

TÉZISFÜZET

**Híd az elmélettől a gyakorlatig: szimuláció alapú ütemezési
teljesítményértékelések Alkalmazás
Életciklus-Menedzsmenthez**

Szerző:

JAKAB Róbert

Témavezető:

Prof. Dr. habil. KOSZTYÁN Zsolt Tibor



PANNON EGYETEM

Veszprém
2024

Tartalomjegyzék

1	Bevezetés	1
1.1	Alkalmazás Életciklus-Menedzsment jelentősége	1
1.2	Kutatási célok és kérdések	2
2	Szakirodalmi háttér és kutatási feltételezések	3
2.1	Alkalmazás Életciklus-Menedzsment	3
2.2	Projektmenedzsment megközelítések	5
2.3	Erőforrás-korlátos projekt ütemezési probléma	6
2.4	Kockázatkezelés ALM környezetben	7
2.5	Kutatási feltételezések	7
3	Eredmények	8
4	A kutatás tézisei	24
5	Összefoglalás	25
5.1	Újszerű eredmények, hozzájárulás az irodalomhoz	26
5.2	Gyarkolati vonatkozások	27
5.3	A kutatás összegzése	28
6	A szerző kapcsolódó publikációi	30
	Felhasznált irodalom	31

1 Bevezetés

Az elmúlt évtizedekben a szoftverek alkalmazása a tudományos területeken és az üzleti folyamatokban egyaránt nélkülözhetetlenné vált. A szoftveralkalmazások (applikációk) fejlesztésének és karbantartásának evolúciója éppen ezért fontos területté vált mind a tudományos életben, mind az üzleti szférában. A szoftverfejlesztési folyamatban a hangsúly az egyszeri, lineáris megközelítésről az agilis, rendszeres szoftverfrissítéseket magába foglaló, rugalmas tartalomkezelésre helyeződött át. Számos gyártó kínál olyan fejlesztést és üzemeltetést kiszolgáló eszközláncokat, amelyek támogatják az életciklus-fejlesztést az ilyen szoftverkonceptiókhoz, azonban erről a kapcsolódó tudományos szakirodalomban még mindig kevés információ található. Nincsen például még egyértelmű és széleskörben elfogadott definíció, valamint a releváns módszerek hatékonyságáról sem található megfelelő és kielégítő információ a tudományos irodalomban.

1.1 Alkalmazás Életciklus-Menedzsment jelentősége

Az **Alkalmazás Életciklus-Menedzsment**¹ számos iparágban (például IT, autóipar, légi közlekedés) már a vállalkozások kulcsfontosságú fókuszpontjává vált, amelynek célja, hogy megoldást nyújtson a felmerülő fejlesztési és üzemeltetési feladatokra az alkalmazás-szoftverek teljes életciklusán át, a kezdetektől egészen azok kivezetéséig. Az elmúlt néhány évtizedben a szoftverközpontú gazdaság irányába történő páratlan átalakulás szemtanúi lehettünk, amelyben a vállalkozások szoftveralkalmazásokat használnak mind az innováció, mind a folyamatok egyszerűsítése, valamint az ügyfelek számára történő értékteremtés érdekében. Éppen ezért a szoftverfejlesztés és karbantartás gazdasági hatása is drámaian megnőtt (Mishra és Alzoubi, 2023; Al-Saqqá és tsai., 2020; van den Ende és van Marrewijk, 2014).

Belátható tehát, hogy a szoftverprojektek kudarcra ezért gazdaságilag katasztrofális is lehet. A késések, a költségvetés túllépése és az optimálistól elmaradó szoftverminőség nemcsak költséges, hanem a piaci versenyképességet is rontja. Az ALM megfelelő alkalmazása elengedhetetlen e kockázatok mérsékléséhez és a szoftverfejlesztési kihívások gazdasági hatékonyságának maximalizálásához. Ez annak köszönhető, hogy az ALM képes kezelni a teljes szoftveralkalmazási életciklust, és kulcsszerepet játszik annak biztosításában, hogy a befektetések fenntartható gazdasági megtérülést eredményezzenek. Ezért kulcsfontosságú a hatékonyság a szoftveralkalmazásoknál, például a megfelelő ütemezési módszertan alkalmazása az ALM környezetben, akár a fejlesztés, akár az üzemeltetés fázisában.

Mivel az akadémiai irodalomban nem létezik egyértelműen és általánosan elfogadott definíció az ALM-re, a szoftverkészítő vállalatok megragadták a lehetőséget, hogy a maguk javára fordítsák a helyzetet és saját céljaiknak megfelelően alakítsák

¹Angolul Application Lifecycle Management (ALM), magyarra fordítva még az Alkalmazás Életciklus-Kezelés és az Applikáció Menedzsment kifejezéseket is használja a szakirodalom.

azt. Ez ahhoz vezetett, hogy az üzleti érdekeket szolgáló, eltérő megfogalmazások és definíciók már a tudományos elemzések és fejlesztések továbblépését is hátráltathatják. A megfelelő akadémiai és szakmai háttérmunka hiányában ezért az üzleti szereplők a már elérhető projektmenedzsment módszereket alkalmazzák, mint amilyenek például a tradicionális- (TPM), az agilis- (APM) és a hibrid projektmenedzsment (HPM), amelyek esetleg csak részben felelnek meg céljaik eléréséhez, és valószínűleg alacsonyabb hatékonysággal működnek, mint egy, az ALM környezetre szabott ütemezési megoldás.

1.2 Kutatási célok és kérdések

A disszertációm kutatásának hármas célja a következő volt:

- ✓ Szakirodalmi kutatás, meghatározni
 - [+] az elérhető ALM definíciót és annak értelmezési tartományát;
 - [+] egy egységesített, átfogó definíciót, amely alapul szolgálhat egy ALM modellnek, elősegítve a további módszertani kutatásokat.
- ✓ A mátrixos reprezentáció alkalmazhatóságának a vizsgálata, ALM ütemezésre, beleértve
 - [+] a szimulációs (modellezett) környezet felállítását;
 - [+] a TPM, APM és HPM ütemezések megoldhatósági vizsgálatát;
 - [+] a TPM, APM és HPM ütemezések hatékonyságának vizsgálatát.
- ✓ a kockázati tényezőket és azok hatását az ALM ütemezési struktúrájában.

Validálásként egy releváns ALM esettanulmány kivitelezése ütemezési teljesítményértékeléssel.

Tekintve a téma jelentőségét, valamint a korábban említett célkitűzéseket, jelen dolgozat a következő kutatási kérdésekre keresi a választ:

KK1: Hogyan azonosítható a rendelkezésre álló tudományos irodalom alapján egy tervezési modell, amely az alkalmazás életciklus-menedzsmentet (ALM) reprezentálja, és amely felhasználható ütemezési feladatokhoz?

KK2: A jelenlegi projektmenedzsment ütemezési módszertanok (TPM, APM, HPM) eredményeznek-e megvalósítható megoldásokat az ALM környezetben, és azok hogyan teljesítenek?

KK3: Melyek az ütemezési probléma kockázati tényezői az ALM-környezetben, és ezek hogyan befolyásolják a megvalósíthatóságot és az ütemezési teljesítményt?

2 Szakirodalmi háttér és kutatási feltételezések

Ebben a fejezetben a kitűzött ALM specifikus kutatási téma megértéséhez szigorúan szükséges területek kerülnek tömörítve bemutatásra. Ezek ismerete a jelen mű kutatásmódszertani részének megértéséhez és az eredmények kontextusba helyezéséhez, értelmezéséhez szükségesek. A 3.1 alfejezet az Alkalmazás életciklus menedzsmenttel kapcsolatos információkat, a 3.2 és 3.3 alfejezetek az ütemezési problémával kapcsolatban, illetve a 3.4 alfejezet a kockázatkezeléssel kapcsolatos információkat tartalmazzák. Az elméleti áttekintés után a 3.5 alfejezet a kutatási feltételezéseket mutatja be az elméleti anyagok átvizsgálása után.

2.1 Alkalmazás Életciklus-Menedzsment

Szoftveres alkalmazásnak vagy applikációnak nevezünk egy meghatározott típusú szoftvert, amelyet arra terveztek, hogy egy adott funkciót vagy kapcsolódó funkciók készletét hajtsa végre a felhasználók számára. Ezen alkalmazások jellemzője, hogy felhasználóbarátok, és tipikusan egy meghatározott cél érdekében lettek létrehozva. Tudományos kontextusban az „alkalmazás” a szoftverek egy részhalmaza, amely kifejezetten a különféle területek felhasználói igényeinek kielégítésére kifejlesztett programokra vonatkozik. A továbbiakban az ALM-kontextusban az alkalmazásnak ezen jelentését értem.

A szoftverprojektek teljesítménye történelmileg elmarad a hagyományos, például építőipari projektekhez képest. Erre a jelenségre már a kilencvenes években felfigyeltek és publikálták a CHAOS jelentésben, amit azóta is rendszeresen frissítenek (StandishGroup, 2020). Számos szoftverfejlesztés-specifikus tényező befolyásolja ezt az alulteljesítést, amelyek más termékfejlesztéssel kapcsolatos projektekben nincsenek vagy csak részben vannak jelen. Ilyen például a relatíve komplex tervezési folyamatok, az egyértelmű követelmények időbeni rendelkezésre állásának fontossága, a konkrét együttműködési módszerek és eszközök összehangolása, emeli ki a tanulmány. Mivel az ALM is alapvetően a szoftveres applikáció fejlesztése és üzemeltetése köré szerveződik, ezért ezt a jelenséget mindenképpen figyelembe kell venni.

Az ALM, mint fogalom, az applikáció életciklus kezelés köré szerveződik, amelynek történelmi alapja a termékéletciklus-menedzsment (angolul Product Lifecycle Management - PLM). Kezdetben a szoftver csupán egy rész volt a PLM-ben, azonban az egyre nagyobb számban megjelenő és egyre nagyobb komplexitású szoftverek kezelése szakosodást igényelt, amit először szoftverfejlesztési életciklusnak neveztek el (SDLC²), majd a későbbiekben már mint applikáció vagy alkalmazás életciklus-menedzsment (ALM) jelent meg a szakirodalomban. Ennek az egyedülálló IT-területnek a látszólagos fontossága ellenére a kutatói közösség általában a PLM- és ALM-eszköz készítőkre hagyta a két terület összefésülését, akik hosszú

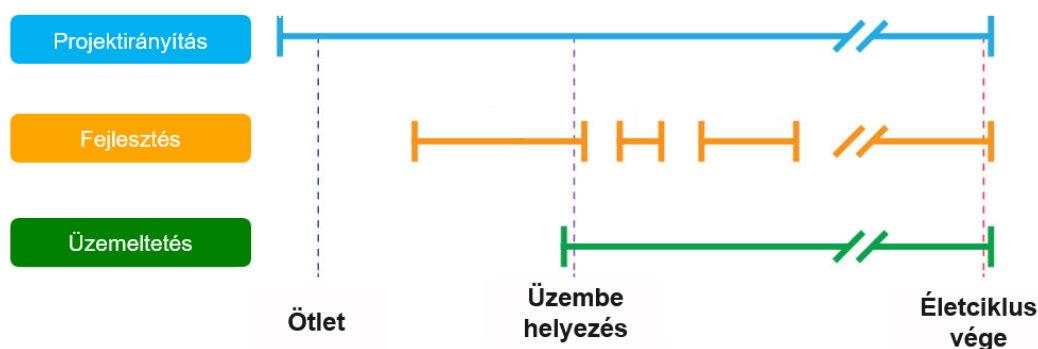
²Angol rövidítése a Software Development Life-Cycle kifejezésnek.

távon azok konvergenciáját (Deuter, Otte és tsai., 2019; Deuter és Rizzo, 2016; Rao és Palaniappan, 2020) támogatják. Ennek köszönhető, hogy szignifikáns átfedések és kapcsolatok vannak a PLM és az ALM projekt- és programmenedzsment elemei között. Például, a hardver és a szoftver műtermékek közötti kapcsolatok, a változáskezelés, az együttműködés és a jelentéskezelések (Deuter és Rizzo, 2016) területén.

Belátható, hogy a temérdek mennyiségű okoseszköznek köszönhetően, például az infokommunikációs szektorban vagy akár az autóiparban is, az ALM kihívásával kell szembenéznük a gyártóknak, éppen ezért esett a választásom ezen terület kutatására.

A disszertáció esettanulmánya is megmutatja, hogyan valósul meg az ALM egy autóipari beszállító cégnél, és milyen kihívásokkal állnak szemben a mindennapi üzleti életben a vállalatok.

Az *Alkalmazás Életciklus-Menedzsment terület* újdonsága az eddigi projekt szemlélethez képest több kutató szerint is (Chappell és tsai., 2010; Corallo és tsai., 2020; Rossberg, 2019; Rossman, 2010) abban rejlik, hogy az ALM egy olyan átfogó szoftverfejlesztési és üzemeltetési megközelítés, amely a szoftveralkalmazás teljes élettartamát felöleli, az ötlet megszületésétől a szoftverfejlesztésen keresztül annak üzembehelyezésén át, a végső visszavonásáig. Ebben az életciklusban több szempont is jelen van, de ami az ALM-et egyedivé teszi a hagyományos projektmenedzsmenthez vagy szolgáltatásorientált megközelítéshez képest, az az, hogy a fejlesztési folyamatban a fő fázis (első, hosszabb narancssárga szakasz) után potenciálisan további, nem várt feladatok, sőt alprojektek jelenhetnek meg elszórtan (kisebb narancssárga szakaszok) ahogy az alábbi 1. ábrán is látható. Ez a fajta megközelítés egyedülálló az alkalmazásfejlesztésben, hiszen már az Üzemeltetési (zöld színnel jelölt) szakaszban több tervezett vagy nem tervezett változás történik, a projekt hivatalos beüzemelése, átadása után. Ezek nem csak hibajavítás jellegűek lehetnek, hanem további fejlesztési, tartalmi bővítési feladatok, akár alprojektek is, amelyek például az időközben fellépő piaci és felhasználói igények azonnali kielégítésére szolgálnak.



1. ábra. Az ALM fő folyamatai (Chappell és tsai., 2010) [saját fordítás]

Tekintettel arra, hogy a tartalmi változások hatása egy szoftverfejlesztésnél meghatározóbban érvényesül, a klasszikus projektmenedzsment folyamatmodellek,

mint például a vízésés fejlesztési modell, amely már a projekt elején egyértelmű specifikáció-meghatározást vár el, és csak minimális változtatást tolerál, nem tud hatékonyan teljesíteni ebben az alapvetően megváltozott környezetben. Így módszertani szempontból rugalmasabb megközelítések válhatnak szükségessé a helyzet megoldására, amilyen például az agilis-projektmenedzsment (SGI, 2019) vagy a részben hagyományos, részben agilis kombinációt tartalmazó, úgynevezett hibrid projektmenedzsment megközelítések ebben az ALM-környezetben. Továbbá látni kell, hogy óriási hiányosságok vannak még az ALM-mel kapcsolatos tudományos területen, annak újdonsága és speciális konfigurációja miatt. Azonban azokra a kérdésekre, hogy miként lehet jól azonosítani és leírni az ALM problémát, és hogyan lehet pontosan az ilyen ALM specifikus helyzeteket hatékonyan kezelni, egyelőre nincs elérhető, mindenre kiterjedő tudományos irodalmi forrás a szakterületen.

Összefoglalva tehát látható, hogy az ALM szakirodalma a tudományos körökben még mindig szűkös, és alapos vizsgálatra van szükség a megfelelő definíció és hatásköri leírás megtalálásához. Egy ilyen áttekintés a szisztematikus szakirodalmi áttekintéssel (SLR³) lehetséges, annak érdekében, hogy az alapvető ALM definíció és használható modell a KK1-re tisztázható legyen.

2.2 Projektmenedzsment megközelítések

A KK2-vel kapcsolatos módszertani kutatás az ütemezési feladatkör felderítésére és analízisre fókuszál. Ennek elérése érdekében a már létező, és az informatikai területeken is gyakran használt projektmenedzsment megközelítéseket vizsgáltam. Ezek közül hármát választottam ki, a tradicionális, az agilis és ezek kombinációját, a hibrid megközelítéseket. Természetesen további projektmenedzsment megközelítések vizsgálata is lehetséges, a téma releváns vizsgálata érdekében azonban úgy ítélem meg, hogy ezek vizsgálata szignifikánsan különböző eredményekkel szolgálhat, így a további megközelítések jövőbeni vizsgálatok tárgyát képezik majd.

A *tradicionális projektmenedzsment (TPM)* megközelítésben (például építési projekt vagy szoftverfejlesztési projekt, amely vízésés folyamatmodellt követ) az a fő kérdés, hogy mennyi időbe és ráfordított erőforrásba kerül az előre meghatározott követelmények megvalósítása. Ezért, habár a tartalom, amit végre kell hajtani adott, az idő, a költség és a minőség szükség esetén átváltható. Ez a megközelítés egy-nél több kivitelezési módot tesz lehetővé (olyan technológiákat, amelyek eltérő idő-/költség-/erőforrásigényt igényelnek), lásd a 1. táblázatban a felsorolt megközelítések összehasonlítását (Creemers, 2015).

A *agilis projektmenedzsment (APM)* megközelítésben az a fő kérdés, hogy az adott költségkereten és időintervallumon belül (például egy sprintben) mennyi szolgáltatást lehet beépíteni. Míg egy sprinten belül a tartalom fix, így új feladatok nem megengedettek, addig a tervezési időszakokban a sprintek között, azok átrendezhetőek,

³Angol rövidítés: Systematic Literature Review

új feladatok is megjelenhetnek. Az általános cél az, hogy minden megközelítés a lehető legmagasabb szinten valósítsa meg az előírt tartalmi célt (Rasnacis és Berzisa, 2017; SGI, 2019).

A *hibrid projektmenedzsmentben* (HPM) a hagyományos és az agilis módszerek ötvöződnek, lehetővé téve új tevékenységek megjelenését és bevonását bármikor, valamint több projekt egyidejű kezelésére is alkalmas (Reiff és Schlegel, 2022).

1. táblázat. Különbő projektmenedzsment megközelítések összehasonlítása

Megközelítés	Projekt struktúra	Új feladatok	Több végrehajtási mód
Tradicionális (TPMa)	Fix	Nem lehetséges	Lehetséges
Agilis (APMa)	Flexibilis	Nem lehetséges	Nem lehetséges
Hibrid (HPMa)	Flexibilis	Lehetséges	Lehetséges

Az agilis és hibrid projektek jellemzője, hogy az ügyfeleket bevonják a fejlesztési folyamatba, ezáltal nő a felmerülő igényekkel való megbirkózás képessége, azonban ez adaptív és rugalmas gondolkodást igényel a projektmenedzsmentben is. Az agilis projektmenedzsment megközelítésben a projektek lebonyolítása rugalmasabb, a projektstruktúra alkalmazkodni tud a változó ügyféligényekhez (Dingsøyr és tsai., 2012). A különböző projektmenedzsment-megközelítések összefoglalása a 1. táblázatában látható a konfigurációs lehetőségek szempontjaiból.

A fent leírt menedzsment megközelítések ALM kontextusban való megvalósíthatósága és a hatékonyság összehasonlítása még nem került kiértékelésre, így egy ilyen összehasonlítás nagyban hozzájárul a szakirodalomhoz és az üzleti érdekeltek támogatásához is.

2.3 Erőforrás-korlátos projekt ütemezési probléma

A gazdasági folyamatok matematikai modellezésének alkalmazása a közelmúltban jelentősen hozzájárult a módszertani eszközök fejlődéséhez. Az 1950-es évek óta az erőforrás-korlátos projektütemezési problémát (RCPS⁴) alaposan tanulmányozták a projekttervezés területén. Ez a klasszikus probléma magában foglalja a tevékenységek ütemezését, figyelembe véve mind a prioritást, mind az erőforrás-korlátokat, egy olyan célfüggvény optimalizálása érdekében, mint például a projekt teljes időtartamának vagy a teljes költségek minimalizálása. Az évek során számos kutató dolgozott ki pontos és heurisztikus megoldásokat erre a problémára (lásd: Moukrim és tsai. (2015), Kreter és tsai. (2018), Tritschler és tsai. (2017), Abdolshah (2014), Demeulemeester és Herroelen (2006)) különféle megközelítéseket és kiterjesztéseket is megvizsgáltak. Munkájukban Hartmann és Briskorn (2021) átfogó áttekintést adnak és osztályozzák az RCPSP legjelentősebb bővítményeit.

⁴Angol rövidítés a Resource Constrained Project Scheduling Problem kifejezésből.

Az ALM konkrét esetére azonban még nem áll rendelkezésre kvantitatív módszertani elemzés és értékelés az ütemezési teljesítményre vonatkozóan. Ezért a disszertáció egyik fő lépéseként egy ilyen ütemezési megvalósíthatóság és teljesítményértékelés a célja, amely elindíthatja a kutatást akár egy lehetséges ALM-specifikus bővítéshez is az RCPSP számára.

2.4 Kockázatkezelés ALM környezetben

A KK3 kérdésköréhez kapcsolódóan, az ALM-hez kapcsolódó kockázatértékelés különösen fontos és érdekes lehet nem csak az akadémikusok, hanem az üzleti élet szereplői és a terület szakértői számára is, mivel a kockázat megvalósulása általában az életciklus során jelentkező negatív hatásokhoz kapcsolódik. Ezért minél felkészültebb a projektmenedzser egy ilyen helyzetben, annál jobban kezelheti azokat, ami azt jelenti, hogy a kockázatok korai szakaszban történő felismerése és mérséklése potenciálisan kevesebb vagy kisebb hatással lehet a meglévő megvalósítási terüinkre, illetve csökkentheti a potenciális túlköltekezés veszélyét is. Mivel az alkalmazások életciklus-menedzsmentje még az informatikai területen is sajátos terület, több tényező relevanciáját is meg kell vizsgálni, illetve az ALM-specifikus kockázatokot is azonosítani kell. Ez a kutatás összefoglalót ad a projektspecifikus és az ALM-specifikus kockázati tényezők megállapítására és azok összehasonlításához mind elméleti síkon, mind a szimulációból származó eredményekből. A 4. (Eredmények) fejezetben található 3. táblázatában azonosítom és leírom azokat a főbb faktorokat, amelyeket az ALM terület vizsgálata során relevánsnak találtam az irodalmi áttekintés során.

2.5 Kutatási feltételezések

A kutatási kérdéseket a szakirodalom alapos átvizsgálása után az itt szereplő három kutatási feltételezés kialakítása követte.

KF1: Azonosítható egy olyan tervezési modell, amely a különböző, egyesített szakirodalmi ALM definíciókon alapszik.

KF2: A projektmenedzsment megközelítésekhez (TPM, APM, HPM) kapcsolódó mátrixos tervezési módszer kiterjeszthető, amellyel megoldható az ALM ütemezési probléma. Egy szimulációs keretrendszer elkészíthető a rugalmas függőségek és nem tervezett feladatok kezelésére.

KF3: Léteznek projekthez kapcsolódó kockázati tényezők, amelyek kiterjeszthetők az ALM ütemezési problémákra, habár a projekt és az ALM hatókörének különbségei miatt ALM-specifikus kockázatok is megjelennek, amelyek hatással lehetnek az erőforrásokra, költségekre, időzítésre és befolyásolhatják a megvalósíthatóságot, valamint az ütemezési teljesítményt.

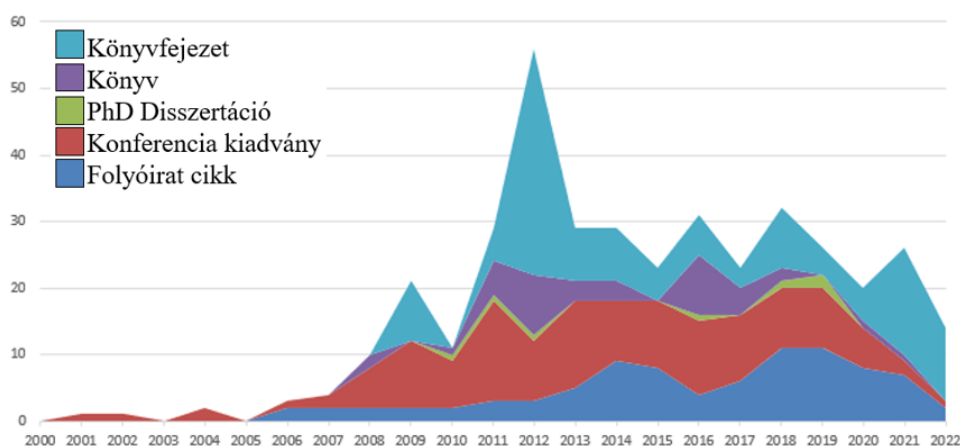
A következő fejezetben a kutatási feltételezések igazolására tett erőfeszítések és a kutatás tézisei kerülnek ismertetésre.

3 Eredmények

Ebben a fejezetben formailag a kutatási kérdés után az kutatás téziséig vezető út tömörített formája és az ahhoz tartozó eredmények kerülnek bemutatásra.

KK1: *Hogyan azonosítható a rendelkezésre álló tudományos irodalom alapján egy tervezési modell, amely az alkalmazás életciklus-menedzsmentet (ALM) reprezentálja, és amely felhasználható ütemezési feladatokhoz?*

A kutatásomban a rendelkezésre álló ALM-specifikus tudományos irodalom alapos, szisztematikus szakirodalmi áttekintés (SLR) módszerével vizsgáltam az ALM definíciójának meglétét és tulajdonságait, hogy azt, mint bemeneti tényezőt tudjam értékelni egy módszertani modell alkalmazhatóságához. A kutatás támogatására a PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analysis) módszert alkalmaztam (Page és tsai., 2021). A Google Scholar keresőmotor angol nyelvű "ALM definíció" kulcsszókereső találatait vizsgáltam. Ezután egy kritikai elemzést folytattam a fellelt ALM definíciók mélyebb feltárásának és értelmezésének érdekében. A keresés összesen 3230 forrást vizsgált meg, majd szűkítette le 76 relevánsra, amelyek explicit definíciót tartalmaztak. Mivel az ALM egy, a különböző szoftver beszállítók által erősen befolyásolt terület, ezért a fellelt eredmények egy része nagymértékben támaszkodik nem akadémiai forrásokra is. A disszertáció szintjéhez elvárt és a tudományos kutatás érdekében ez a tanulmány kizárólag minőségi, lektorált forrásokra korlátozódik, mint például folyóiratcikkek, konferencia-kiadványok, könyvek és könyvfejezetek, PhD értekezések.



2. ábra. Cikkek, konferenciaelőadások, értekezések, könyvek és könyv fejezetek éves megoszlása

A források a megjelenés éve szerinti megoszlásának áttekintése a 2. ábrán látható. Az ábra jól mutatja, hogy az ALM terület összességében jelenleg egy szűk, fejlődő terület a szakirodalomban, a konferencia-előadások és cikkek domináns száma azt mutatja, hogy intenzív egyeztetések folynak a területen.

A források minőségi szempontú elemzése után látható volt, hogy még a lektorált források között is észlelhető minőségi különbségek vannak, így két fő kategória lett kijelölve. Az *Élvonalbeli akadémiai* források, amelyek a tudományos folyóiratok felső kvadránsait reprezentálják a Scimago-rangsor szerint (Q1 & Q2), valamint a Qualis rangsoron alapuló konferenciák felső felét (A1 & A2 & B1). A többi, még mindig lektorált minőségi forrás a *Kiterjesztett akadémiai* forrásokba került rangsorba ebben az áttekintésben, ideértve a folyóiratok Q3 & Q4 cikkeit, konferencia kiadványokat, PhD értekezéseket, könyveket és könyvfejezeteket.

A szűrések és a teljes átolvasás után a források csak nagyon korlátozottan tartalmaznak információt kifejezetten az ALM definícióról, összesen csupán 76 releváns forrásban volt explicit definíció jelen. Ez meglehetősen alacsony szám, és a szerzők több esetben kiemelik az üzleti vonatkozású erős kapcsolatokat és a gyártó-specifikus ALM-definíciós változásokkal való összefüggéseket. Ezen definíciókon aztán egy kritikai elemzést végeztem el és végül osztályoztam őket. Az elemzés után a talált ALM definíciókat 7 kategóriába osztottam, amelyeket a 2. táblázat foglalja össze, ez tartalmazza az azonosított ALM definíciók hatókörét a legfontosabb kiemelésével bal oldalon, valamint a fő forrásokat hivatkozásokkal (saját fordítás angol nyelvről) és tartalmát a táblázat jobb oldalán.

Érdekes eredmény, hogy míg az *Élvonalbeli akadémiai* forrásokban a leggyakrabban az E) definíció szerepelt, amely az ALM-et úgy tekinti, *mint paradigmát, amely irányítást, fejlesztést, üzemeltetést/karbantartást tartalmaz*, kiemelve az absztrakció magas szintjét. A *Kiterjesztett akadémiai* forrásokban a C) definíció volt a legelterjedtebb, amely az *ALM-re, mint a tevékenységek koordinációs-keretére utal* (beleértve a követelményeket, a modellezést, a fejlesztést, az összeállítást és a tesztelést) a *munkatermékek* (a tevékenységek összekapcsolására szolgáló folyamatok érvényesítése; a kapcsolatok és kapcsolatok kezelése a tevékenységek között) fejlesztési munkatermékek, valamint a fejlesztés előrehaladásáról szóló jelentés) az SW életciklusa során. Emellett a szisztematikus szakirodalmi áttekintést követően kritikai elemzést folytattam az egységes definíció létrehozására. Ez a megközelítés a definíciók kombinált és finomított szintézisét adja, amellyel egy egységesített, a módszertani kutatáshoz széles alapot nyújtó komplex definíciót készítettem.

ALM értelmezése mint...	Definíciók az akadémiai irodalomból
A) egy folyamat a SW életciklusra	A termék életciklus menedzsment (PLM) és megfelelője a szoftverben, nevezetesen az alkalmazás életciklus-menedzsment (ALM) az az átfogó üzleti folyamat, amely a terméket vagy szolgáltatást a kezdetektől az életciklusának végéig irányítja a lehető legnagyobb üzleti érték elérésének érdekében. Mind a vállalkozás, mind annak ügyfelei számára. (Lachainer, 2011) A PLM/ALM egyesíti a folyamatokat, az erőforrásokat és az eszközöket a termékek hatékony megvalósításához, a kezdetektől a szolgáltatás végéig. Ez magában foglalja a szakértők implicit tudását az eljárásokban, folyamatokban és az eszközökben rögzített explicit tudást is. A PLM/ALM a hozzáértéstől (know-how) a tárgyi tudásig (know-what) és a megértésig (know-why) terjed. (Ebert, 2013), (Gatrell, 2016)
B) SW fejlesztés és karbantartás	Az Alkalmazás Életciklus-Menedzsment (ALM), a szoftverfejlesztés és -karbantartás széles körben használt életciklusa. (J. Rossberg, 2012), (Ramler, 2012)
C) dokumentum kezelő eszköz a SW életciklusa során	Az ALM azért jelent meg, hogy koordinálja a tevékenységeket és a műtermékeket (pl. követelmények, forráskód, tesztesetek) a szoftvertermékek életciklusa során. (Kääriäinen és Välimäki, 2009), Gatrell (2016) A fejlesztési életciklus-tevékenységek koordinálása, beleértve a követelményeket, a modellezést, a fejlesztést, az összeállítást és a tesztelést, a következők révén: 1. az e tevékenységeket összekapcsoló folyamatok érvényesítése; 2. az e tevékenységek által felhasznált vagy generált fejlesztési műtermékek közötti kapcsolatok kezelése; 3. jelentés a fejlesztési erőfeszítések egészének előrehaladásáról. Az ALM-et gyakran olyan keretrendszernek tekintik, amelynek célja az összes életciklus-tevékenység szinkronizálása ahelyett, hogy egy adott életciklus-tevékenységre összpontosítana. (Schwaber 2006)
D) a SW életciklusának kibővített verziója átadás utáni fázisokkal	Az ALM a számítógépes programok termékéletciklus-kezelése, amely szélesebb körű megközelítés, mint az SDLC, amely a tipikus szoftverfejlesztési szakaszokra korlátozódik. Ezzel szemben az ALM a fejlesztési életciklus utáni szakaszokat is meghatározza. (OGC ITIL, 2002), (Arya, 2011), (Chappell, 2011)
E) paradigma: irányítás, fejlesztés, üzemeltetés/karbantartás	Az Alkalmazás Életciklus Menedzsment (ALM) a legújabb paradigma a szoftvertermékek irányításával, fejlesztésével és karbantartásával kapcsolatos különféle tevékenységek integrálására és menedzselésére. Az ALM három funkció kombinációja: irányítás, fejlesztés és üzemeltetés, valamint három mérföldkő: (az ötletelés kezdete), a telepítés és az élettartam vége. (Chappell, 2008) (Rossberg, 2014)
F) egy szolgáltatás a fejlesztés lezárása után	Az alkalmazásmenedzsment az operatív alkalmazásrendszerek problémamegoldási folyamatának életciklus-orientált vezérlését jelenti, kivéve az alapvető alkalmazásfejlesztési szolgáltatásokat. (Kueper, 2011)
G) minőségbiztosítási rendszer	A szabványos fejlesztéstől a szoftverkiadásig terjedő munkafolyamat létrehozása, amelyet gyakran ALM-folyamatnak is neveznek, különösen kritikus fontosságú a szervezetek számára, hogy megfeleljenek a szigorú IT-megfelelőségi kötelezettségeknek. (Tracy, 2006)

2. táblázat. Összegző táblázat az ALM definíciókról (saját fordítás az eredeti angol szövegezésből)

A következőkben ennek az egyesített ALM definíciónak az elemeit írom le, a sor elején kiemelve a kulcsszavak találhatóak.

1. Az ALM hatásköre. Az ALM egy holisztikus megközelítés a szoftveralkalmazások kezelésére, azok teljes életciklusa során, a kezdetektől azok kivonásáig. Különféle tevékenységek és munkatermékek integrálásával és menedzselésével valósul meg 3 funkcióval, nevezetesen az irányítás, fejlesztés és üzemeltetés, ahova beleértjük a karbantartást is. Az irányítás egy átfogó tevékenység az ALM teljes élettartama alatt; jelentősége azonban nagyobb a folyamat elején, a befolyásoló tényezői miatt. A fejlesztés leginkább a klasszikus szoftverfejlesztési megközelítéshez kapcsolódik, amely a főbb K+F tevékenységet

foglalja magába. Az üzemeltetés és a karbantartás meglehetősen hasonlít egy szolgáltatáshoz, mivel azonban ebben a fázisban a hibajavítás mellett további fejlesztési feladatok is előfordulhatnak, különböző méretekből, ezen feladatok megjelenése és kezelése egyedivé teszi az ALM-et.

2. *Az ALM fázisai.* Az ALM-nek három fő mérföldköve van: ötletelés, üzembehelyezés és az élettartam vége. 7 fázisból áll, amelyek magukban foglalják a követelmények összegyűjtését, a tervezést, a fejlesztést, a tesztelést, a beüzemelését, a karbantartást és a szoftver kivonását.
3. *Fő komponensek.* Az ALM alapvető komponensei olyan folyamatokon és eszközökön keresztül támogatják az életciklust, mint a verziókezelés, a problémakövetés, a folyamatos integráció és a beüzemelés automatizálása. Ezek az összetevők döntő szerepet játszanak az ütemezésben és az erőforrások elosztásában.
4. *Ütemezési kihívások.* Az ALM-ben az ütemezéshez sajátos kihívások kapcsolódnak. Ide tartoznak a következők: erőforrás allokáció, mivel a tevékenység megvalósítása is többnyire véges erőforrásokhoz kötődik; feladatok végrehajtási sorrendje, mivel a fejlesztési és karbantartási szakaszban a tevékenységek ütemezését másként kell kezelni; az erőforrások ütemezési módszertanon alapuló időbecslése nehéz lehet és nem egyszerű; az erőforrás-kihasználás és ütemezés optimalizálása, mivel a jelenleg létező módszertanok még nem bizonyultak optimálisnak az ALM sajátosságaihoz.
5. *ALM fejlesztési módszertanok.* Az ALM szorosan integrálódik a szoftverfejlesztési folyamatba. Az ALM-en belüli ütemezéshez figyelembe kell venni a rugalmas szoftverfejlesztési módszereket, mint például az agilist, részben vagy teljesen hibrid megközelítésként alkalmazva. Jelenleg azonban nincsenek az ALM-hez kapcsolódó, bizonyítottan optimális módszertanok vagy keretrendszerek.
6. *Rugalmaság és alkalmazkodóképesség.* Az ALM ütemezési módszereinek rugalmasnak és adaptívnak kell lenniük, hogy alkalmazkodjanak a változó követelményekhez, a váratlan problémákhoz és a változó projektprioritásokhoz. A változások kezelésének és menedzselésének folyamata nemcsak a fejlesztés, hanem az üzemeltetési/karbantartási szakaszában is megtörténik. Ezek a további változtatási kérések az egyszerű feladatok méretétől egészen a kisebb alprojekt méreteig terjedhetnek.
7. *Teljesítmény mérés és mutatók.* Az ütemezéshez kapcsolódó kulcsfontosságú teljesítménymutatók (KPI⁵-ok) mérése és nyomon követése az ALM-ben szintén

⁵Key Performance Indicator - Kulcsfontosságú Teljesítménymutató

kulcsfontosságú a rugalmas szerkezetkezelés és a szükséges szerződéses feltételek miatt. Ezek a mérőszámok magukban foglalják a projekt időtartamát, az erőforrás-kihasználást és a feladatok elvégzésének arányát.

Az előbbieken leírt egységesített definíció elemei kitérnek tehát a feladatok köreire, beleértve a kezdetektől a szoftveralkalmazás kivonásáig. Ebben a tartományban definiálhatóak tehát mindazok a feladatok az életciklusa során, amelyek valamilyen idő és erőforrás szükséglettel, minőségi paraméterrel bírnak, és logikai összefüggés is lehet közöttük a végrehajtási sorrend meghatározása érdekében. Ezen tulajdonságaik megfelelően reprezentálhatóak az úgynevezett mátrixos tervezési módszerrel, amely a 4. ábrán látható, tehát ez a modell tehát megfelelő az ALM specifikus probléma leírására.

A **KK1** kutatási kérdés az elérhető ALM-definíciókra és jellemzőikre összpontosított, valamint arra, hogy egy közös definíció miként nyújthat alapot a jövőbeli kutatásokhoz. A fellelt definíciókat és azok összefoglalását bemutattam, összesen 7 különböző értelmezést tükrözve. A definíciók lefedik az alkalmazások életciklusmenedzsment hatóköreinek és tartalmának jelenlegi akadémiai megértését, és megerősítik a mátrixos tervezési modell alkalmazásának lehetőségét, mindezt az egységes ALM-definíció felhasználásával, a jövőbeni módszertani kutatásokhoz. Ezen megállapítások megfelelnek a **KF1** elvárásainak, következésképpen a **KF1** elfogadható.

KK2: *A jelenlegi projektmenedzsment ütemezési módszertanok (TPM, APM, HPM) eredményeznek-e megvalósítható megoldásokat az ALM környezetben, és azok hogyan teljesítenek?*

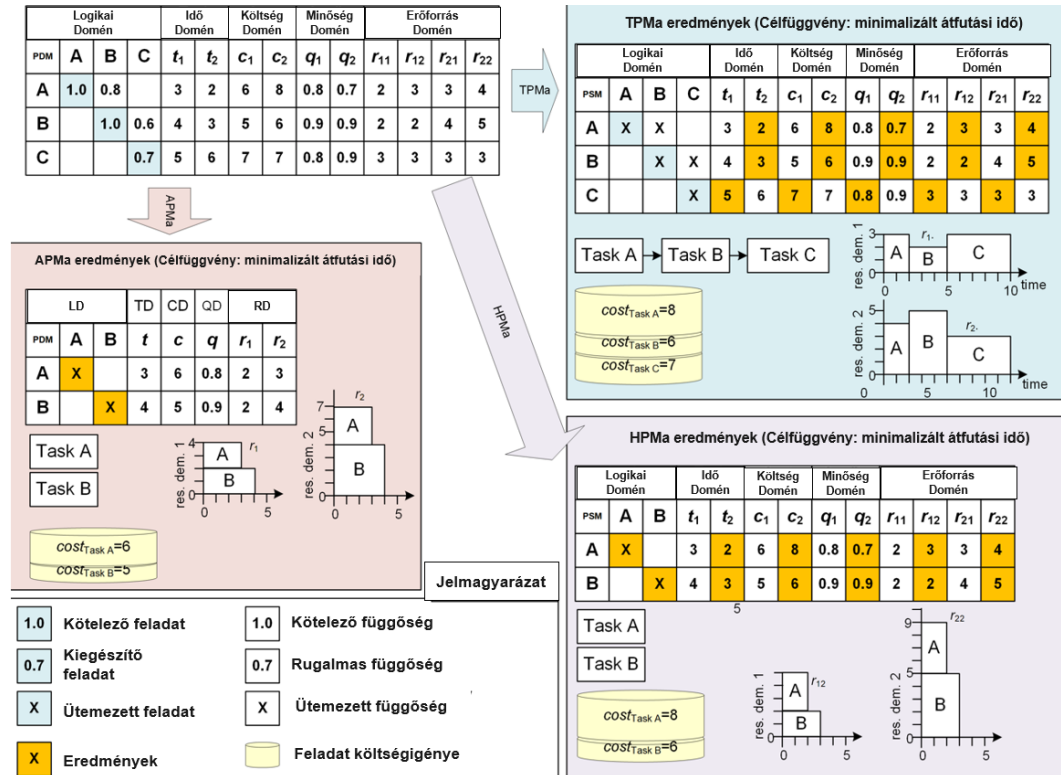
A kérdés vizsgálatára egy szimulációs keretrendszerrel állítottam fel, amely a különböző projektmenedzsment megközelítéseket valósította meg, így az eredmények jól mérhetőek és összehasonlíthatóak lettek. Ennek az összefoglaló leírása következik ebben a szekcióban.

Ágens alapú projektmenedzsment megközelítések megvalósítása

A TPM, APM és HPM esetében egy ágens⁶ került megvalósításra az ütemezési problémamegoldás végrehajtására. Ennek megfelelően tradicionális projektmenedzsment ágensnek (TPMa) hívják, az agilis megközelítésben az ágens az APMa, a hibrid megközelítésnél pedig a HPMa, lásd a 3. ábrát az ügynökök ütemezési problémákkal kapcsolatos viselkedésének összehasonlításához.

A 3. ábra bal felső sarkában látható a projekt domain mátrix (PDM) nevű struktúrában való reprezentáció, amely 3 kötelező tartományt tartalmaz, nevezetesen logikai- (LD, Logic Domain), idő- (TD, Time Domain) és költség domént (CD, Cost

⁶Egy programozott szoftver-ügynök, amely egy specifikus feladatot hajt végre.



3. ábra. Projektmenedzsment megközelítések és ágens alapú megvalósításaik összehasonlítása, ha a célfüggvény a minimális teljes projektidő. (t_j, c_j, q_j jelöli az idő/költség/erőforráskérés/minőségi paramétereket, valamint, a végrehajtási módjait a j , r_{ij} az i erőforrás j végrehajtási módjában.)

Domain), valamint két kiegészítő tartományt, a minőség- (QD, Quality Domain) és erőforrás domént (RD, Resource Domain) (Kosztján, 2022).

Szimulációs keretrendszer

Az ütemezési problémák különböző ágensekkel (TPMa, APMa, HPMa) való megvalósíthatóságának meghatározására tehát egy szimuláció készült, amely a probléma megoldására összpontosított. Amennyiben a probléma megoldható volt, vagyis a megjelenő kockázatok nem hátráltatták a megoldást, a megoldás „túlélte” ezt a kihívásokkal teli, kockázatos környezetet. Ehhez Monte Carlo Szimulációs módszert (MCS) választottam, amely az egyik leggyakrabban alkalmazott kockázatkezelési módszer. A módszer jól bevált technika a projekt kockázatainak és bizonytalanságainak szimulálására. Az MCS-ben a kockázati hatások, például a késések, a költség-túllépések és a túlmunka is jól szimulálhatók a (Kwak és Ingall, 2007) feladatok idő-/költség-/erőforrásigényének megváltoztatásával. Az MCS-ben a feladatigények elméleti vagy empirikus eloszlásokat követnek. Az MCS-t mátrix alapú technikákkal kombinálva a kockázatok egymásra utaltsága is modellezhető. Rugalmas projektstruktúrák esetén a projekt átstrukturálható (Kosztján és Szalkai, 2018, 2020), és ez a bővítés kulcsfontosságú a rugalmasság kezeléséhez, például az agilis és hibrid projekteknél. A szimuláció során a megvalósíthatóság értékelése megtörténik, amint a

projekt túléli a kockázatokat, és megvalósítható projektstruktúrát eredményez, vagy meghíúsul és megvalósíthatatlan projektté válik.

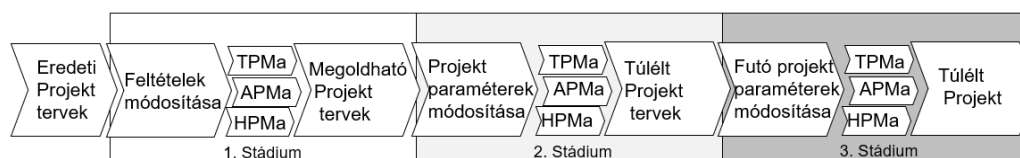
Adatbázisok

A megfelelő projekttervek projektadatbázisból való kiválasztása nehézkes volt, mivel egyik ismert projektgenerátor (például ProGen (Kolisch és Sprecher, 1997), RanGen I (Demeulemeester, Vanhoucke és tsai., 2003) és II (Vanhoucke és tsai., 2008)) vagy projektadatforrás (pl. mint MMLIB (Peteghem és Vanhoucke, 2014) és PSPLIB (Kolisch és Sprecher, 1997)) sem különbözteti meg a kötelező és kiegészítő feladatokat, sem kezeli a rugalmas függőségeket. Ezért nincsenek a feladat befejezéséhez vagy a feladat függőségeihez kapcsolódó pontszámértékek (score) sem megadva. Ennek ellenére a rugalmas függőségek és feladatvégrehajtási prioritások figyelembevétele nélkül a rugalmas projekttervek nem modellezhetők, mert az alacsonyabb prioritású (kiegészítő) feladatok nem hagyhatóak el, a projektterv nem alakítható át. Mivel még mindig nem létezik olyan elérhető, valódi projektadatbázis, amely a prioritások empirikus eloszlását vagy a rugalmas függőségeket tartalmazná, ezért a feladatok/függőségek és prioritások kiválasztása normáelosztást követett ebben a megvalósításban. Ezen kívül, specifikusan ALM struktúrát leíró adatbázis sem létezik még, ezért mivel az ALM tulajdonképpen egy szoftver applikáció megvalósítására összpontosít, így az IT projektek kapcsolódnak a legszorosabban az elérhető kategóriákból, így most ezeket a projekteket elfogadjuk mint bemenetet. Három különböző adatkészlet lett felhasználva, ezek a következők:

"A" Adatkészlet: az adatkészlet kiválasztott adatokat tartalmaz az ismert projektadatbázisokból: PSPLIB (j30 adatkészlet) és MMLIB (MMLIB50 adatkészlet). Az adatbázis-kiválasztás a meghatározott szempontok alapján történt, beleértve a tevékenységek számát és a soros/párhuzamos mutatókat, amelyek értékei a legjobban illeszkednek az IT szektor projektjeihez.

"B" Adatkészlet: a meglévő szabványos adatkészletekből kiválasztott példányok mellett a projektpéldány-generátorok által létrehozott projektadatok a másik rész forrása, a széles körben elfogadott ProGen (Kolisch és Sprecher, 1997) generátort használva.

"C" Adatkészlet: a Batselier és Vanhoucke, 2015 által bemutatott adatbázis empirikus projektadataiból áll az IT területről.



4. ábra. A szimulációs környezet stádiumainak leírása

Az 4. ábra a szimulációs keretrendszert mutatja. Ebben a szimulációban a kockázati hatások, például a korlátok módosítása, lásd az első szakaszt (1. Stádium) és

a költség- és időtúllépések, második és harmadik szakasz (2. és 3. Stádium) hatását mérséklék a projektmenedzsment ügynökei, azaz az ágensek. A megvalósítható projektek további tulajdonságait a projektmenedzsment ágensek kezelik.

Leíró statisztikák

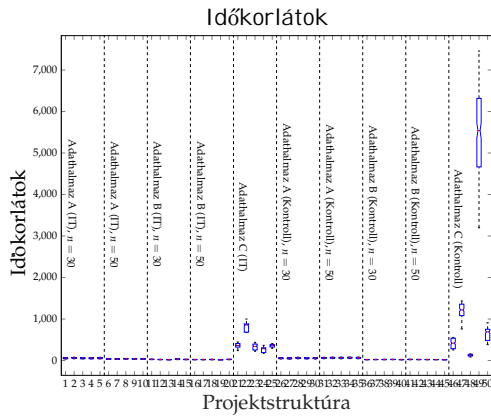
A 5. ábrán látható szimulációs eredmények 48 000 ütemezési probléma leíró statisztikáit mutatják, amelyek 50 projektstruktúra halmazán alapulnak. Az 1-25-ig terjedő projektstruktúrák generált és valós informatikai projektekből álltak, a kontrollcsoportok (26-50) pedig az építési projektstruktúrákat követték. Mivel a feladatok befejezésének és a feladatok közötti függőségek 0-50%-a rugalmas, a megszorításokat minden egyes ütemezési problémához egyedileg lettek kiszámolva. A 5. ábra az idő-, költség-, minőség-, pontszám- és erőforrás-korlátokat mutatja projektstruktúrák és rugalmassági paraméterek szerint. A korlátok a projektigények elméleti tartományának $\frac{1}{3}$ és $\frac{2}{3}$ arányában vannak megadva. Ezek a korlátozások minden PManál azonosak voltak, ezért összehasonlíthatók. A korlátok megadása azonban illeszkedik a projekttervek lehetőségeihez. Ezért megbizonyosodhatunk arról, hogy a C adatkészlet valódi projektjei nagyobb idő- és költségigényekkel rendelkeznek (lásd a 5(a,c) ábrán a 21-25 és 46-50 projektstruktúrákat). Ebben az esetben a minőségi követelmények is magasabbak (lásd a 5(e) ábrán a 21-25 és 46-50 projektstruktúrákat). Másrészt a generált projektek (a B adatkészletből) rendelkeznek a legnagyobb erőforrásigénnyel (lásd a 15-20., 35-20. projektstruktúrákat a 5(g) ábrán). A MANOVA klaszterből csak egy projektstruktúra (49) mutat releváns különbséget a megszorításokban (összevetve a 5(a) és 5(c) és 5(k) részábrákat).

Megvalósíthatóság kontra rugalmasság

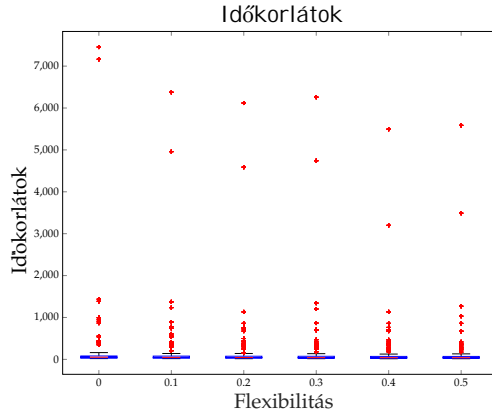
A 6(a) ábra a projektmenedzsment ágensek megvalósíthatósági arányát mutatja szakaszonként és rugalmasságonként. A túlélési arány megadja az adott szakaszban a TPMa, APMa vagy HPMa által kezelt megvalósítható projektütemezési problémák arányát. Szakaszról szakaszra egyre kevesebb projekt éli túl a korlátok változását (1. szakasz), az igények és struktúrák változásait a tervezési szakaszban (2. szakasz) és a követési szakaszban (3. szakasz). Különösen a 3. szakaszban (lásd a 6(b) ábrát) a TPMa (pirossal jelölve) érzékenyebb az igények változásaira, míg a rugalmas megközelítések általában kevésbé érzékenyek (lásd a 6(b) ábrát), még akkor is, ha a rugalmassági arány magas (lásd a 6(a) ábrát).

A 5(d,f,h,j) ábrával összhangban a 6(a) ábra azt mutatja, hogy általában a rugalmasság növekedése növeli az összes megközelítés megvalósíthatósági arányát. Ezt a lehetőséget azonban elsősorban agilis és hibrid megközelítésekkel lehet kihasználni. Ezenkívül alacsonyabb rugalmasság esetén (< 20%) a TPMa több megvalósítható projektet ad, mint az APMa (lásd a 6(a) ábrát).

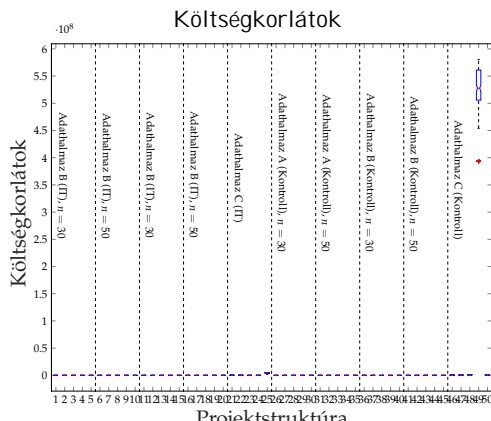
Érdekes eredmény, hogy a HPMa jobban kihasználta a rugalmasság adta lehetőségeket. A HPMa megvalósíthatóbb projekteket tesz lehetővé, mint az agilis megközelítés.



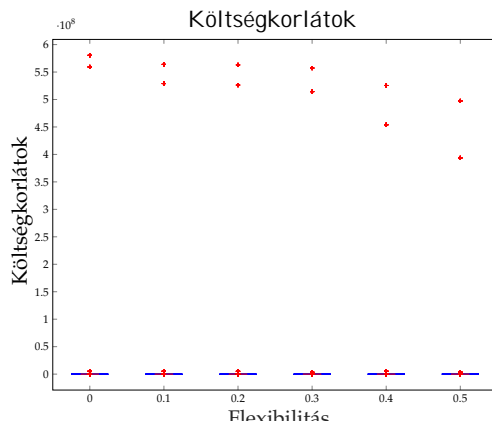
(a) $p = 0$



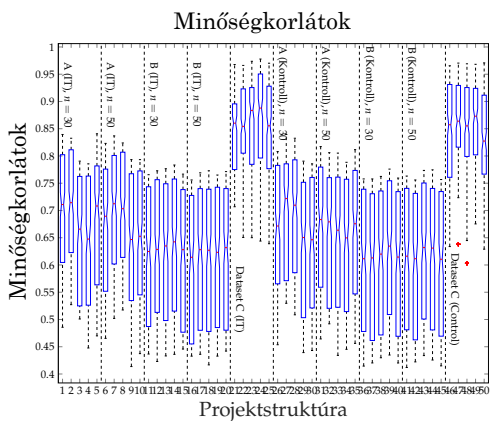
(b) $p = 0.0070$



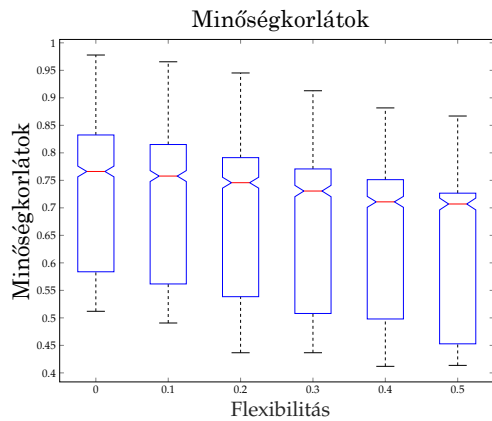
(c) $p = 0$



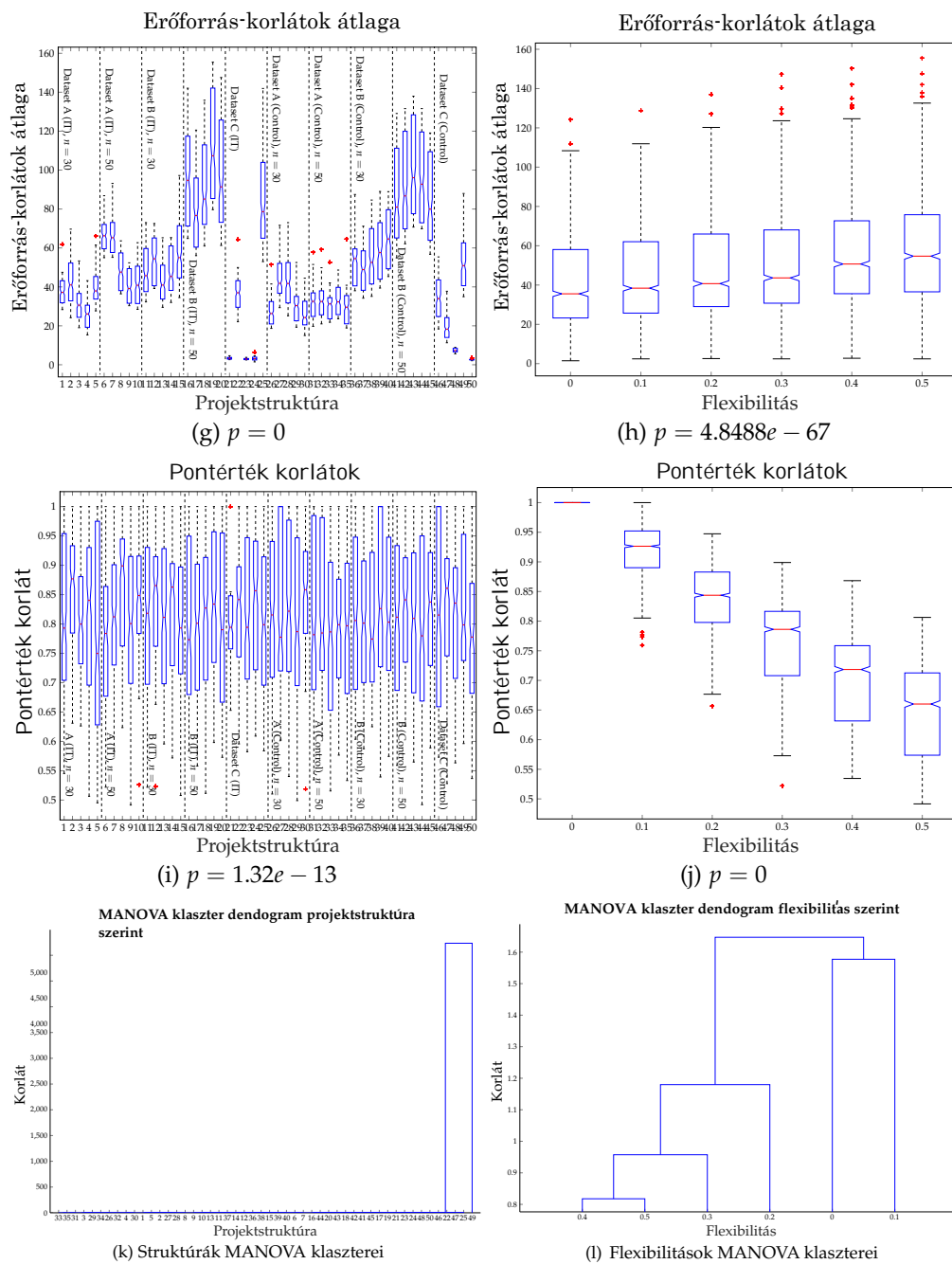
(d) $p = 0.9358$



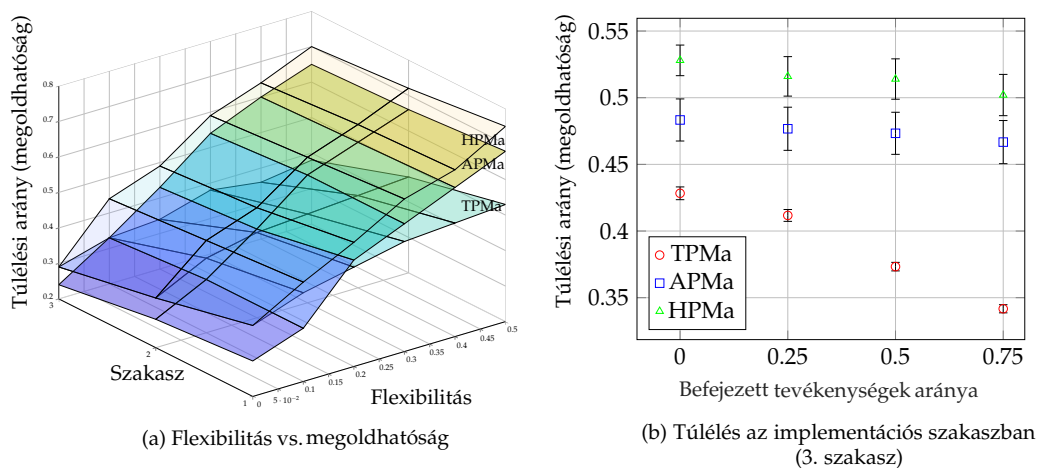
(e) $p = 0$



(f) $p = 2.8632e - 159$



5. ábra. A (M)ANOVA eredményei a korlátokra, a projektstruktúrára és a rugalmasságra vonatkozóan



6. ábra. Projektmenedzsment ügynökök megvalósíthatósági aránya a rugalmasság alapján

Összefoglalva, a **KK2** kutatási kérdés a projektmenedzsment-megközelítések megvalósíthatóságára és teljesítményére összpontosított az ALM hatókörében. A szimulációk a **KF2** alapján az IT területen több rugalmassági tényező és különböző célfüggvények alapján mutatták be a megoldások megvalósíthatóságát és érzékenységet. Ennek eredményeként a **KF2** elfogadásra kerülhet.

KK3: *Melyek az ütemezési probléma kockázati tényezői az ALM-környezetben, és ezek hogyan befolyásolják a megvalósíthatóságot és az ütemezési teljesítményt?*

A kockázatot az eredményre vonatkozó bizonyosság hiányaként jellemezzük, amely lehet pozitív változás vagy negatív fenyegetés. A hatékony kockázatkezelés magában foglalja az olyan veszélyek felismerésének és szabályozásának folyamatát, amelyek akadályozhatják a szervezet üzleti céljainak elérését (Government Commerce, 2007). A kockázatkezelés a projektmenedzsmentben már megszokott feladat, a kockázatkezelés mögött az ISO/IEC 31000 (Barafort és tsai., 2019) szabványban is meghatározott potenciális termékproblémák azonosítása, értékelése, elemzése, értékelése és mérséklése áll. A kockázatkezelés a teljes termékéletciklus-folyamatot végig kíséri. A kockázatot a fenyegetésekkel való kapcsolata miatt általában kerülendő dolognak érzékeljük, és ahogy korábban látható volt, az ALM környezet a projekt hatóköréhez képest kibővült, így további teret enged a potenciális kockázati tényezők megjelenésének. Sajnos az ALM kockázati tényezőit még alig kutatták, a szakirodalom elsősorban leszűkített ALM hatóköröket tartalmazza. A következőkben azokat az ALM környezettel kapcsolatos kockázatok kerülnek bemutatásra, amelyek a rendelkezésre álló szakirodalom alapján strukturális és *ütemezési* szempontból azonosíthatók. Ez azt jelenti, figyelmen kívül hagyom a számos gyakrabban referált, de kontextusa miatt különböző, ALM bevezetéssel kapcsolatos kockázat leírását, például az ALM szervezetváltási szempontból releváns kockázatokat, habár

ehhez a szakirodalomban már azonosítottak eljárásokat (Akgun és tsai., 2020; Tüzün és tsai., 2019), majd később a működéshez kapcsolódóan is jelentős kockázati tényezőket írtak le a szakirodalomban (Cheng, 2010).

A kockázatkezelési megközelítések eltérőek -az ALM-környezetben is gyakran használt- agilis megközelítések esetében, mivel az iteratív visszacsatolással rendelkező agilis ideológia célja, hogy „korán megbukjon”⁷ a fejlesztés a belső teszteléseiben, ennél fogva időben tudjon reagálni a felmerülő problémákra, a fejlesztés sikeres lezárásáig. Bugarová és Šimíčková (2019) elemzést készítettek a hagyományos és az agilis kockázatkezelés összehasonlítására, és kiemelik mindkét oldal előnyeit és hátrányait. Rámutatnak arra, hogy a szervezetek projektszervezeteket használnak az új termékek fejlesztéséhez és bevezetéséhez szükséges változások kezelésére. A mai versenykörnyezetben csak az jár sikerrel, aki képes kezelni a kockázatokat és hatékonyabban tudja megvalósítani a projektet.

Az ALM és PM hatókör fentebb tárgyalt különbségei miatt a kockázati körök is további elemzést igényelnek. A projekt- és szoftverszintű kockázatok elméleti kiterjesztésre szorulnak az ALM hatókörére is. Ezen a területen az akadémiai kutatások nagyon korlátozottan elérhetőek, Choetkiertikul és Sunetnanta (2012) egy kockázatgyűjtő- és értékelési eszközt javasolt, amely többnyire az elosztott szoftverfejlesztéssel kapcsolatos kockázatokra összpontosít. Azonban főleg az általános életciklus-kezelési terület kockázatkezelését (Castaneda és tsai., 2020; Hummer és tsai., 2019; Niemann és Pisla, 2018; Sonnemann és tsai., 2015) vagy a szoftverfejlesztési életciklust kutatják (Roy, 1962; Sahu és tsai., 2014). Tehát a következőkben, a modell értelmezés korlátozásaként, a releváns kockázatokat, például a projekt kockázatokat kezeljük, amelyeknek megfelelőnek és elfogadhatónak kell lenniük az ALM modellünk számára. A korlátozás a jövőben az ALM hatókörének további tanulmányozásával feloldható.

A 3. táblázatban látható a szakirodalomból kinyert eredmények a téma kutatása után, amelyek a projektmenedzsmenttel és az ALM kockázati tényezőkkel kapcsolatos faktorokat mutatják be, valamint, hogy mely kutatók vizsgálták a témákban. Ezt követően rövid betekintést adok arra vonatkozóan, hogy ezek hogyan kapcsolódnak az általános projektmegközelítésekhez, ezen belül is a SW-projektekhez, és főként az az ALM-hez. Mivel jelen disszertáció középpontjában az ütemezés és a hozzá kapcsolódó tevékenységek elemzése állt, így a fő hangsúly az ALM-re is jellemző, szoftver-fejlesztés és karbantartás specifikus kockázatok, valamint az ALM-re kifejezetten jellemző, a nem tervezett tevékenységek kidolgozásán és annak hatásain volt. Az, hogy mely kockázat van jelen mindkettőben, vagy elsősorban az ALM környezetben, az utolsó oszlopban jelöltem.

Összességében látható, hogy sok kockázati tényező már a szoftverfejlesztéssel

⁷Az angol "fail early" agilis elv fordítása.

Kockázati Faktor	Projektmen. releváns	ALM releváns	Elsősorban ALM-ben
Terjedelemlépcső	Komal és tsai. (2020) és Madhuri és tsai. (2018) Ajmal és tsai. (2022)	Aiello és Sachs (2016) és Rossberg (2019)	Nem
Változás a követelményekben	Kossmann (2016) és Venkatesh és Balani (2016)	Chanda és tsai. (2013)	Nem
Költségvetési túllépések	Albtoush és Doh (2019) és Jackson (2002)	Banjanin és Strahonja (2018) és Ebert (2013)	Nem
Ütemezésbeli késések	Majerowicz és Shinn (2016) és Park (2021)	Aiello és Sachs (2016) és Tudenhöfner (2011)	Nem
Erőforrás korlátok	Issa és Tu (2020) és Mishra (2020)	Rossberg (2019) és Rossman (2010)	Nem
A probléma megvalósíthatósága	Issa és Tu (2020) és Rahman és tsai. (2021) Beek és tsai. (2024)	Aiello és Sachs (2016)	Nem
Minőségbeli problémák	Komal és tsai. (2020), Shafqat és tsai. (2022) és Wawak és tsai. (2020)	Akgun és tsai. (2020) és Otibine és tsai. (2017)	Nem
Nyomon követhetőség hiánya	Nem	Akgun és tsai. (2020) és Corallo és tsai. (2020)	Igen
Verziókezelési problémák	Nem	Kääriäinen és Välimäki (2008) és Pirklbauer és tsai. (2009)	Igen

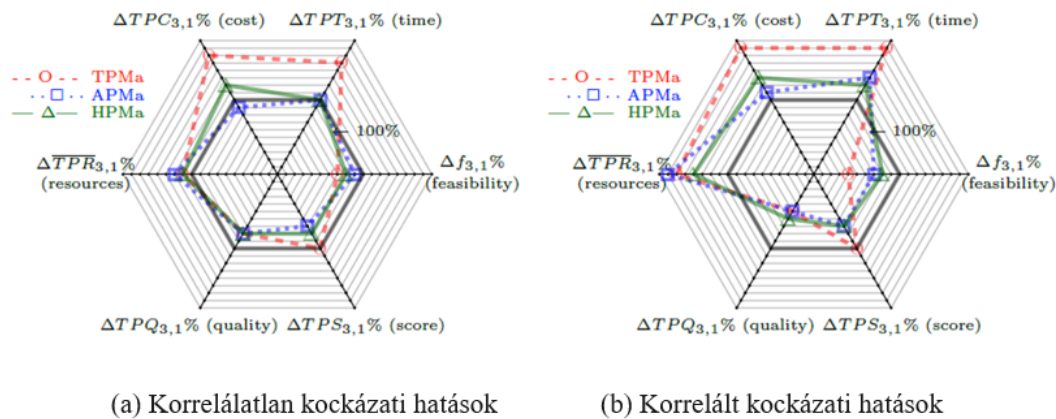
3. táblázat. Projektmenedzsment és alkalmazás-életciklus menedzsment környezetben megjelenő kockázati tényezők

kapcsolatosan is jelen volt, ezek a potenciális problémaforrások megmaradtak és értelmezésük az ALM környezetben is továbbra relevánsak. A szakirodalom szűkös forrásai az ALM specifikus kockázati tényezőkre is kitérnek, amelyek inkább jellemzőek az ALM-re, azonban számuk egyelőre nagyon limitált, további kutatásokat igényel.

A kockázatsökkentés teljesítménye

A módszertani vizsgálat során az ágensek mellett a kockázati tényezők vizsgálata is szerepelt. A vizsgált faktorok a költség, idő, erőforrás használat, megoldhatóság, minőség és a célfüggvény teljesítésének a kockázati szintje volt. A szimulációból a 7. ábra mutatja a feltárt projektmenedzsment megközelítések kockázatsökkentési

teljesítményét. Az ideális kockázatcsökkentési stratégia minden projekttervet megvalósíthatónak tart.



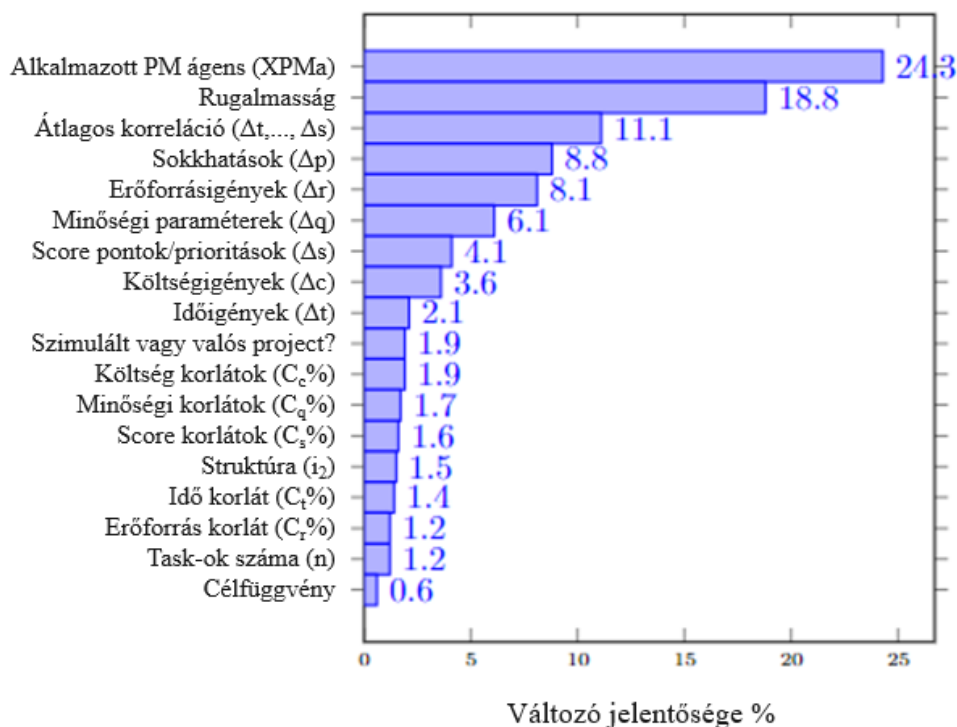
7. ábra. Projektmenedzsment megközelítések kockázatcsökkentési tevékenysége

Mivel a TPMa minden feladatot megtart, így az elégedettségi pontszáma is minden esetben maximális, de ennek a stratégiának az az ára, hogy több projekttervet "elveszít", mint más stratégiák. Ezen túlmenően, ha csak a megvalósítható projektterveket vesszük figyelembe, a TPMa mutatja a legnagyobb tendenciát a késésekre és a túl magas költségű helyzetekre. Ha a kockázati tényezők mérsékelten korrelálnak, a TPMa jelentős mennyiségű többletforrást igényel.

Az APMa egészen más képet mutat. Érdekes módon az agilis technika az egyetlen megközelítés, amely a kockázati tényezők ellenére csökkenti a projekt költségeit. Ennek a stratégiának az ára azonban az, hogy a legnagyobb minőség- és terjedelemszökkenést éri el.

Érdekes az is, hogy amikor a kockázati tényezők mérsékelten korrelálnak, akkor a kényszerű párhuzamosítás miatt ebben a stratégiában a legnagyobb mértékben nő meg az erőforrásigény. A HPMa a legtöbb projekttervet megvalósíthatóan tartja, és ez a megközelítés egyensúlyt teremt a többmódusú és átstrukturálási technikák között, ami azt jelenti, hogy ez a stratégia jól mérsékli a kockázati hatásokat annak érdekében, hogy a projektterveket a korlátok között tartsa. Mindeközben jobban megőrzi a hatókört, mint az agilis technikák.

Ha a kockázati tényezők egymással korrelálnak, akkor nagyban fokozzák egymás kockázati hatásait. A kockázati tényezők közötti kölcsönös függőség ezen hatásai különösen a TPMa alkalmazása esetén jelentkeznek. A TPMa nagyon érzékeny az idő-, költség- és erőforrásigény változásaira és ezek egymásra épülésére, ami összhangban van a szoftverprojektekben eddig szerzett tapasztalatokkal. Az agilis technikák jobban mérsékelhetik a kockázati hatásokat. Ha azonban a kockázati tényezők egymással korrelálnak, akkor a kényszerű párhuzamosítás miatt ez a technika is érzékeny az erőforrásokra.



8. ábra. Változók jelentősége a megvalósíthatóság szempontjából

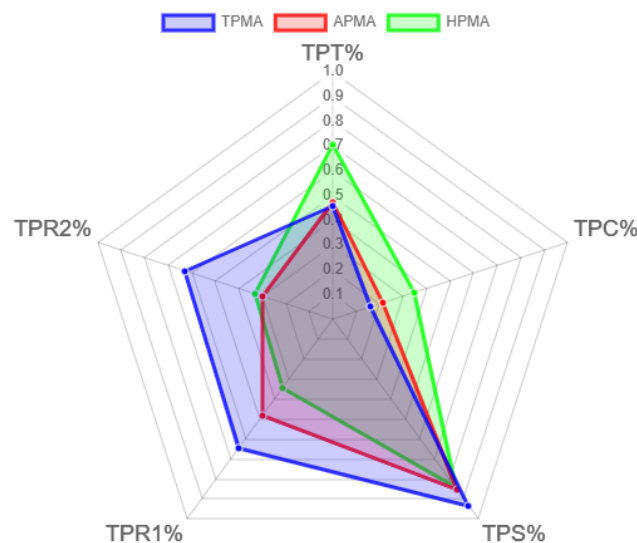
A 8. ábra a projektmenedzsment ágensek hatását mutatja. Minden feltárt szerkezeti tulajdonság, mint például a projekt szerkezete (Struktúra) és rugalmassága alacsony szintű kockázati tényezők, mint például a költségek változása (Δc), időtartam (Δt), erőforrásigény (Δr) és magas szintű kockázatok, azaz amikor a TPT, TPC, TPQ vagy TPS értékek megsértik a korlátjaikat, amiken keresztül értékeljük őket ($C_x\%$). Az eredmény szerint az alacsony szintű kiváltó okok és szerkezeti paraméterek nagyobb közvetlen hatással vannak a megvalósíthatóságra. A projekt megvalósíthatóságának fenntartásához a legfontosabb változó a kiválasztott projektmenedzsment ágens (XPMa, 24,3%). Emellett a második legfontosabb változó a rugalmassági ráta (18,8%). A kockázati tényezők közötti korreláció fontosabb (11,1%), mint maguk a kockázati tényezők. Az eredmények szerint a TPMa a legérzékenyebb azokra a sokkokra, ahol a feladatigényeknek csak néhány (azaz 10%-a) módosulnak, de ezek a változások (akár 10-szer) magasabbak.

A kockázati tényezők ($\Delta r, \dots, \Delta t$) fontosabbak, mint a megállapodásból eredő korlátok ($C_t\%, \dots, C_r\%$). Ez a megfigyelés azt bizonyítja, hogy a szerződéskötési szakaszt követően a projektmenedzsmentnek több kihívást jelent a projektterv megvalósíthatóságának biztosítása. A nagyobb kihívást az erőforrás-allokáció jelenti, mind a hagyományos, mind a rugalmas projektmenedzsment megközelítésben.

Összefoglalva, a **KK3** kutatási kérdés az ALM terület kockázati tényezőire fókuszált, és arra, hogy a kockázati tényezők hogyan befolyásolják a megvalósíthatóságot és az ütemezési teljesítményt. A szakirodalmi áttekintésből származó ALM-specifikus kockázati tényezőket kutattam és elemeztem. A szimulációba bevonható kockázati tényezőket részletesebben kiértékeltem a **KF3** alapján és megmagyaráztam az összefüggéseket. Ennek eredményeként a **KF3** elfogadásra került.

Validáció

A szimulációs eredmények alátámasztása érdekében ALM-környezetben validálás készült egy kurrens üzleti probléma feldolgozásával.



9. ábra. Radardiagram az ALM ágensek teljesítményéhez az esettanulmányban

Az esettanulmányt egy 1871-ben alapított globális autóiipari beszállítónál végeztem el, amely fékrendszerek, belső elektronika, autóbiztonsági, hajtáslánc- és alvázalkatrészek, gumiabroncsok és különféle egyéb autóalkatrészek gyártására szakosodott, piac vezető vállalat. A szervezet 58 országban működik, teljes árbevétele 33,8 milliárd euró, alkalmazottainak száma pedig megközelítőleg 190 000.

A validációs esettanulmányban az elektronikus fékrendszerek szoftveralkalmazása volt a fókuszban, ahol a vállalat csúcskategóriás beszállító, és jól ismert cégekkel versenyez. Az ALM problémakör kihívása az évtized elején jelent meg, és a cég strukturális és módszertani szempontból agilis módszertanok bevezetésével válaszolt, ami részben támogatta a kitűzött célokat. Ezért tovább vizsgálták, hogyan támogassák a kihívást jelentő folyamatos SW-frissítési kéréseket. Az ALM elnevezési konvenciót kifejezetten (még) nem használják, azonban a probléma definíciója nagyon jól illeszkedik erre az esetre. Amint az a 9. ábra radardiagrammján is látható, a szimulációs eredmények és az esettanulmányok eredményei megegyeznek és erősítik egymást.

Ágens/Célfüggvény	Idő	Költség	Pontszám	Erőforrás 1	Erőforrás 2
TPMA	Legrosszabb	Legrosszabb	Legjobb	Legjobb	Legjobb
APMA	Közepes	Közepes	Közepes	Közepes	Legrosszabb
HPMA	Legjobb	Legjobb	Legrosszabb	Legrosszabb	Közepes

4. táblázat. Összefoglaló táblázat az ALM ágensek teljesítményéről

A 4. táblázatban látható összefoglaló, amely utal az esettanulmányok eredményeire. Ezen eredmények és észrevételek bemutatásra és megvitatásra kerültek a vállalatvezetéssel és a szervezetfejlesztési projektek szakértőivel, megerősítve, hogy hibrid módszer alkalmazásánál maradnak és nem vezetnek be a teljesen agilis módszertan alkalmazását a cégnél. Az ábráról jól látható, hogy a hibrid és agilis módszerek jobban szerepeltek mint a tradicionális projektmenedzsment megközelítés amit eredetileg is használt a cég. Számukra a cél a határidők és a költségkeret megtartása volt, ezért is szerepel ez a táblázat elején. A táblázatban található pontszám a teljesítés tartalmára vonatkozik, a hibrid módszertan esetében a legrosszabb értékelés ugyanakkor nem volt súlyos hatással a vevői elégedettség tekintetében, hiszen az agilis módszertan megértése és használata a vevői oldalon is megtörtént, így a tartalmi egyeztetések könnyebbé váltak. Az erőforrások használatán még optimalizálni igyekszik a vállalat, az új módszertan bevezetése után javulást várnak a begyakorlási görbe mentén.

4 A kutatás tézisei

A kutatási kérdéseket, feltételezéseket, valamint a hozzájuk kapcsolódó eredményeket figyelembe véve három fő tézist fogalmaztam meg.

KT1: Az egyesített ALM definíció alapján -amely tartalmazza az idő, költség, erőforrás és minőségi szempontokat is- a mátrix alapú tervezési modell alkalmazható ütemezési célokra az ALM környezetben, mivel az képes kezelni az ALM tervezett és nem tervezett tevékenységekre vonatkozó rugalmas követelményeket.

KT2: Az ALM ütemezési probléma reprezentálható kiterjesztett projektmenedzsment ütemezési problémaként, és a meglévő projektütemezési módszerek (TPM, APM, HPM) különböző megvalósításokhoz vezetnek az ALM környezetben:

KT2.1: A TPM megközelítés adja a legtöbb megvalósítható projektet, ha a rugalmasság alacsonyabb, és minden esetben a legmagasabb ügyfél-elégedettséget éri el, mivel az összes meghatározott feladatot végrehajtja, és kevesebb erőforrást igényel időegységenként. Ugyanakkor a legmagasabb költséget és a leghosszabb projekteket eredményezi.

KT2.2: Az APM megközelítés lényegesen több megvalósítható projektet eredményez, mint a TPM, amennyiben magasabb a rugalmasság, és általában a legrövidebb projekteket és a legalacsonyabb költségvetést produkálja, ha ettől eltérő

célfüggvényt választunk, azonban időegységenként több erőforrást igényel, mint a TPM.

KT2.3: A HPM megközelítés a legtöbb megvalósítható projektet eredményezi nagyobb rugalmasság esetén, és a legjobb teljesítményt mutatja az elérni kívánt célokhoz (költség, idő, minőség), továbbá a HPM biztosítja a legjobb projektütemezési eredményt.

KT3: A projekt hatóköréből összesen 9 kockázati tényező nyert megerősítést az ALM-környezetben, összefüggésben az idő, a költségek, az erőforrások és a minőség tényező; a hatókör változására fókuszáló, az ALM ütemezésre jellemző ALM-specifikus kockázati tényezők is azonosításra kerültek:

KT3.1: Az ALM környezetben a 3 leginkább befolyásoló kockázati tényező sorrendben: az alkalmazott projektmenedzsment módszer, a strukturális rugalmasság, és a kockázati tényezők közötti összefüggés.

KT3.2: Az ALM környezetben az alacsony szintű kockázatok (költség, idő, erőforrás-változás) nagyobb hatást fejtenek ki, mint a magas szintű kockázatok (célfüggvények korlátjainak teljesítése).

KT3.3: A TPM a legérzékenyebb a sokkhatásokra, ahol a feladatok követelményeinek mindössze 10%-os módosulásával, a hatás az időre, erőforrásokra, költségre akár tízszeres változást is eredményezhet.

5 Összefoglalás

A technológia gyors fejlődése megköveteli a szoftverprogramok vagy alkalmazások alapos megértését és hatékony kezelését, amelyek elengedhetetlenek a modern üzleti tevékenységekhez olyan iparágakban, mint az infokommunikációs, az autóipar, az egészségügy, a repülőgépipar és sok más terület. Az elmúlt évtizedekben erőteljes eltolódást láthattunk a szoftveralapú gazdaság felé. A szoftverek létrehozásának és karbantartásának egyre nagyobb gazdasági hatása van. Az Alkalmazás Életciklus-Menedzsment (ALM) keretet kínál egy ilyen megoldáshoz, mivel a teljes szoftverkezelési folyamatot a kezdetektől az alkalmazás fejlesztésén és karbantartásán keresztül egészen a kivonulásig kezeli, így hosszú távú gazdasági megtérülést biztosít a befektetések számára.

Összegzésként megállapítható, hogy a disszertáció a bevezetésben is említett és kitűzött céljai teljesültek. Ezeken felült, az esettanulmány értékelési folyamatában részt vevő szakemberek a napi rutinon kívüli kérdések feltevésével és megválaszolásával is hangsúlyozták a pozitív hatást, segítve őket a munkájuk újragondolásában, amit tudományos eredményekkel is alátámasztva.

5.1 Újszerű eredmények, hozzájárulás az irodalomhoz

Először is meg kell jegyezni, hogy az ALM-mel kapcsolatos szakirodalom jelenleg egy relatíve szűk terület, és több szempontból is bővítésre szorul. Köszönhetően annak, hogy döntően üzleti forrásokból származó publikációk nyomán indult, a mai napig sincs egyértelmű vagy egybehangzóan elfogadott ALM-definíció.

1. A disszertáció első részében a szakirodalmi áttekintésre fókuszált, tágabb értelemben az ALM részletesebb megismerésére, illetve a meglévő ALM definíciók felkutatására, annak érdekében, hogy következő lépésként egységes koncepciót lehessen létrehozni a további módszertani támogatásra. A keresztmetszeti, szisztematikus irodalom áttekintés módszerével a szakirodalom széles köre szolgált a meglévő definíciók alapjául. Ezután egy kritikai áttekintés folyt a hatókörök és attribútumok integrálására szolgáló egységes ALM-definíció létrehozására. Az első jelentős hozzájárulás tehát az ALM definícióval kapcsolatos szisztematikus irodalmi áttekintés volt. Ez egy longitudinális vagy meta-kutatás alapjaként is használható, például a SIMILAR módszerhez (Kosztján, Csizmadia és tsai., 2021) az ALM-irodalom további bővítéséhez. A kiegészítésként javasolt ALM definíció alapja lehet további akadémiai kutatásoknak, új távlatokat nyitva a módszertani kutatások előtt, hiszen a probléma már az üzleti életben jelen van, ahogy az az esettanulmányból is kiderült. Cikk formájában a vizsgálat eredményei jelenleg publikálás alatt állnak: *he Evolution of Definition in Application Lifecycle Management – A Systematic Literature Review Article with a Critical Analysis*.

2. Egy mátrix alapú módszer az IT területen az ALM feladatok ütemezési megvalósíthatóságának vizsgálata. A meglévő ágensként programozott projektmenedzsment megközelítésekkel (TPM, APM, HPM). Hasonló, megvalósíthatósággal kapcsolatos összehasonlítások korábban nem léteztek az ALM szakirodalomban. Az összetett szimulációk valós adatokat is használtak bemenetként. Ezért a második fő hozzájárulás innen származik. A módszer publikálásra került a *Survive IT! IT projekttervezési megközelítések túlélési elemzése* címmel, valamint az ALM-specifikus változat is előkészületben van.

3. Egy autóiipari beszállító cég ALM-hez kapcsolódó esettanulmány. A korábban bemutatott mátrix alapú módszerrel, kiterjesztve azt meglévő ALM környezetre, egy esettanulmány készült. Ez a számos szakértővel, vezető autóiipari beszállító vezetővel készült, összetett HW-SW vonatkozású megközelítés korábban nem szerepelt a szakirodalomban. Az Alkalmazás életciklus-menedzsment ütemezési problémáját a vezetőkkel folytatott interjúk és belső vizsgálatok után ismerték fel és valósították meg. A számszerűsített adatok és az ütemezési problémák több megközelítéssel (TPM, APM, HPM) történő elemzése mélyebb összefüggéseket és potenciális, további szervezetfejlesztéseket tárt fel a vállalat számára a nagyobb hatékonyság érdekében.

5.2 Gyakorlati vonatkozások

A szisztematikus szakirodalmi áttekintés gyakorlati szempontból is értékes betekintést nyújthat a szakemberek számára, mivel korábban nem volt hasonló áttekintés. Ha a cél az alapkoncepcióról és a lefedettség mértékéről a szakértők számára információkat gyűjteni, ez a dolgozat cikkével és elméleti értékelésével valósítható meg.

A javasolt módszer a hagyományos, agilis és hibrid projektmenedzsment-megközelítéseket hasonlítja össze a különböző érdekelt felek szemszögéből. Javaslatot tesz egy olyan meta-hálózat elemzési módszerre, amelyet szoftverfejlesztési projekteknél eddig nem alkalmaztak, és kiterjesztette azt az ALM környezetre is. Az elemzés kimutatta, hogy minden módszernek nem csupán előnyei, hanem hátrányai is vannak. Többségük összhangban van a tapasztalatokkal, de más módszerek mélyebb elemzést igényelnek. Először is, hasonlóan a tapasztalatokhoz, a hagyományos projektmenedzsment megközelítések produkálták a leginkább megvalósíthatatlan projektterveket. Ez az eredmény teljesen megegyezik a CHAOS jelentés (SGI, 2019) eredményeivel, ahol a hagyományos projektmenedzsment megközelítést követő, úgynevezett vízesés típusú projektek háromszor több sikertelen projektet eredményeztek. Ez a tanulmány azonban azt is kimutatta, hogy előny csak akkor jelentkezik, ha a feladatok és függőségek legalább 20%-a rugalmas. A tanulság az, hogy amikor ezt a követelményt nem lehet teljesíteni, az agilis projektmenedzsment megközelítés több sikertelen (vagyis megvalósíthatatlan) projektet eredményezhet.

Ütemezési vonatkozások a TPM-re Az ütemezést tekintve a hagyományos projektmenedzsment szemlélet és a megvalósított TPMa csak a többmódusú feladatvégzés szempontjából működik. Ez a megközelítés azt feltételezi, hogy a feladatokat különféle módokon lehet elvégezni. Ezzel szemben az agilis technikák rugalmas projektstruktúrát feltételeznek, ahol a feladatok közötti függőségek rugalmasak lehetnek, és az alacsonyabb prioritású feladatokat el lehet halasztani a következő projektig, de általában csak egy végrehajtási mód van megadva. Az eredmények azt mutatták, hogy egy rugalmas projektkörnyezet esetén, ahol magas a rugalmasság foka, ez a megközelítés valóban nagyobb arányú megvalósíthatóságot produkál és ezáltal lényegesen több projektet tud sikeressé tenni, mint a hagyományos megközelítések. Ez az előny azonban eloszlik, ha a technológia szigorú függőséget igényel.

A projekt rugalmasságának köszönhetően a másik lenyűgöző eredmény, hogy az agilis projektmenedzsment megközelítés általában a legrövidebb és legolcsóbb projekteket adja, még akkor is, ha egyetlen megvalósítási mód áll rendelkezésre. Ennek a stratégiának az ára azonban a szűkebb tartalom és az alacsonyabb minőség. Emiatt elengedhetetlen az ügyfelek bevonása, akik számára meg kell határozni a projektből kizárandó tevékenységek körét. Ugyanakkor a fejlesztők számára óriási kihívást jelent számos párhuzamos tevékenység egyidejű kezelése.

A hibrid projektmenedzsment-megközelítés mind a rugalmasságot, mind a végrehajtási módok választását kihasználhatja az ütemezéshez; ezért eredményezi a legjobb ütemterveket és a leginkább megvalósíthatókat, a kockázatelemzést követően

pedig a leginkább túlélő projekttervekkel rendelkezőket adja meg, de ezek az értékek csak adott célfüggvényekre a legjobbak.

A hibrid és különösen az APMa jobban teljesítenek a rugalmas projektkörnyezetben. Ebben az esetben megvalósíthatóbb és jobb (azaz rövidebb, olcsóbb stb.) projektek adhatók meg. Ennek ellenére a projekt szerkezete, mint például a méret és a párhuzamosítás, kevésbé fontos tényező a túlélés szempontjából. Az agilis és hagyományos projektmenedzsment megközelítések általában jobban tudják mérsékelni a kockázati tényezők hatásait, míg a hibrid megközelítés a leginkább túlélő projektek biztosítását segíti elő.

A hibrid technikák többféle módot és rugalmas struktúrákat is lehetővé tesznek, ezért feltételezzük, hogy ez a projektmenedzsment legsokoldalúbb technikája. Ezt a feltételezést megerősíti az a tény, hogy ez a technika biztosítja a megvalósítható megoldások legmagasabb arányát és a legjobb ütemezési teljesítményt, ha csak a célfüggvényt vesszük figyelembe. A javasolt adatbázis alapján a HPMa nyújtja a leginkább megvalósítható megoldásokat, ezért egy szoftverfejlesztési projekt nagyobb valószínűséggel éli túl a kockázati hatásokat, ha a projekttervet hibrid projektmenedzsment megközelítéssel kezelik.

Kitekintésképpen, az autóiipari esettanulmány adatai és információi értékes tudássá válhatnak az adott iparág szervezeti felépítéséhez és tevékenységeik ütemezéséhez. Az intelligens aktuátorok jövője és a SW terméké válásának kihívása (SWaaP) lépésről lépésre vezet az alkalmazás életciklus-menedzsment világához. Az ipar számára fontos a tudományos közösség hozzájárulása az ütemezés optimalizálásához és más folyamatokhoz, ezért a jövőben további fejlődés várható.

5.3 A kutatás összegzése

A jobb áttekinthetőség érdekében az alábbi táblázatban találhatóak a Kutatási Kérdések, Feltételezések és Tézisek összefoglalva.

Ref.	Állítás
KK1:	<i>Hogyan azonosítható a rendelkezésre álló tudományos irodalom alapján egy tervezési modell, amely az alkalmazás életciklus-menedzsmentet (ALM) reprezentálja, és amely felhasználható ütemezési feladatokhoz?</i>
KF1:	Azonosítható egy olyan tervezési modell, amely a különböző, egyesített szakirodalmi ALM definíciókon alapszik.
KT1:	Az egyesített ALM definíció alapján -amely tartalmazza az idő, költség, erőforrás és minőségi szempontokat is- a mátrix alapú tervezési modell alkalmazható ütemezési célokra az ALM környezetben, mivel az képes kezelni az ALM tervezett és nem tervezett tevékenységekre vonatkozó rugalmas követelményeket.
KK2:	<i>A jelenlegi projektmenedzsment ütemezési módszertanok (TPM, APM, HPM) eredményeznek-e megvalósítható megoldásokat az ALM környezetben, és azok hogyan teljesítenek?</i>
KF2:	A projektmenedzsment megközelítésekhez (TPM, APM, HPM) kapcsolódó mátrixos tervezési módszer kiterjeszhető, amellyel megoldható az ALM ütemezési probléma. Egy szimulációs keretrendszer elkészíthető a rugalmas függőségek és nem tervezett feladatok kezelésére.
KT2:	Az ALM ütemezési probléma reprezentálható kiterjesztett projektmenedzsment ütemezési problémaként, és a meglévő projektütemezési módszerek (TPM, APM, HPM) különböző megvalósításokhoz vezetnek az ALM környezetben:
KT2.1:	A TPM megközelítés adja a legtöbb megvalósítható projektet, ha a rugalmasság alacsonyabb, és minden esetben a legmagasabb ügyfél-elégedettséget éri el, mivel az összes meghatározott feladatot végrehajtja, és kevesebb erőforrást igényel időegységenként. Ugyanakkor a legmagasabb költséget és a leghosszabb projekteket eredményezi.
KT2.2:	Az APM megközelítés lényegesen több megvalósítható projektet eredményez, mint a TPM, amennyiben magasabb a rugalmasság, és általában a legrövidebb projekteket és a legalacsonyabb költségvetést produkálja, ha ettől eltérő célfüggvényt választunk, azonban időegységenként több erőforrást igényel, mint a TPM.
KT2.3:	A HPM megközelítés a legtöbb megvalósítható projektet eredményezi nagyobb rugalmasság esetén, és a legjobb teljesítményt mutatja az elérni kívánt célokhoz (költség, idő, minőség), továbbá a HPM biztosítja a legjobb projektütemezési eredményt.
KK3:	<i>Melyek az ütemezési probléma kockázati tényezői az ALM-környezetben, és ezek hogyan befolyásolják a megvalósíthatóságot és az ütemezési teljesítményt?</i>
KF3:	Léteznek projekthez kapcsolódó kockázati tényezők, amelyek kiterjeszthetők az ALM ütemezési problémákra, habár a projekt és az ALM hatókörének különbségei miatt ALM-specifikus kockázatok is megjelennek, amelyek hatással lehetnek az erőforrásokra, költségekre, időzítésre és befolyásolhatják a megvalósíthatóságot, valamint az ütemezési teljesítményt.
KT3:	A projekt hatóköréből összesen 9 kockázati tényező nyert megerősítést az ALM-környezetben, összefüggésben az idő, a költségek, az erőforrások és a minőség tényező; a hatókör változására fókuszáló, az ALM ütemezésre jellemző ALM-specifikus kockázati tényezők is azonosításra kerültek:
KT3.1:	Az ALM környezetben a 3 leginkább befolyásoló kockázati tényező sorrendben: az alkalmazott projektmenedzsment módszer, a strukturális rugalmasság, és a kockázati tényezők közötti összefüggés.
KT3.2:	Az ALM környezetben az alacsony szintű kockázatok (költség, idő, erőforrás-változás) nagyobb hatást fejtenek ki, mint a magas szintű kockázatok (célfüggvények korlátjainak teljesítése).
KT3.3:	A TPM a legérzékenyebb a sokkhatásokra, ahol a feladatok követelményeinek mindössze 10%-os módosulásával, a hatás az időre, erőforrásokra, költségre akár tízszeres változást is eredményezhet.

5. táblázat. Kutatás összegzése

6 A szerző kapcsolódó publikációi

Nemzetközi cikkek

- Kosztján, Z. T., Novák, G., **Jakab, R.**, Szalkai, I., & Hegedűs, C. (2022). A matrix-based flexible project-planning library and indicators. In: *Expert Systems With Applications*. DOI: doi.org/10.1016/j.eswa.2022.119472
- Kosztján, Z. T., **Jakab, R.**, Novák, G., & Hegedűs, C. (2020). Survive IT! Survival analysis of IT project planning approaches. In: *Operations Research Perspectives*, 7, 100170. DOI: doi.org/10.1016/j.orp.2020.100170
- Jakab, R.**, Novák, G. (2018). Project management approaches in application management services. In: *Chapters from the Academic Aspect of Project Management-Research and Teaching Methodologies Volume II.*, pp. 152-171. (ISBN: 9786150042190).

Bírálat alatt

- Jakab, R.**, Kosztján, Z. T. (2024). The Evolution of Definition in Application Lifecycle Management – A Systematic Literature Review Article with a Critical Analysis. Under review in: *Journal of Information Technology (JIT SAGE)*.

Konferenciaközlemények

- Kosztján, Z. T., Novák, G., **Jakab, R.**, & Hegedűs, C. (2022). A Matrix-based Flexible Multi-level Project Planning Library and Indicators. In: *Proceedings of the 24th International DSM Conference (DSM 2022)*, Eindhoven, The Netherlands, October, 11-13, 2022 (pp. 48-57). DOI: doi.org/10.35199/dsm2022.06

Konferenciaelőadások

- Jakab, R.** (2023). Defining the way of Application Lifecycle Management. Abstract. PMUni International Conference on Project Management - PMUni 2023, Vienna, Austria.
- Jakab, R.**, & Novák, G. (2023). Defining the way of Application Lifecycle Management. Abstract. OGIK-ISBIS 2023 Conference Proceedings, pp. 41., Pécsi Tudományegyetem, Pécs, Hungary.
- Novák, G., & **Jakab, R.** (2021). Multi-level project planning and simulation using different delay cost profiles. Abstract. 15th International Conference on Economics and Business, Hungarian University of Transylvania, Miercurea Ciuc, Romania.
- Jakab, R.**, & Novák, G. (2019). Application Lifecycle Management: evolution and revolution. Abstract. OGIK-ISBIS 2019 Conference Proceedings, pp. 51-52., Milton Friedman University, Budapest, Hungary. (Winning the Conference Award Best Presentation)
- Novák, G., & **Jakab, R.** (2019). Multi-level Project Planning and Simulation using Earliness/Tardiness Compensation Profiles. Abstract. OGIK-ISBIS 2019 Conference Proceedings, pp. 44-45., Milton Friedman University, Budapest, Hungary.
- Novák, G., & **Jakab, R.** (2018). A parser for standard datasets in project scheduling and simulation. Abstract. In: Bacsárdi, L., Bencsik, G., Pödör Z. OGIK-ISBIS 2018 Conference Proceedings, University of Sopron, pp. 20-21., Sopron, Hungary ISBN: (9786158109802)
- Jakab, R.**, Novák, G. & (2018). Simulation and modeling of flexible projects and Application Management. Abstract. In: Bacsárdi, L., Bencsik, G., Pödör Z. OGIK-ISBIS 2018 Conference Proceedings, University of Sopron, pp. 47-48., Sopron, Hungary ISBN: (9786158109802)

Felhasznált irodalom

- Abdolshah, Mohammad (2014). „A review of resource-constrained project scheduling problems (RCPSP) approaches and solutions”. *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies* 5.4, 253–286. old.
- Aiello, Bob és Sachs, Leslie (2016). *Agile application lifecycle management: Using DevOps to drive process improvement*. Addison-Wesley Professional.
- Ajmal, Mian M, Khan, Mehmood, Gunasekaran, Angappa és Helo, Petri T (2022). „Managing project scope creep in construction industry”. *Engineering, Construction and Architectural Management* 29.7, 2786–2809. old.
- Akgun, Zuleyha, Yilmaz, Murat és Clarke, Paul (2020). „Assessing application lifecycle management (ALM) potentials from an industrial perspective”. *Systems, Software and Services Process Improvement: 27th European Conference, EuroSPI 2020, Düsseldorf, Germany, September 9–11, 2020, Proceedings* 27. Springer, 326–338. old.
- Albtoush, AM Faten és Doh, SI (2019). „A Review on causes of cost overrun in the construction projects”. *Int. J. New Innov. Eng. Technol* 12.3, 15–022. old.
- Banjanin, Goran és Strahonja, Vjeran (2018). „Factors of Risk Reduction in Agile and Lean Enabled Governance of IT Project Portfolios”.
- Barafort, Béatrix, Mesquida, Antoni-Lluís és Mas, Antónia (2019). „ISO 31000-based integrated risk management process assessment model for IT organizations”. *Journal of Software: Evolution and Process* 31.1, e1984.
- Batselier, Jordy és Vanhoucke, Mario (2015). „Construction and evaluation framework for a real-life project database”. *International Journal of Project Management* 33.3, 697–710. old. ISSN: 0263-7863. DOI: <https://doi.org/10.1016/j.ijproman.2014.09.004>. URL: <http://www.sciencedirect.com/science/article/pii/S0263786314001410>.
- Beek, T van der, Souravlias, D, Essen, JT van, Pruyn, J és Aardal, K (2024). „Hybrid differential evolution algorithm for the resource constrained project scheduling problem with a flexible project structure and consumption and production of resources”. *European Journal of Operational Research* 313.1, 92–111. old.
- Buganová, Katarína és Šimíčková, Jana (2019). „Risk management in traditional and agile project management”. *Transportation Research Procedia* 40. TRANSCOM 2019 13th International Scientific Conference on Sustainable, Modern and Safe Transport, 986–993. old. ISSN: 2352-1465. DOI: <https://doi.org/10.1016/j.trpro.2019.07.138>. URL: <https://www.sciencedirect.com/science/article/pii/S2352146519303060>.
- Castaneda, R Riascos, Ostrosi, E, Majić, T, Stjepandić, J és Sagot, J-C (2020). „A method to explore product risk in product lifecycle management of configured products”. *Proceedings of the design society: DESIGN conference*. 1. köt. Cambridge University Press, 687–696. old.
- Chanda, Sandeep, Foggon, Damien, Chanda, Sandeep és Foggon, Damien (2013). „Application lifecycle management”. *Beginning ASP. NET 4.5 Databases*, 235–249. old.
- Chappell, David és tsai. (2010). „What is application lifecycle management?”: *Chappell & Associates*, 6. old.
- Cheng, Yu Chin (2010). „Rapid Application Lifecycle Management: a new Approach with Tool Support.”
- Choetkiertikul, Morakot és Sunetnanta, Thanwadee (2012). „A risk assessment tool using a CMMI Quantitative Approach”. *International Journal of Engineering and Technology* 4.4, 352. old.

- Corallo, Angelo, Latino, Maria Elena, Menegoli, Marta és Pontrandolfo, Pierpaolo (2020). „A systematic literature review to explore traceability and lifecycle relationship”. *International Journal of Production Research* 58.15, 4789–4807. old.
- Creemers, Stefan (2015. júl.). „Minimizing the expected makespan of a project with stochastic activity durations under resource constraints”. *Journal of Scheduling* 18.3, 263–273. old. ISSN: 1099-1425. DOI: 10.1007/s10951-015-0421-5. URL: <https://doi.org/10.1007/s10951-015-0421-5>.
- Demeulemeester, Erik, Vanhoucke, Mario és Herroelen, Willy (2003. jan.). „RanGen: A Random Network Generator for Activity-on-the-Node Networks”. *Journal of Scheduling* 6.1, 17–38. old. ISSN: 1099-1425. DOI: 10.1023/A:1022283403119. URL: <https://doi.org/10.1023/A:1022283403119>.
- Demeulemeester, Erik Leuven és Herroelen, Willy S (2006). *Project scheduling: a research handbook*. 49. köt. Springer Science & Business Media.
- Deuter, Andreas, Otte, Andreas, Ebert, Marcel és Possel-Dölken, Frank (2019). „Developing the Requirements of a PLM/ALM Integration: An Industrial Case Study”. *Product Lifecycle Management (Volume 4): The Case Studies*, 125–143. old.
- Deuter, Andreas és Rizzo, Stefano (2016). „A critical view on PLM/ALM convergence in practice and research”. *Procedia Technology* 26, 405–412. old.
- Dingsøyr, Torgeir, Nerur, Sridhar, Balijepally, VenuGopal és Moe, Nils Brede (2012). „A decade of agile methodologies: Towards explaining agile software development”. *Journal of Systems and Software* 85.6. Special Issue: Agile Development, 1213–1221. old. ISSN: 0164-1212. DOI: <https://doi.org/10.1016/j.jss.2012.02.033>. URL: <http://www.sciencedirect.com/science/article/pii/S0164121212000532>.
- Ebert, Christof (2013). „Improving engineering efficiency with PLM/ALM”. *Software & Systems Modeling* 12.3, 443–449. old.
- Government Commerce, OGC-Office of (2007). *The official introduction to the ITIL service lifecycle*. The Stationery Office.
- Hartmann, Sönke és Briskorn, Dirk (2021). „An Updated Survey of Variants and Extensions of the Resource-Constrained Project Scheduling Problem”. *European Journal of Operational Research*.
- Hummer, Waldemar, Muthusamy, Vinod, Rausch, Thomas, Dube, Parijat, El Maghraoui, Kaoutar, Murthi, Anupama és Oum, Punleuk (2019). „Modelops: Cloud-based lifecycle management for reliable and trusted ai”. *2019 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 113–120. old.
- Issa, S és Tu, Yiliu (2020). „A survey in the resource-constrained project and multi-project scheduling problems”. *Journal of Project Management* 5.2, 117–138. old.
- Jackson, Simon (2002). „Project cost overruns and risk management”. *Proceedings of Association of Researchers in Construction Management 18th Annual ARCOM Conference, Newcastle, Northumber University, UK*. 1. köt., 1–10. old.
- Kääriäinen, Jukka és Välimäki, Antti (2008). „Impact of Application Lifecycle Management—A Case Study”. *Enterprise Interoperability III*. Springer, 55–67. old.
- Kolisch, Rainer és Sprecher, Arno (1997). „{PSPLIB} - A project scheduling problem library: {OR} Software - {ORSEP} Operations Research Software Exchange Program”. *European Journal of Operational Research* 96.1, 205–216. old. ISSN: 0377-2217. DOI: [http://dx.doi.org/10.1016/S0377-2217\(96\)00170-1](http://dx.doi.org/10.1016/S0377-2217(96)00170-1). URL: <http://www.sciencedirect.com/science/article/pii/S0377221796001701>.
- Komal, Bakhtawar, Janjua, Uzair Iqbal, Anwar, Fozia, Madni, Tahir Mustafa, Cheema, Muhammad Faisal, Malik, Muhammad Noman és Shahid, Ahmad Raza

- (2020). „The impact of scope creep on project success: An empirical investigation”. *IEEE Access* 8, 125755–125775. old.
- Kossmann, Mario (2016). *Requirements management: How to ensure you achieve what you need from your projects*. Routledge.
- Kosztján, Zsolt T (2022). „MFPP: Matrix-based flexible project planning”. *SoftwareX* 17, 100973. old.
- Kosztján, Zsolt T, Csizmadia, Tibor és Katona, Attila I (2021). „SIMILAR–Systematic iterative multilayer literature review method”. *Journal of Informetrics* 15.1, 101111. old.
- Kosztján, Zsolt T és Szalkai, István (2018). „Hybrid time-quality-cost trade-off problems”. *Operations Research Perspectives* 5, 306–318. old. ISSN: 2214-7160. DOI: <https://doi.org/10.1016/j.orp.2018.09.003>. URL: <http://www.sciencedirect.com/science/article/pii/S2214716018300812>.
- Kosztján, Zsolt T és Szalkai, István (2020). „Multimode resource-constrained project scheduling in flexible projects”. *Journal of Global Optimization* 76.1, 211–241. old.
- Kreter, Stefan, Schutt, Andreas, Stuckey, Peter J és Zimmermann, Jürgen (2018). „Mixed-integer linear programming and constraint programming formulations for solving resource availability cost problems”. *European Journal of Operational Research* 266.2, 472–486. old.
- Kwak, Young Hoon és Ingall, Lisa (2007). „Exploring Monte Carlo simulation applications for project management”. *Risk Management* 9.1, 44–57. old.
- Madhuri, K Lakshmi, Suma, V és Mokashi, Uma Mohan (2018). „A triangular perception of scope creep influencing the project success”. *International Journal of Business Information Systems* 27.1, 69–85. old.
- Majerowicz, Walt és Shinn, Stephen A (2016). „Schedule matters: Understanding the relationship between schedule delays and costs on overruns”. *2016 IEEE Aerospace Conference*. IEEE, 1–8. old.
- Mishra, Alok és Alzoubi, Yehia Ibrahim (2023). „Structured software development versus agile software development: a comparative analysis”. *International Journal of System Assurance Engineering and Management*, 1–19. old.
- Mishra, Anjay Kumar (2020). „Implication of theory of constraints in project management”. *International Journal of Advanced Trends in Engineering and Technology* 5.1, 1–13. old.
- Moukrim, Aziz, Quilliot, Alain és Toussaint, H el ene (2015). „An effective branch-and-price algorithm for the preemptive resource constrained project scheduling problem based on minimal interval order enumeration”. *European Journal of Operational Research* 244.2, 360–368. old.
- Niemann, J org  s Pisl a, Adrian (2018). „Sustainable potentials and risks assess in automation and robotization using the life cycle management index tool—LY-MIT”. *Sustainability* 10.12, 4638. old.
- Otibine, Tobias, Mbuguah, Samuel, Kilwake, Humphrey  s Tsinale, Harriet (2017. nov.). „Application Lifecycle Management Activities For Quality Assurance In Software Development”. *International Journal of Trend in Research and Development* 4, 2394–9333. old.
- Page, Matthew J  s tsai. (2021). „The PRISMA 2020 statement: an updated guideline for reporting systematic reviews”. *International journal of surgery* 88, 105906. old.
- Park, Jung Eun (2021). „Schedule delays of major projects: what should we do about it?”: *Transport Reviews* 41.6, 814–832. old.

- Peteghem, Vincent Van és Vanhoucke, Mario (2014). „An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances”. *European Journal of Operational Research* 235.1, 62–72. old. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2013.10.012>. URL: <http://www.sciencedirect.com/science/article/pii/S0377221713008357>.
- Pirklbauer, Guenter, Ramler, Rudolf és Zeilinger, Rene (2009). „An integration-oriented model for application lifecycle management”. *International Conference on Enterprise Information Systems*. 1. köt. SCITEPRESS, 399–402. old.
- Rahman, Md Humyun Fuad, Chakraborty, Ripon K és Ryan, Michael J (2021). „Managing uncertainty and disruptions in resource constrained project scheduling problems: a real-time reactive approach”. *IEEE Access* 9, 45562–45586. old.
- Rao, Purnima és Palaniappan, Kandhasami (2020). „Continuous Engineering Through ALM-PLM Integration”. *Product Lifecycle Management Enabling Smart X: 17th IFIP WG 5.1 International Conference, PLM 2020, Rapperswil, Switzerland, July 5–8, 2020, Revised Selected Papers 17*. Springer, 798–811. old.
- Rasnacis, Arturs és Berzisa, Solvita (2017). „Method for Adaptation and Implementation of Agile Project Management Methodology”. *Procedia Computer Science* 104. ICTE 2016, Riga Technical University, Latvia, 43–50. old. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2017.01.055>. URL: <https://www.sciencedirect.com/science/article/pii/S187705091730056X>.
- Reiff, Janine és Schlegel, Dennis (2022). „Hybrid project management—a systematic literature review”. *International journal of information systems and project management* 10.2, 45–63. old.
- Rossberg, Joachim (2019). „Introduction to Application Life Cycle Management”. *Agile Project Management with Azure DevOps*. Springer, 1–36. old.
- Rossman, Bruce (2010). *Application Lifecycle Management-Activities, Methodologies, Disciplines, Tools, Benefits, ALM Tools and Products*. Emereo Pty Ltd.
- Roy, B (1962). „Cheminement et connexité dans les graphes”. *Applications aux problèmes d'ordonnancement*. METRA: Série Spéciale.1.
- Sahu, Kavita, Shree, R és Kumar, R (2014). „Risk management perspective in SDLC”. *International Journal of Advanced Research in Computer Science and Software Engineering* 4.3.
- Al-Saqqa, Samar, Sawalha, Samer és AbdelNabi, Hiba (2020). „Agile software development: Methodologies and trends.” *International Journal of Interactive Mobile Technologies* 14.11.
- SIGI (2019). *CHAOS Manifesto*. Standish Group International.
- Shafqat, A, Oehmen, Josef és Welo, Torgeir (2022). „Planning unplanned design iterations using risk management and learning strategies”. *Journal of Engineering Design* 33.2, 120–143. old.
- Sonnemann, Guido, Gemechu, Eskinder Demisse, Remmen, Arne, Frydendal, Jeppe és Jensen, Allan Astrup (2015). „Life cycle management: Implementing sustainability in business practice”. *Life cycle management*, 7–21. old.
- StandishGroup (2020). „The CHAOS report”. URL: <http://www.standishgroup.com>.
- Tritschler, Martin, Naber, Anulark és Kolisch, Rainer (2017). „A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles”. *European Journal of Operational Research* 262.1, 262–273. old.
- Tudenhöfner, Eduard (2011). *Integration of Performance Management into the Application Lifecycle*. diplom. de.

- Tüzün, Eray, Tekinerdogan, Bedir, Macit, Yagup és İnce, Kürşat (2019). „Adopting integrated application lifecycle management within a large-scale software company: An action research approach”. *Journal of Systems and Software* 149, 63–82. old.
- van den Ende, Leonore és van Marrewijk, Alfons (2014). „The ritualization of transitions in the project life cycle: A study of transition rituals in construction projects”. *International Journal of Project Management* 32.7, 1134–1145. old. ISSN: 0263-7863. DOI: <https://doi.org/10.1016/j.ijproman.2014.02.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0263786314000295>.
- Vanhoucke, Mario, Coelho, JosÃ©, Debels, Dieter, Maenhout, Broos és Tavares, Luís V. (2008). „An evaluation of the adequacy of project network generators with systematically sampled networks”. *European Journal of Operational Research* 187.2, 511–524. old. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2007.03.032>. URL: <http://www.sciencedirect.com/science/article/pii/S037722170700375X>.
- Venkatesh, Bharti és Balani, Lalit (2016). „Requirement management a key to successful project management for software systems”. *Voice of Research* 5.1.
- Wawak, Sławomir, Ljevo, Žaneta és Vukomanović, Mladen (2020). „Understanding the key quality factors in construction projects—A systematic literature review”. *Sustainability* 12.24, 10376. old.