A megbízhatóságközpontú karbantartási stratégia (Reliability Centered Maintenance, RCM) a korábbi stratégiák egyfajta keverékeként, az állapotjellemző paraméterek és a várható karbantartási periódusok alapján jellemzi a rendszer megbízhatóságát. A tömegtermelésre szakosodott vállalatoknál jelent meg

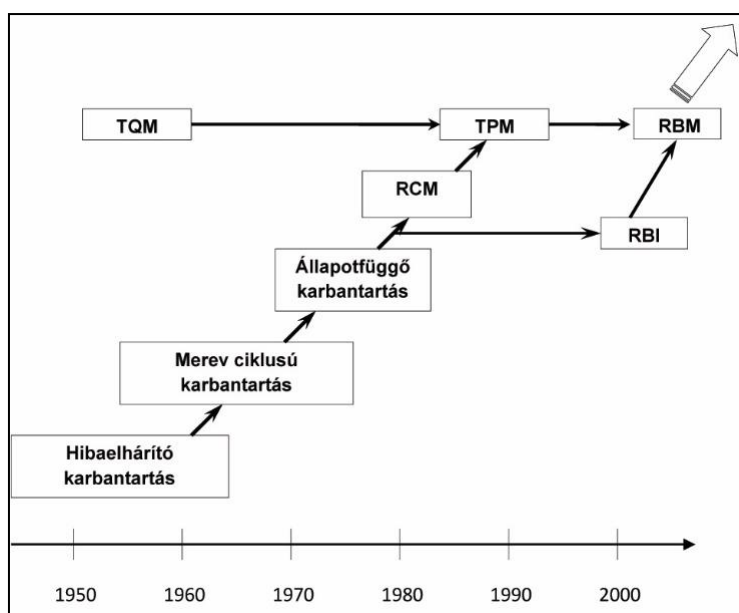
---

<sup>3</sup> A Kormány 1035/2012. (II.21.) Korm. határozata Magyarország Nemzeti Biztonsági Stratégiájáról.  
[http://2010-2014.kormany.hu/download/f/49/70000/1035\\_2012\\_korm\\_határozat.pdf](http://2010-2014.kormany.hu/download/f/49/70000/1035_2012_korm_határozat.pdf);  
letöltés: 2016.02.22.

először a teljes körű hatékony karbantartás (Total Productive Maintenance, TPM) módszere, amely az üzemeltetési folyamatokba a hatékonyság érdekében a karbantartók mellett a fejlesztőket és a felhasználókat is bevonja (teljes körű minőségmenedzsment, TQM).

A stratégia más szakterületekre is kiterjeszhető (pl. fejlesztés, beszerzés, értékesítés), ezáltal olyan vállalati rendszer alakítható ki, amely maximalizálja a rendszer hatékonyságát.

A kockázatalapú stratégia (Risk Based Maintenance, RBM) a meghibásodási valószínűségek és a lehetséges következmények értékelésével (Risk Based Inspection, RBI) határozza meg a felülvizsgálati és a karbantartási igényeket. Az állapotfüggő karbantartással együtt alkalmazva hatékony módszer dolgozható ki, amely növeli a rendszerelemek rendelkezésre állását, előre meghatározott kockázati szint mellett minimalizálja a karbantartásra fordítandó összeget és gyors információáramlást biztosít az üzemeltetési tevékenység biztosításához.<sup>4</sup>



5. ábra: A karbantartási stratégiák fejlődése<sup>5</sup>

A szoftvertervezés és -üzemeltetés fentiekhez hasonló stratégiáit a Nemzetközi Szabványügyi Szervezet és a Nemzetközi Elektrotechnikai Bizottság ISO/IEC 14764-es szabványa foglalja össze.<sup>6</sup>

<sup>4</sup> Karbantartási stratégiák fejlődése.  
<http://www.delta3n.hu/gepvedelem/karbantartasi-strategiak-fejlo%C3%A9se>; letöltés: 2016.02.01.

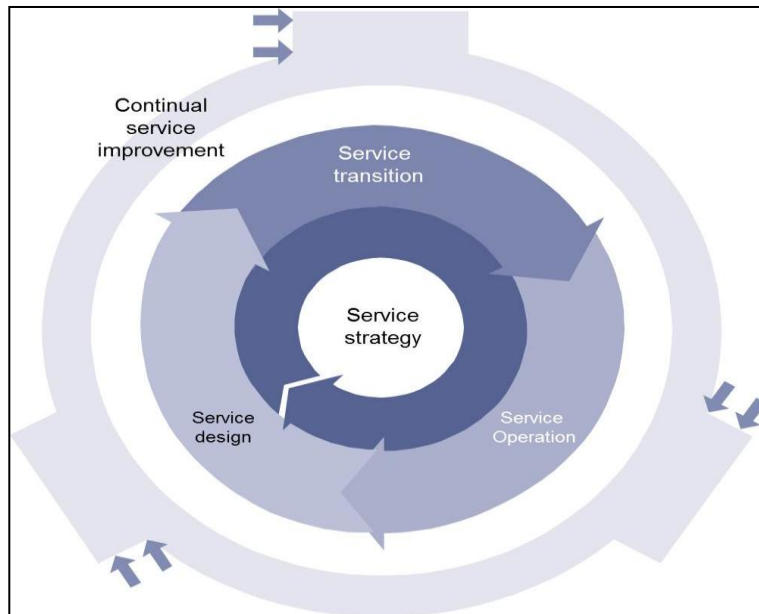
<sup>5</sup> Uo.

<sup>6</sup> ISO/IEC 14764: 2006 – Software Engineering – Software Life Cycle Processes – Maintenance.  
[http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=39064](http://www.iso.org/iso/catalogue_detail.htm?csnumber=39064) [nem ingyenes tartalom]

## IT-szolgáltatásmenedzsment

Üzleti modelleket vizsgálva megállapítható, hogy a piaci szereplők termékkatalógusai jellemzően új és meglévő termékek, megoldások keverékéből tevődnek össze. A bejáratott termékek éveken keresztül bevételi forrást jelentenek, a versenyhelyzetben azonban elengedhetetlen folyamatosan újabb és újabb szolgáltatások bevezetése.

Az IT-szolgáltatásmenedzsment bevált gyakorlatait („best practices”) összefoglaló ajánlásgyűjtemény, az ITIL életciklus-szemléletmódja jól szemlélteti ezt a körforgást, miközben megpróbálja egyértelműen meghatározott feladat- és felelősségi körökkel jól elhatárolni az életciklus egyes szakaszaiban jelentőséggel bíró munkafolyamatokat.



6. ábra: Az ITIL-életciklus<sup>7</sup>

Ez az életciklus-szemlélet az ITIL (Information Technology Infrastructure Library) legfrissebb, 2011-ben bevezetett harmadik kiadásában jelent meg.<sup>8</sup> Az IT-szolgáltatások nyújtása során szerzett tapasztalatokat összegző kiadványok öt alappillére épülnek, amelyek a következők:

- szolgáltatásstratégia (SS – Service Strategy);
- szolgáltatástervezés (SD – Service Design);

<sup>7</sup> [www.conferenzaitmsmf.it](http://www.conferenzaitmsmf.it); letöltés: 2015.11.21.

<sup>8</sup> What is ITIL® Best Practice? <https://www.axelos.com/best-practice-solutions/itil/what-is-itil>; letöltés: 2015.11.21.

- szolgáltatásátadás (ST – Service Transition);
- szolgáltatásüzemeltetés (SO – Service Operation);
- folyamatos továbbfejlesztés (CSI – Continual Service Improvement).

A felsoroltak megvalósításához az ITIL folyamatokat és funkciókat definiál. A folyamatok tevékenységek rendezett, dokumentált együttese, jól meghatározott kimenettel és mérhető eredményekkel. A folyamatokat megvalósító csapatok a funkciók, ahol minden csoporttagnak jól definiált szerepe van. E szerepek alapján a felelősségi körök is egyértelműen meghatározhatóak, az egyes tevékenységekhez kötődően mátrixszerűen ábrázolhatóak. Az alábbi példában a felelősségi szinteket az angol terminológia kezdőbetűi jelölik, az ITIL ezek alapján nevezi RACI-mátrixnak a felelősségi mátrixot.

RACI-mátrix <sup>9</sup>	szerep A	szerep B	szerep C	szerep D
tevékenység 1	AR	C	I	I
tevékenység 2	A	R	C	C
tevékenység 3	I	A	R	I
tevékenység 4	I	A	R	I
tevékenység 5	I	R	A	I

A **stratégia** a szolgáltatást folyamatos mozgásban, működésben tartó motor tengelye. A stratégia határozza meg a szolgáltató távlati céljait, pozícióját a versenytársakhoz képest, a célok eléréséhez végrehajtandó terveket és az ehhez szükséges működési rendet. Mindez szükséges ahhoz, hogy a szervezet várt eredményei megvalósuljanak. A stratégia fontos rész célja, hogy az adott szervezet megértse, milyen képességekkel rendelkezik a kitűzött célok megvalósításához. A stratégia – azon túl, hogy meghatározza a szolgáltatások előállításának módját az ügyfél által várt üzleti eredmények érdekében – kiter a szolgáltatások felügyeletére is.

A **szolgáltatástervezés** a stratégia első számú megvalósítója: megtervezi a szolgáltatásokat azok munkamódszereivel, folyamataival, irányelveivel együtt. Segíti a szolgáltatások bevezetését a támogatott környezetekbe, mindvégig szem előtt tartva a költséghatékonyságot és az ügyfél-elégedettséget. A tervezés folyamán kerül sor a kockázatok azonosítására is. A szolgáltatástervezés felel a menedzsmentinformációs rendszerek, mint például a szolgáltatások követelményeit tartalmazó szolgáltatásportfólió és a folyamatos továbbfejlesztést támogató mérési módszerek, mérőszámok tervezéséért.

<sup>9</sup> R – Responsible – felelős, A – Accountable – számonkérhető, C – (to be) Consulted – bevonandó, I – (to be) Informed – tájékoztatandó.

Az átfogó és integrált szolgáltatástervezés egyaránt figyelembe veszi a szervezet (dolgozók, kultúra), a partnerek (szállítók, gyártók), a folyamatok (szerepek, kompetenciák, tevékenységek) és a termékek (technológia, rendszerek, eszközök) igényeit, adottságait is.

A tervezés eredményeként megszülető megoldás szolgáltatási tervcsomag (Service Design Package – SDP) formájában kerül átadásra. Az SDP tartalmazza a szolgáltatással szemben támasztott követelményeket, a szolgáltatás leírását, a szervezeti felkészültségre vonatkozó adatokat, valamint a szolgáltatás életciklustervét. A tervezés egyik kulcsfontosságú eleme a szolgáltatásszintű megállapodás (Service Level Agreement – SLA), amely rögzíti a szolgáltató és az ügyfél felelősségeit. Az SLA-ban megfogalmazott célok elérését segíti az üzemeltetésszintű megállapodás (Operational Level Agreement – OLA), amely az ügyfél saját IT-részlege és a többi belső divízió között jön létre. A szolgáltatásátadás fő célja biztosítani, hogy a szolgáltatások teljesítsék a megrendelő (üzleti oldal) várakozásait – legyen szó új vagy módosított, esetleg visszavont szolgáltatásról. Ide sorolható: szolgáltatáskiadások üzembe állítása, szolgáltatások változásával kapcsolatos kockázatok kezelése, pontos nyilvántartás vezetése a szolgáltatásnyújtáshoz szükséges eszközökről (pl. konfigurációmenedzsment-adatbázis – CMDB), szoftverek és dokumentációk másolatainak hiteles tárolása (hiteles médiatár – DML), változásmenedzsment stb.

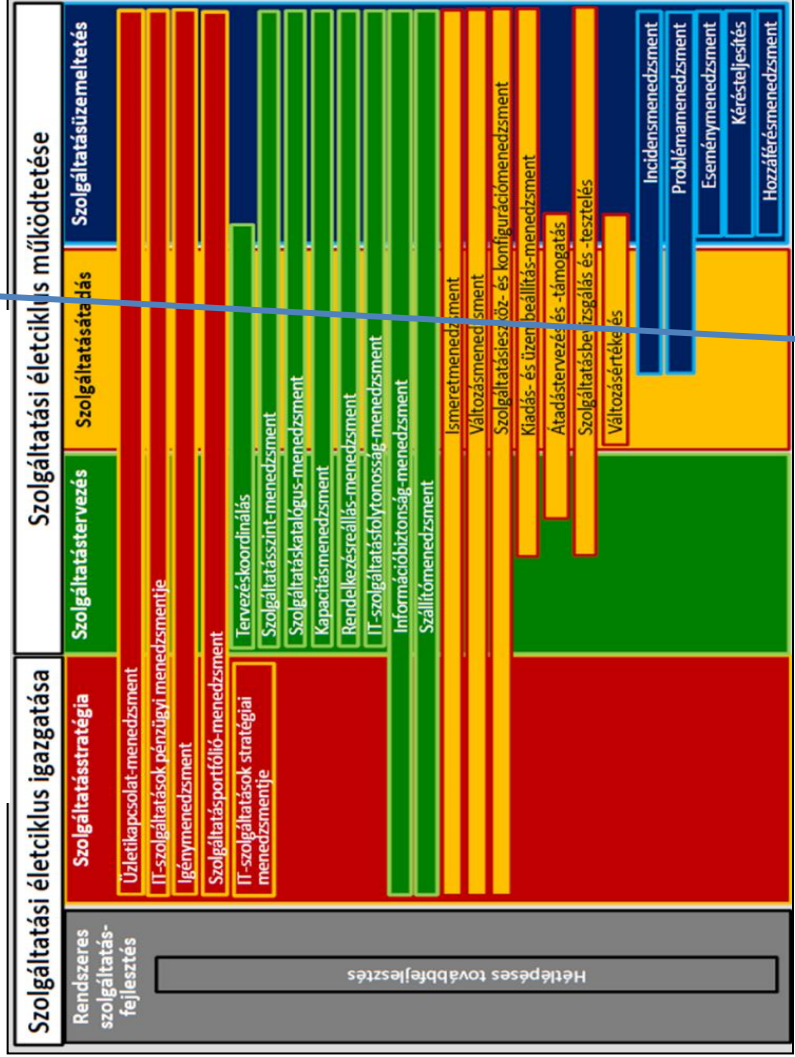
A szolgáltatás**üzemeltetés** feladata, hogy a megtervezett és előkészített szolgáltatásokat a megállapodott szinten nyújtsa és felügyelje. Folyamatai:

- eseménymenedzsment: a rendszerelemek, illetve a szolgáltatás állapotváltozásainak figyelemmel kísérése;
- incidensmenedzsment: nem tervezett szolgáltatásmegszakadások, minőségcsökkenések kezelése;
- problémamenedzsment: incidensek ismeretlen okainak feltárása, azok dokumentált rögzítése;
- kéresteljesítés: a felhasználóktól származó szolgáltatáskérések (pl. hozzáférés, áthelyezés, fogyóeszközök rendelése) felügyelete;
- hozzáférésmenedzsment: hozzáférési jogosultságok biztosítása az információbiztonsági irányelvek figyelembe vételével.

Az üzemeltetés az incidenseket hatásuk és sürgősségük alapján priorálja, elhárításukra megcélzott megoldási időt definiál. Az incidensek fogadására létrehozott funkció az ügyfélszolgálat, amely kezdeti diagnózist követően a tapasztalatok és az ismert hibák adatbázisa alapján megkísérli az incidenskezelést. Amennyiben nagyobb szakértelemmel rendelkező részlegek bevonása szükséges, akkor az ügyfélszolgálat eskalálja az incidenskezelést. Funkcionális eskaláció során az egyre szűkebb szakterületre specializálódott csoportokhoz, esetleg külső vállalkozókhöz, míg hierarchikus eskaláció során a szervezeti struktúrában magasabb pozícióban elhelyezkedő menedzserek hatáskörébe kerül át a feladat.

7. ábra. Az ITIL 26 folyamata  
ITIL Foundation Course alapszintű tananyag

Fejlesztés  
vs.  
Üzemeltetés





Eszkalációra tehát akkor van szükség, amikor az üzemeltető szaktudása nem elegendő. A tudástár az ismerthiba-adatbázis folyamatos karbantartásával bővíthető, de nem elhanyagolható a napi munkavégzés során felhalmozott tapasztalatok szerepe sem. Helyes irány a funkcionális eszkáliciók számának csökkentésére az üzemeltetők képzése. Helytelen a nagyobb szaktudással rendelkezők – adott esetben fejlesztők – „megint megcsinálom helyette” hozzáállása. Az üzemeltető számára a tudás hiánya mellett akadályt jelenthet még:

- információ hiánya: dokumentáltsággal és a tudástárak naprakészen tartásával megelőzhető;
- jogosultság hiánya: nem merülhet fel, ha a szerepek egyértelműen meghatározásra kerültek (lásd: RACI-mátrix);
- eszköz hiánya: megfelelő karbantartási stratégia megválasztásával mind a felügyelethez, mind a folyamatos mérésekhez, mind a javításhoz szükséges eszközök rendelkezésre állása biztosítható.

A szolgáltatások folyamatos **továbbfejlesztése** átível az összes többi ITIL-életciklus felett. A szolgáltatásminőség javítása érdekében az összes folyamatot és funkciót felügyeli, folyamatos méréseket végez és szem előtt tartja a költséghatékonyságot is. A folyamatok mérése során elterjedt kifejezés a KPI (Key Performance Indicator). Ez olyan fő teljesítménymutatót takar, amely érdemi visszajelzést ad az adott folyamat minőségéről, ezért alakulása hosszabb időn keresztül monitorozandó.<sup>10</sup>

Ha az ITIL-életciklusokat a „fejlesztés és az üzemeltetés szétválasztása” szemszögéből nézzük, jól látható, hogy a határt a szolgáltatásátadásnál kell meghúzni. Ez a határ azonban nem egy éles vonal, mint ahogy a folyamatok sem szoríthatók az egyes életciklusszakaszok határai közé, bizonyos esetekben azokon jóval túlnyúlnak.

## DevOps

Napjainkban a felhasználói visszajelzések üteme egy termékkel, szolgáltatással kapcsolatban olyan mértékben felgyorsult, hogy annak a szolgáltatónak, aki lépést kíván tartani a felhasználói igényekkel, gyökeresen át kell alakítania a jól bevált, berögzült munkafolyamatait. Ez a szemlélet a DevOps (Development and Operations), amely elmosza a határokat a fejlesztés és az üzemeltetés között.<sup>11</sup> A hagyományos megközelítés értelmében a fejlesztőknek mindig egy jól meghatározott idő áll rendelkezésükre a következő kiadásig (release). Amit ezen idő alatt nem sikerült kifejleszteni, átkerült a következő kiadásba. Az üzemeltetés számára mindig elegendő idő áll rendelkezésre az új funkciók tesztelésére.

A felgyorsult igényeket kielégítő új szemlélet, a folytonos végrehajtás (continuous delivery) és a folytonos integrálás (continuous integration) teljesen új munkamódszereket kíván.

<sup>10</sup> ITIL Foundation Course alapszintű tananyag (IQSOFT – John Bryce Oktatóközpont, 2015).

<sup>11</sup> GREEN, Daniel: Mi is az a DevOps? <https://schoherzbasis.hu/hirek/reszletek/Mi-is-az-a-DevOps>; letöltés: 2016.02.13.

A fejlesztés elvárja, hogy minden új szolgáltatásuk azonnal megjelenjen a felhasználóknál. Az üzemeltetésnek meg kell birkóznia azzal a kihívással, hogy az új fejlesztések folyamatosan átkerüljenek a tesztkörnyezetbe és „gyártásba” is kerüljenek (production). Úgy kell megőrizni a termékek minőségét és stabilitását, hogy a tesztre jóval kevesebb idő jut, mint korábban. Éppen ezért a fejlesztőknek támogatniuk kell az üzemeltetést olyan automatikus diagnosztikai eszközökkel, amelyek folyamatosan visszajelzéseket küldenek a rendszer állapotáról, minőségi paramétereiről (lásd KPI-k).

Ahhoz tehát, hogy megtaláljuk az egyensúlyt az innováció és a stabilitás között, elengedhetetlen a szemléletváltás a fejlesztés és az üzemeltetés munkafolyamataiban, valamint a két fél közti kommunikációban.

A fejlesztő szemszögéből:

- közelebb kell kerülni a termék „életútjához”: jobban meg kell ismerni a felhasználói szokásokat és azokat a rendszerarchitektúrákat, ahol szolgáltatásaik beüzemelésre kerülnek;
- jobban bele kell folyni a tesztelésbe: nem elég, ha egy kód hibamentes, meg kell ismerni, hogy hogyan teljesít a felhasználók által üzemeltetett eszközökön;
- meg kell ismerni az üzemeltetés által használt mérési módszereket, meg kell érteni a mérőszámokat, meg kell ismerni a folyamatok működését, azok hatásait egymásra.

Az üzemeltető szemszögéből:

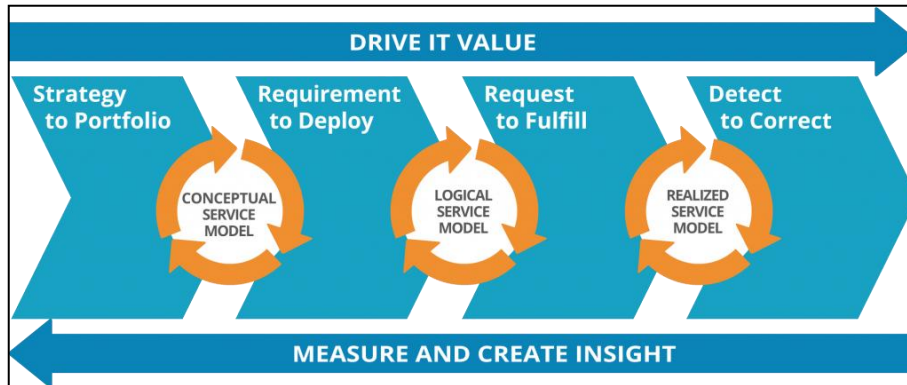
- ismerni kell, milyen változások kerülnek bele a rendszerbe; ehhez bele kell folyni a fejlesztésbe a felhasználói igények specifikálásától kezdődően;
- automatizálni kell a lehető legtöbb munkafolyamatot, nemcsak a változtatások, hanem a visszaállítások kezelésére is;
- megfelelő verziószámozással kell ellátni a rendszert;
- minden munkafolyamatot mérni kell a fejlesztés, a tesztelés és a termelés során.

A folytonos integráció szerepe akkor jelentős, amikor különböző fejlesztőcsapatok egy nagyobb projekt különböző elemein egymással párhuzamosan dolgoznak. Az egyes csapatok tagjai először saját fejlesztéseiket integrálják, majd a továbbfejlesztett részek integrálásával áll elő a termék aznapi új verziója. Ez a munkafolyamat minden egyes fejlesztőt és csapatot arra készítet, hogy a lehető legkorábbi munkafázistól kezdve együttműködjön a munkatársaival. A folytonos integráció gyakorlott résztvevői azt a képességet is elsajátítják, hogy saját változtatásaik hatását folyamatosan nyomon kövessék a rendszer stabilitása és a rendszerelemek együttműködése szempontjából.

A folytonos végrehajtás a következő szintre emeli a folytonos integrációt: minden egyes új („integrált”) verziót azonnal továbbítja tesztelésre (QA – Quality Assurance), majd az üzemeltetőkön keresztül a felhasználókhöz.<sup>12</sup>

### IT4IT

Az Open Group<sup>13</sup> által 2014-ben bevezetett szabvány, az IT4IT újfajta megközelítéssel kíván támpontot adni az IT-vezetők számára új szolgáltatások bevezetése és integrálása során. Az új szabvány csak akkor lehet sikeres, ha nem vezet be a korábrinál összetettebb, bonyolultabb eljárásrendeket. Emellett fő céljai a költségek csökkentése, a nagyobb fokú automatizáltság, valamint átláthatóság az üzleti partnerek számára.



8. ábra: IT4IT értéklánca<sup>14</sup>

Az IT4IT a megvalósított üzleti érték szempontjából közelít a szolgáltatások fejlesztéséhez és bevezetéséhez. Az értékteremtési láncot négy fő szakaszra osztja:

- **stratégiától a portfólióig** (Strategy to Portfolio, S2P): stratégiát nyújt a kiegyensúlyozott szolgáltatásportfólióhoz, minőségi adatokkal támogatja a döntéshozást és kulcs-mérőszámokat (KPI) biztosít az üzleti tervekhez és a partnerekkel folytatott kommunikációhoz;
- **követelménytől a megvalósításig** (Requirement to Deploy, R2D): az üzleti igények megvalósítása a megfelelő időben; hagyományos és agilis fejlesztési stratégiákat egyaránt támogat, miközben kulcsfontosságú határoz meg a fejlesztés és az integrálás során, amelyek folyamatos továbbfejlesztést tesznek lehetővé;

<sup>12</sup> SHARMA, Sanjeev: Understanding DevOps. <https://sdarchitect.wordpress.com/understanding-devops/>; letöltés: 2016.02.13.

<sup>13</sup> IT4IT – The Open Group szabvány. <http://www3.opengroup.org/IT4IT>; letöltés: 2016.02.21.

<sup>14</sup> Business Process Incubator.

- **kéréstől a teljesítésig** (Request to Fulfill, R2F): átlátható katalógust vezet az elérhető szolgáltatásokról, méri és menedzseli a kérés teljesítések hatékonyságát és a szolgáltatások összköltségét;
- **észleléstől a javításig** (Detect to Correct, D2C): megelőzi, illetve megoldja a szolgáltatásnyújtás során felmerülő hibákat, problémákat; hatékony, átlátható üzemeltetéssel törekszik a javításhoz szükséges átlagos idő (MTTR – mean time to repair) csökkentésére.

Az IT4IT a teljes értékláncot olyan referenciaarchitektúrával támogatja meg, amely teljes körű (holisztikus) iránymutatást ad az IT-szolgáltatások bevezetéséhez.

### Összegzés

A fentiekben bemutatott IT-menedzsment ajánlások és szabványok mindegyike megjeleníti a sikeres szolgáltatásmenedzsment eléréséhez szükséges eljárásokat – fejlesztői és üzemeltetői oldalról egyaránt. Tisztán látható, hogy míg az ITIL tiszta és egyértelmű folyamatokkal és funkciókkal definiálja az egyes életciklusokat, a korszerű és agilis módszertanok az ITIL alapelveire támaszkodva halványítják el a határvonalat fejlesztés és üzemeltetés között a gyors, költséghatékony és a felhasználói igényekkel lépést tartó, folyamatos szolgáltatásfejlesztés érdekében.

Mindig az adott szervezet struktúrája, kultúrája, szolgáltatási köre és IT-portfóliója határozza meg, hogy a fejlesztés és az üzemeltetés közti határvonalat milyen élesen húzzuk meg. Az együttműködést azonban folyamatosan biztosítani kell a fejlesztés és az üzemeltetés között, hiszen csak ebben a formában biztosíthatók a felgyorsult felhasználói igényeket kielégítő, teljes körű informatikai szolgáltatások, valamint az információbiztonsági törvényben megfogalmazott szigorú biztonsági követelmények, az adatok és az információk bizalmassága, sértetlensége és rendelkezésre állása, valamint az elektronikus információs rendszer és elemeinek sértetlensége és rendelkezésre állása tekintetében.<sup>15</sup>

### FELHASZNÁLT IRODALOM

- 2013. évi L. törvény az állami és önkormányzati szervek elektronikus információbiztonságáról (Ibtv. 5.§)
- A Kormány 1035/2012. (II.21.) Korm. határozata Magyarország Nemzeti Biztonsági Stratégiájáról  
[http://20102014.kormany.hu/download/f/49/70000/1035\\_2012\\_korm\\_határozat.pdf](http://20102014.kormany.hu/download/f/49/70000/1035_2012_korm_határozat.pdf);  
letöltés: 2016.02.22.
- GREENE, Daniel: Mi is az a DevOps?  
<https://schonherzbazis.hu/hirek/reszletek/Mi-is-az-a-DevOps>; letöltés: 2016.02.13.

<sup>15</sup> 2013. évi L. törvény az állami és önkormányzati szervek elektronikus információbiztonságáról (Ibtv. 5.§)

- HOLTAI András – MAGYAR Sándor – PUSKÁS Béla: Az informatikai üzemeltetés általános kérdései. In: Felderítő Szemle, 2005. november. 14. évf. 4. sz.  
<http://knbsz.gov.hu/hu/letoltes/fsz/2015-4.pdf>; letöltés: 2016.02.22.
- ISO/IEC 14764: 2006 – Software Engineering – Software Life Cycle Processes – Maintenance. [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=39064](http://www.iso.org/iso/catalogue_detail.htm?csnumber=39064)  
[nem ingyenes tartalom]
- IT4IT – The Open Group szabvány. <http://www3.opengroup.org/IT4IT>; letöltés: 2016.02.21.
- ITIL Foundation Course alapszintű tananyag (IQSOFT – John Bryce Oktatóközpont, 2015).
- Karbantartási stratégiák fejlődése.  
<http://www.delta3n.hu/gepvedelem/karbantartasi-strategiak-fejlo%C3%A9se>;  
letöltés: 2016.02.01.
- OBUCH László: Informatikai rendszerek tervezése  
[http://aktivitas-tizsk.hu/elearning/Obuch\\_Laszlo/Tervezesi\\_alapismeretek.pdf](http://aktivitas-tizsk.hu/elearning/Obuch_Laszlo/Tervezesi_alapismeretek.pdf);  
letöltés: 2016.02.03.
- SHARMA, Sanjeev: Understanding DevOps.  
<https://sdarchitect.wordpress.com/understanding-devops/>; letöltés: 2016.02.13.
- What is ITIL® Best Practice?  
<https://www.axelos.com/best-practice-solutions/itil/what-is-itil>; letöltés: 2015.11.21.
- [www.conferenzaitmf.it](http://www.conferenzaitmf.it); letöltés: 2015.11.21.
- [www.delta3n.hu](http://www.delta3n.hu); letöltés: 2016.02.01.