

TÉZISFÜZET

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

*Szerző:*

JAKAB Róbert

*Témavezető:*

Prof. Dr. habil. KOSZTYÁN Zsolt Tibor



PANNON EGYETEM

Veszprém  
2024



## 1 Bevezetés

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

Az **Alkalmazás Életciklus-Menedzsment**<sup>1</sup> számos iparágban (IT, autóipar, légi közlekedés stb.) a vállalkozások kulcsfontosságú fókuszpontjává vált, amelynek célja, hogy megoldást nyújtson az alkalmazásszoftverek teljes életcikluson át, a kezdetektől egészen azok kivezetéséig. Az elmúlt néhány évtizedben a szoftverközpontú gazdaság irányába történő páratlan átalakulás szemtanúi lehettünk, 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-Saqqa és tsai., 2020; van den Ende és van Marrewijk, 2014).

A szoftverprojektek kudarca gazdaságilag katasztrofális lehet. A késések, a költségvetés túllépése és az optimálistól elmaradó szoftverminőség nemcsak költséges, hanem a piaci versenyképességet is rontja. Az ALM megfelelő használata elengedhetetlen a kockázatok mérsékléséhez és a szoftverfejlesztési kihívások gazdasági hatékonyságának maximalizálásához, mivel az ALM kezeli a teljes szoftveralkalmazási életciklust, és kulcsszerepet játszik annak biztosításában, hogy a befektetések fenntartható gazdasági megtérülést eredményezzenek. Ezért kulcsfontosságú a hatékonyság az alkalmazások kezelésben, például a megfelelő ütemezési módszertan alkalmazása egy ALM környezetben.

Mivel az akadémiai irodalomban nem létezik egyértelműen és általánosan elfogadott definíció az ALM-re, a gyártók megragadták a lehetőséget, hogy a maguk javára fordítsák a helyzetet és saját céljaiknak megfelelően módosítsák azt. Ez a kétséges helyzet ahhoz vezetett, hogy már a tudományos elemzések és fejlesztések folytatását is hátráltatja. A megfelelő támogatás hiányában ezért az üzleti szereplők a már elérhető projektmenedzsment módszereket alkalmazzák, mint például a tradicionális- (TPM), az agilis- (APM) és a hibrid projektmenedzsment (HPM),

---

<sup>1</sup>Agolul Application Lifecycle Management (ALM), magyarra fordítva még az Applikáció Életciklus-Kezelés és az Applikáció Menedzsment kifejezést is használja a szakirodalom.

amelyek esetleg csak részben felelnek meg céljaik eléréséhez, és valószínűleg alacsonyabb hatékonysággal működnek.

A disszertáció kutatásának hármas célja a következő volt:

- ✓ Szakirodalmi kutatás
  - [+] az elérhető ALM definíció és értelmezési tartomány meghatározására,
  - [+] egységesített átfogó definíció meghatározása, amely támogatja a további módszertani kutatásokat.
- ✓ A mátrixos reprezentáció alkalmazhatóságának megerősítése ütemezési vizsgálatra, beleértve:
  - [+] szimulációs (modellezett) környezet felállítása,
  - [+] TPM, APM és HPM ütemezések megoldhatósági vizsgálata,
  - [+] TPM, APM és HPM ütemezések hatékonyságának vizsgálata.
- ✓ Megvizsgálni a kockázati tényezők hatását az ALM ütemezési struktúrájában.

Validálásként egy releváns ALM esettanulmány kivitelezése ütemezési teljesítmény-értékeléssel.

## 2 Kutatási kérdések

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

**KK1:** Hogyan készíthető a rendelkezésre álló tudományos irodalom alapján olyan tervezési modell, amely az alkalmazás életciklus-menedzsment (ALM) problémáját reprezentálja, és amely felhasználható ütemezési feladatok módszertani fejlesztéséhez?

**KK2:** A jelenlegi projektmenedzsment módszertanok (TPM, APM, HPM) eredményeznek-e megvalósítható megoldásokat az ALM környezetben? Hogyan teljesítenek az ALM problémák ütemezésében?

**KK3:** Melyek az ütemezési probléma kockázati tényezői az ALM környezetben? Mely projekttervezési és ütemezési megközelítések enyhítik a legtöbb kockázatot az ALM környezetben? Hogyan befolyásolják az ALM-specifikus kockázati tényezők a megvalósíthatóságot és az ütemezési teljesítményt?

### 3 Szakirodalmi háttér és kutatási feltételezések

#### Applikáció Életciklus-Menedzsment

Alkalmazásnak vagy applikációnak nevezünk egy meghatározott típusú szoftvert, amelyet arra terveztek, hogy egy adott funkciót vagy kapcsolódó funkciók készletét hajtson végre a végfelhasználók számára. Az alkalmazások felhasználóbarátok, és meghatározott célokat szolgálnak. 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 ezt a tágabb jelentését értjük.

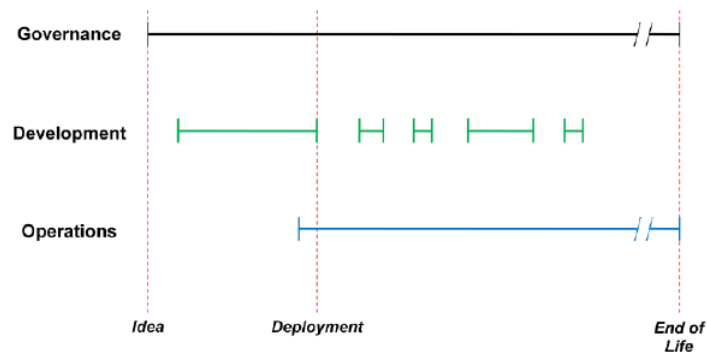
A szoftverprojektek teljesítménye történelmileg rosszabb a hagyományos projektekhez képest, pl. építkezés, és erre a jelenségre már a kilencvenes években felfigyelték a CHAOS jelentésben (StandishGroup, 2020). Számos szoftverfejlesztés-specifikus tényező befolyásolja ezt az alulteljesítést, amelyek más termékfejlesztéssel kapcsolatos projektekben nincsenek vagy csak részben vannak jelen, mint például a relatíve komplex tervezési folyamatok, az egyértelmű követelmények időbeni elérhetőségének fontossága, a konkrét együttműködési módszerek és eszközök szinkronizációja, stb. emeli ki a tanulmány.

A termékéletciklus-menedzsment általános megközelítése a PLM-mel (Product Lifecycle Management) valósult meg, kezdetben annak a szoftver csak mint egy részlete, azonban az egyre növekvő SW-bonyolultság miatt ennek kezelése szakosodást igényel, amit először szoftverfejlesztési életciklusnak neveztek. (SDLC<sup>2</sup>) menedzsment, későbbi már mint alkalmazás életciklus-management. Ennek az egyedülálló IT-területnek a látszólagos fontossága ellenére a kutatói közösség általában a PLM- és ALM-eszköz készítőkre hagyta a két terület összefűzését, akik hosszú távon azok konvergenciáját (Deuter, Otte és tsai., 2019; Deuter és Rizzo, 2016; Rao és Palaniappan, 2020) támogatják. Természetesen még mindig erős átfedések és kapcsolatok vannak a PLM és ALM projekt- és programmenedzsment elemei között, például a HW és az SW termékek közötti kapcsolatok, a változáskezelés, az együttműködés és a jelentéskezelések (Deuter és Rizzo, 2016) területén. Ez is mutatja, hogy az óriási mennyiségű okoseszköz, pl. az infokommunikációs szektorban vagy akár az autóiparban is az ALM kihívásával kell szembenéznük a gyártóknak. A disszertáció esettanulmánya is megemlíti, hogyan valósul meg az ALM egy autóipari beszállító cégnél, és milyen kihívások állnak szemben a mindennapi üzleti életben a cégek.

Az *Applikáció Életciklus-Menedzsment területe* sok kutató szerint (Chappell és tsai., 2010; Corallo és tsai., 2020; Rossberg, 2019; Rossman, 2010) olyan átfogó szoftverfejlesztési megközelítés, amely a szoftveralkalmazás teljes élettartamát felöleli, a szoftverfejlesztéstől kezdve annak üzembe helyezésén át, a végső visszavonásáig. Ebben az életciklusban több szempont is jelen van, de ami az ALM-et egyedivé teszi a

<sup>2</sup>Angol rövidítése a Software Development Life-Cycle kifejezésnek.

hagyományos projektmenedzsmenthez vagy szolgáltatásorientált megközelítéshez képest, az az, hogy a fejlesztési (Development) folyamatban a fő fázis (első, hosszabb zöld szakasz) után potenciálisan további nem várt feladatok, sőt alprojektek jelenhetnek meg elszórtan (kisebb zöld szakaszok) ahogy az alábbi 1 ábrán is látható. Ez a fajta megközelítés egyedülálló az alkalmazásfejlesztésben, hogy már az Üzemeltési (Operations) szakaszban több tervezett vagy nem tervezett változás történik.



1. ábra. Az ALM fő folyamatai (Chappell és tsai., 2010)

Tekintettel arra, hogy a változások egy SW-fejlesztésnél meghatározóbban érvényesülnek, a klasszikus projektmenedzsment folyamatmodellek, mint például a vízésés fejlesztési modell, amely már a projekt elején egyértelmű specifikáció-meghatározást vár el, -változtatás nélkül, vagy csak minimális változtatásokkal-szenved ettől a fajta SW fejlesztési módtól. Így módszertani szempontból rugalmasabb megközelítések kívánatosak a helyzet megoldására, mint például az agilis-projektmenedzsment (SGL, 2019) vagy a részlegesen hagyományos és agilis kombinációt tartalmazó, úgynevezett hibrid projektmenedzsment megközelítések egy ilyen ALM-környezetben. Meg kell állapítsuk, hogy óriási hiányosságok vannak a tudományos területen, hogy miként lehet jól azonosítani és körbeírni ezt a problémát, és hogyan lehet pontosan az ilyen ALM specifikus helyzeteket hatékonyan kezelni.

Összefoglalva, látható, hogy az ALM szakirodalom a tudományos körökben még mindig szűkös, és alapos vizsgálatra van szükség a megfelelő definíció és hatáskör megtalálásához. Egy ilyen áttekintés a szisztematikus szakirodalmi áttekintéssel (SLR<sup>3</sup>) lehetséges.

### Projektmenedzsment megközelítések

A *tradicionális projektmenedzsment* (TPM) megközelítésben (például építési projekt vagy szoftverfejlesztési projekt, amely egy vízésés folyamatmodellt követi) az a fő kérdés, hogy mennyibe kerül az előre meghatározott követelmények megvalósítása. Ezért habár a tartalom adott amit végre kell hajtani, az idő, a költség és a minőség szükség esetén átváltható. Ez a megközelítés egynél több kivitelezési módot

<sup>3</sup>Angol rövidítés: Systematic Literature Review

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 kérdés, hogy az adott költségkereten és időintervallumon belül (pl. sprintben) mennyi szolgáltatást lehet beépíteni. Míg egy sprinten belül a tartalom fix, így új feladatok nem megengedettek, addig a tervezési időszakban a sprintek között, azok átrendezhetőek, 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. Különböző projektmenedzsment megközelítések összehasonlítása

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

Az agilis és hibrid projektek jellemzője, például az ügyfelek bevonása a fejlesztési folyamatba, az erős vezetői támogatás biztosítása, valamint a felmerülő igényekkel való megbirkózás képessége, 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 összefoglaló táblázata a 4. bekezdés 4 táblázatában látható az érdekelt felek különböző szempontjaiból származó előnyök és hátrányok tekintetében.

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

### 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 (RCPSP<sup>4</sup>) 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

<sup>4</sup>Angol rövidítés a Resource Constrained Project Scheduling Problem kifejezésből.

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 és osztályozást kínál az RCPSP legjelentősebb bővítményeiről.

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 alkalmazásának vizsgálata a cél, amely elindíthatja a kutatást akár egy lehetséges ALM-specifikus bővítés előtt is az RCPSP számára.

Az ALM-hez kapcsolódó kockázatértékelés különösen fontos és érdekes lehet nem csak az akadémikusok, hanem az üzleti élet szereplői és a területi szakértői számára is, mivel a kockázat realizálása általában az életciklus során jelentkező negatív hatásokhoz kapcsolódik. Ezért minél felkészültebbek vagyunk, annál jobban kezelhetjük őket, ami azt jelenti, hogy a kockázatok korai szakaszban történő felismerése és mérséklése potenciálisan kevesebb vagy kisebb hatással lehet a meglévő tervünkre. Mivel az alkalmazások életciklus-menedzsmentje még az informatikai területen is sajátos terület, több tényező relevanciáját is meg kell vizsgálni, illetve az ALM-specifikus kockázatokat is azonosítani kell. Ez a kutatás összefoglalót ad a projektspecifikus és az ALM-specifikus kockázati tényezők megállapítására és azok összehasonlításához mind elméleti síkon, mint a szimulációból származó eredményekből is. A 4. bekezdésben található 5. táblázatban azonosítom és leírom azokat a főbb pontokat, amelyeket az ALM terület vizsgálata során találtam az irodalmi áttekintés során.

### 3.1 Kutatási feltételezések

A kutatási kérdéseket a szakirodalom alapos átvizsgálása után felmérve lehetővé vált a kapcsolódó kutatási feltételezések kialakítása. Három kutatási feltételezés létezik:

**KF1:** Létrehozható egy olyan modell, amely egyesíti a szakirodalomból származó különböző ALM-attribútumokat, amely teljesíti a rugalmas tervezési megközelítést az idő-, költség-, erőforrás- (megújuló és nem megújuló) és minőségi igények figyelembevételével, beleértve a nem tervezett feladatokat is.

**KF2:** A projektmenedzsment megközelítésekhez (TPM, APM, HPM) kapcsolódó mátrix tervezési módszer kiterjeszthető, amely lehetővé teszi, hogy az ütemező ügynök megoldja a problémát, és megvalósítható megoldásokat eredményezzen az ALM környezetben. Az ALM-problémák ütemezhetők, hogy közel optimális megoldásokat találjanak meghatározott korlátok mellett. A szimulációs keretrendszer létrehozható rugalmas függőségek és nem tervezett feladatok kezelésére.



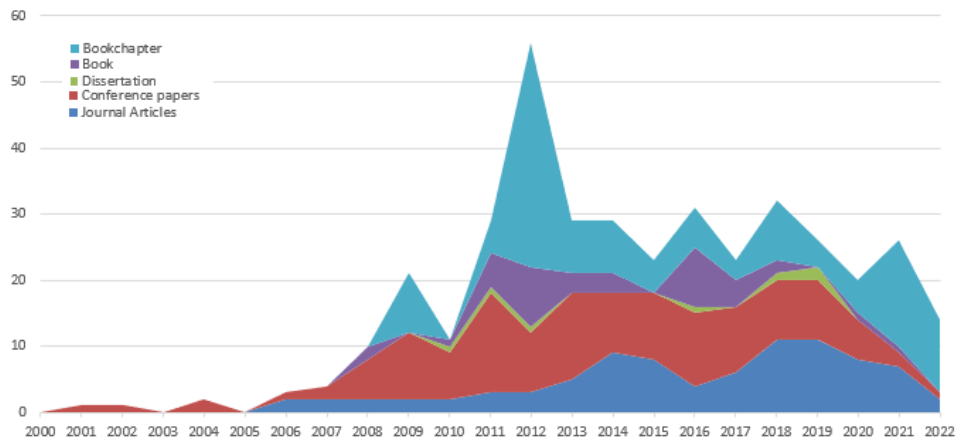
**KF3:** Vannak meglévő projekthez kapcsolódó kockázati tényezők, amelyek kiterjeszthetők az ALM ütemezési problémákra, hogy belefoglalják a nem tervezett feladatok megjelenését is. A nem tervezett többlettevékenységek magas aránya miatt ALM-specifikus kockázatok jelennek meg a projektmenedzsmenthez képest. A nem tervezett tevékenységek erőforrásokra, költségekre és időzítésre gyakorolt hatása befolyásolhatja a megvalósíthatóságot és az ütemezési teljesítményt.

## 4 Eredmények, a kutatás tézisei

**KK1:** *Hogyan készíthető a rendelkezésre álló tudományos irodalom alapján olyan tervezési modell, amely az alkalmazás életciklus-menedzsment (ALM) problémáját reprezentálja, és amely felhasználható ütemezési feladatok módszertani fejlesztéséhez?*

A kutatásom a rendelkezésre álló ALM-specifikus tudományos irodalom alapos szisztematikus szakirodalmi áttekintésével vizsgálja az ALM definíciójának meglétét. 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 definíciója" kulcsszókereső találatait tartalmazza. Ezután egy kritikai elemzés folytattam a meglévő alkalmazás életcikluskezelési (ALM) definíciók feltárását. A keresés összesen 3230 forrást vizsgált meg, majd szűkítette le 76 relevánsra, amelyek explicit információkat tartalmaznak. Mivel az ALM egy, a különböző támogatószoftver beszállítók által erősen befolyásolt terület, és ezért nagymértékben támaszkodik nem akadémiai forrásokra is, ezért ez a tanulmány kizárólag lektorált forrásokra korlátozódik, mint például folyóiratcikkek, konferencia-kiadványok, könyvek és könyvfejezetek, PhD értekezések, hogy fenntartsa a megfelelő minőségi kritériumot a tudományos munkához méltóan.

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.



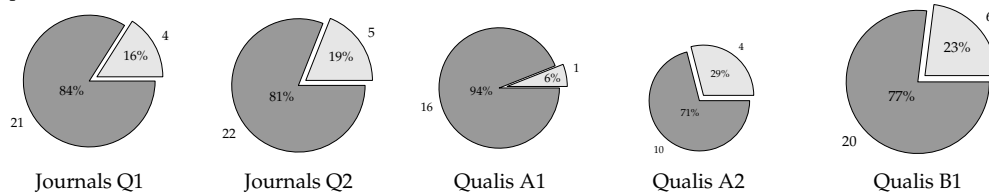
2. ábra. Cikkek, konferenciaelőadások, értekezések, könyvek és fejezetek éves megoszlása a vetítés utáni években

A források minőségi szempontú elemzése után látható volt, hogy a források között észlelhető minőségi különbségek vannak, így két fő kategória lét kijelölve. A *Top Academic* bejegyzések, amelyek a tudományos folyóiratok felső részét 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 az *Extended Academic* rangsorába tartozik ebben az áttekintésben, ideértve a folyóiratok Q3 & Q4 cikkeit, konferencia-előadásokat, PhD értekezéseket, könyveket és könyvfejezeteket.

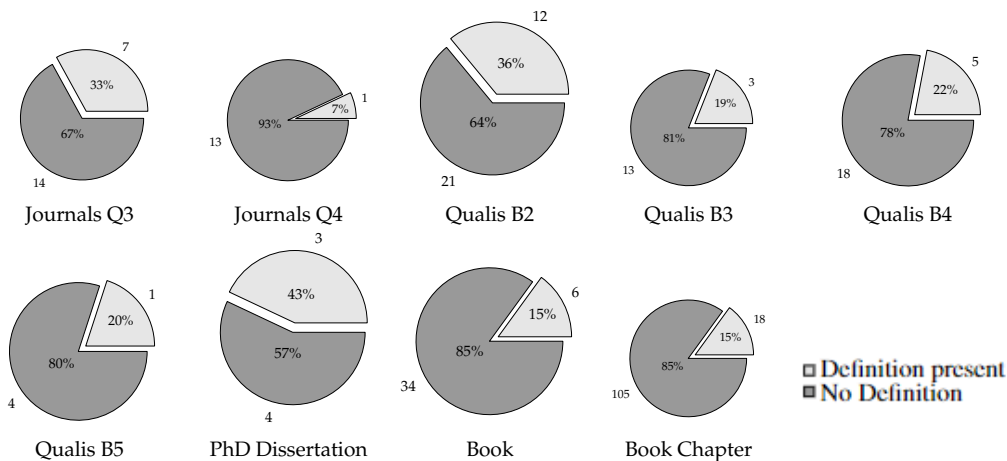
A 2. táblázat azt mutatja, hogy a szűrések és a teljes átolvasás után a források csak nagyon korlátozott információt tartalmaznak kifejezetten az ALM definícióról, összesen 76 releváns forrás tartalmaz explicit ALM definíciót. Ez meglehetősen alacsony szám, és a szerzők több esetben kiemelik az üzleti vonatkozású erős kapcsolatokkal és a gyártó-specifikus ALM-definíciós változásokkal való összefüggést azok eszközkészleteinek a meghatározásaként.

Érdekes eredmény, hogy míg a *Top Academic* forrásokban a leggyakrabban az E) definíció szerepelt, amely az ALM-et úgy tekinti, *mint paradigmát, amely kormányzást, fejlesztést, üzemeltetést/karbantartást tartalmaz*, ami kiemeli az absztrakció magas szintjét tulajdonképpen. Az *Extended Academic* 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. A 3. ábrán a fellelt definíciók megoszlása látható.

Top Academic entries:

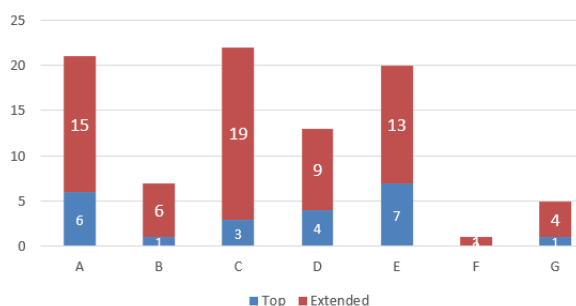


Extended Academic:



2. táblázat. ALM definition availability in Top and Extended Academic ranking sources

A szisztematikus irodalomkutatás során talált ALM definíciókat a 3. táblázat foglalja össze, amely tartalmazza az azonosított ALM definíciók hatókörét a legfontosabb kiemelésével bal oldalon, valamint a fő forrásokat hivatkozásokkal és tartalmát a táblázat jobb oldalán angol, eredeti nyelven.



3. ábra. Summary of various definition scopes present in included entries

Emellett a szisztematikus szakirodalmi áttekintést követően kritikai elemzést folytattam az egységes definíció felállítására. Ez a megközelítés a definíciók kombinált és finomított szintézisét adja, amellyel egy egységesített, a módszertani kutatáshoz széles alapot nyújtó komplex definíciót készítettem.

Scope: ALM is...	Definition
A) a process for SW PLM/SDLC	Product life-cycle Management (PLM) and its equivalent in software, namely application life-cycle management (ALM), are the overall business process that governs a product or service from its inception to the end of its life in order to achieve the best possible value for the business of the enterprise and its customers and partners. PLM/ALM combines processes, people, and tools for the effective engineering of products—from their inception until the end of service. It involves tacit knowledge of experts and explicit knowledge, codified in procedures, process, and tools. PLM/ALM stretches from know-how to know-what and know-why. (Lachainer, 2011), (Ebert, 2013), (Gatrell, 2016)
B) SW development AND maintenance.	Application Lifecycle Management (ALM), a widely-used lifecycle for software development and maintenance. (J. Rossberg, 2012), (Ramler, 2012)
C) Artefact management tool for SDLC.	ALM “has emerged to indicate the coordination of activities and the management of artefacts (e.g., requirements, source code, test cases) during the software product’s lifecycle” (Kääriäinen and Välimäki, 2009), Gatrell (2016) The coordination of development lifecycle activities, including requirements, modeling, development, build and testing, through the following: 1. enforcement of the processes that interconnect these activities; 2. management of relationships and links between the development artefacts used or generated by these activities; and 3. reporting on progress of the development effort as a whole. ALM is often seen as a framework that aims at synchronizing all the lifecycle activities instead of focusing on any specific lifecycle activity” (Schwaber 2006)
D) an SDLC extended with phases after development.	ALM is the product lifecycle management of computer programs that is a wider approach than the SDLC, which is limited to the phases of the typical software development stages. In contrast, ALM defines stages after the development lifecycle as well. (OGC ITIL, 2002),(Arya, 2011), (Chappell, 2011)
E) a paradigm: governance, development, operation/maintenance.	Application Lifecycle Management (ALM) is a recent paradigm for integrating and managing the various activities related to the governance, development and maintenance of software products. ALM is a combination of three functions: governance, development and operations, and three milestones: (start of) ideation, deployment and end-of life. (Chappell, 2008) (Rossberg, 2014)
F) ALM is a service for after development part only	application management refers to the lifecycle-oriented control of the problem resolution process for operational application systems excluding any fundamental application development services. (Kueper, 2011)
G) ALM for quality assurance	Establishing a standardized development-to-release workflow, often referred to as the ALM process, is particularly critical for organizations in their efforts to meet tough IT compliance mandates. (Tracy, 2006)

### 3. táblázat. Összegző táblázat az ALM definíciókról

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

1. *Az ALM hatásköre.* Az ALM egy holisztikus megközelítés a szoftveralkalmazások kezelésére azok teljes életciklusa során, a kezdetektől azok kivonásáig. Különböző tevékenységek és munkatermékek integrálásával és menedzselésével valósul meg 3 funkcióval, nevezetesen a kormányzás, fejlesztés és üzemeltetés, ahova beleértjük a karbantartást is. Az kormányzás egy átfogó irányítási tevékenység az ALM teljes élettartama alatt; jelentősége azonban nagyobb a folyamat elején, a befolyásoló tényezői miatt. A fejlesztés leginkább a klasszikus SW fejlesztési megközelítéshez kapcsolódik, amelyek főbb K+F tevékenységet foglalnak magukban. A működtetés és a karbantartás meglehetősen hasonló egy szolgáltatáshoz; mivel azonban ebben a fázisban a hibajavítás mellett további fejlesztési feladatok is előfordulhatnak különböző méretekben, ezen a 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 kivonását.
3. *Fő összetevők.* Az ALM alapvető összetevői 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 agilis, részben vagy teljesen hibrid megközelítésként alkalmazva. Jelenleg azonban nincsenek az ALM-hez kapcsolódó, bizonyítottan optimális módszertanok vagy keretrendszerek.
6. *Rugalmasság és alkalmazkodóképesség.* Az ALM ütemezési módszereinek rugalmasnak és adaptívnek 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 üzemeltetés karbantartási szakaszában is megtörténik. Ezek a további változtatási kérések az egyszerű feladatok méretétől egészen a kisebb alprojekt méretekig terjedhetnek.
7. *Teljesítmény mérés és mutatók.* Az ütemezéshez kapcsolódó kulcsfontosságú teljesítménymutatók (KPI<sup>5</sup>-ok) mérése és nyomon követése az ALM-ben szintén 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.

A **KK1** kutatási kérdés az elérhető ALM-definíciókra és jellemzőikre összpontosított, valamint arra, hogy egy közös definíció miként nyújthat szilárd alapot a jövőbeli kutatásokhoz. A fellelt definíciókat és azok összefoglalását bemutattam,

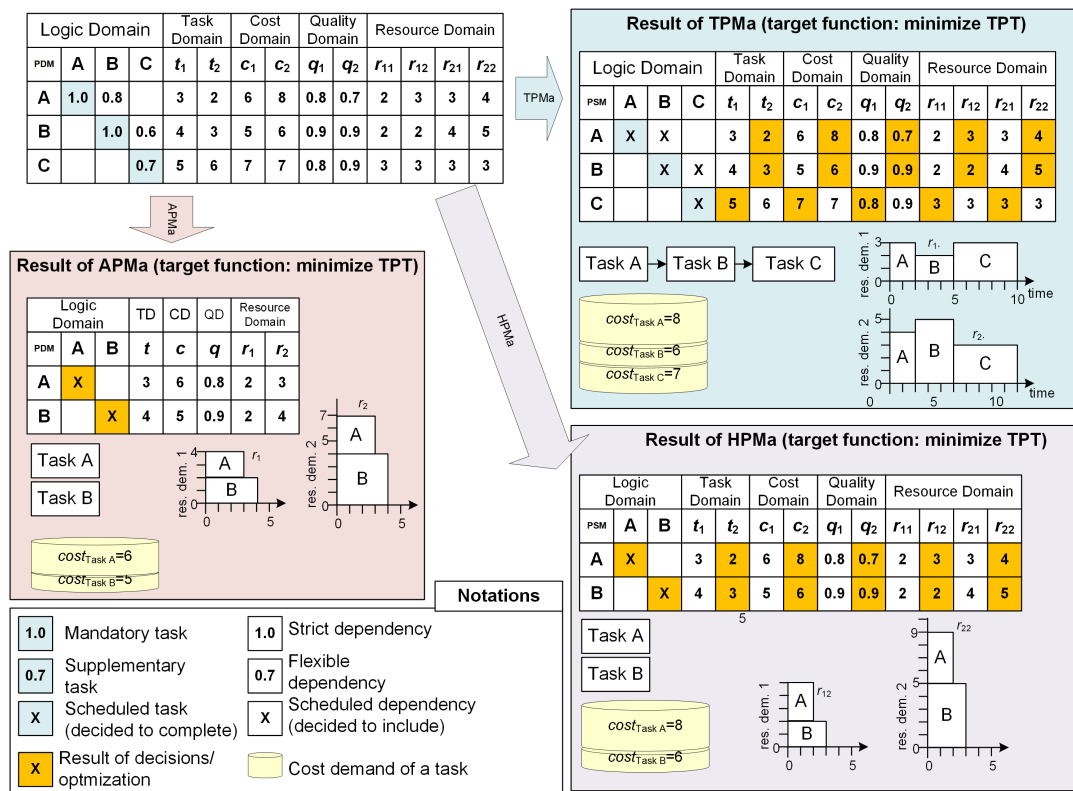
<sup>5</sup>Key Performance Indicator - Kulcsfontosságú Teljesítménymutató

összesen 7 különböző értelmezést tükrözve. A definíciók lefedik az alkalmazások életciklus-menedzsment hatóköreinek és tartalmának jelenlegi akadémiai megértését, az egységes ALM definíció részleteivel a jövőbeni módszertani kutatásokhoz is alapot adnak, megfelelve a **KF1**-nek. Következésképpen a **KF1** elfogadható.

**KK2:** A jelenlegi projektmenedzsment módszertanok (TPM, APM, HPM) eredményeznek-e megvalósítható megoldásokat az ALM környezetben? Hogyan teljesítenek az ALM problémák ütemezésében?

### Ágens alapú projektmenedzsment megközelítések megvalósítása

A TPM, APM és HPM esetében egy ágens<sup>6</sup> van programozva 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 4. ábrát az ügynökök ütemezési problémákkal kapcsolatos viselkedésének összehasonlításához.



4. 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.)

A projekt domainmátrix (PDM) nevű struktúrában való reprezentáció, amely 3 kötelező tartományt tartalmaz, nevezetesen logikai tartományt (LD - Logic Domain),

<sup>6</sup>Egy programozott szoftver-ügynök, amely egy specifikus feladatot hajt végre.

időtartományt (TD - Time Domain) és költségtartományt (CD - Cost Domain), valamint két kiegészítő tartományt, nevezetesen a minőségi tartományt (QD - Quality Domain) és erőforrás tartomány (RD - Resource Domain) lásd a 4. ábrát példaként (Kosztján, 2022).

Ágensek	TPMa		APMa		HPMa	
Érdektelt fél	Előnyök	Hátrányok	Előnyök	Hátrányok	Előnyök	Hátrányok
Vevő	Magas minőség Teljes terjedelelem	Legjobb Alacsonyabb megvalósíthatóság	Legrovidebb	Alacsonyabb minőség Kevesebb tartalom	Legmagasabb megvalósíthatóság Legjobb ütemezés	Nincs többlet megvalósítás
Menedzsment Felkészültség	Alacsonyabb erőforrásigény adott időben	Magasabb költség	Alacsonyabb költség	Magasabb erőforrásigény adott időben Magasabb erőforrásigény	Legmagasabb megvalósíthatóság Legjobb ütemezés	Nincs többlet megvalósítás

4. táblázat. Az ágensek a érdekelt felek alkalmazásából származó előnyök és hátrányok tekintetében

A különböző ügynökök megvalósítása különböző célokat szolgálhatnak, lásd a 4. összehasonlító táblázatot.

### Szimulációs keretrendszer

Az ütemezési problémák különböző ágensekkel (TPMa, APMa, HPMa) való megvalósíthatóságának meghatározására 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 a Monte Carlo szimuláció (MCS) lett választva, amely az egyik leggyakrabban alkalmazott kockázatkezelési módszer. Ez egy hasznos és bevált technika a projekt kockázatainak és bizonytalanságainak szimulálására. Az MCS-ben a kockázati hatások, például a késések, a költség-túllépések és a túlmunka 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, 2020a), 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 meghiúsul és megvalósíthatatlan projektté válik.

### Adatbázisok

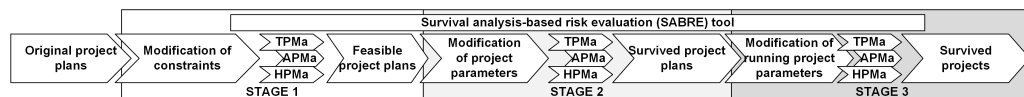
A megfelelő projekttervek projektadatbázisból való kiválasztása nehézkes volt, mivel egyik ismert projektgenerátor (például ProGen (Kolisch és Sprecher, 1997), RanGen I (Demeulemeester, Vanhoucke és tsai., 2003) és II (Vanhoucke és tsai., 2008)) sem nyílt projektadatforrás (pl. mint MMLIB (Peteghem és Vanhoucke, 2014) és PSP-LIB (Kolisch és Sprecher, 1997)) különbözteti meg a kötelező és kiegészítő feladatokat, vagy nem kezeli a szigorú és rugalmas függőséget. Ezért nincsenek a feladat befejezéséhez vagy a feladat függőségeihez kapcsolódó pontszámértékek. Ennek ellenére a rugalmas függőségek és feladatvégzési prioritások figyelembevétele nélkül

a rugalmas projekttervek nem modellezhetők, mert az alacsonyabb prioritású (kiégészítő) feladatok nem halogathatók, a projektterv nem alakítható át. Mivel még mindig nem létezik valódi projektadatbázis, amely a prioritások empirikus eloszlását vagy a rugalmas függőségeket tartalmazná, ezért a feladatok/függőségek és prioritások kiválasztása normáelosztást követett. Ezért három különböző adatkészlet lett felhasználva.

**Adatkészlet A:** Az adatkészlet kiválasztott adatokat tartalmaz az ismert projekt adatbázisokból: 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.

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

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



5. ábra. The proposed simulation framework

A 5. ábra a szimulációs keretrendszert mutatja. Ebben a szimulációban a kockázati hatások, például a korlátok módosítása, lásd az első szakaszt (Stage 1) és a költség- és időtúllépések, második és harmadik szakasz (Stage 2,3) hatását mérséklik a projektmenedzsment ügynökei. A megvalósítható projektek további tulajdonságait a projektmenedzsment ügynökök kezelik.

### Leíró statisztikák

A 6. 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 6. 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 6(a,c) ábrán a 21-25 és 46-50 projektstruktúrákat). Ebben az esetben a minőségi



követelmények is magasabbak (lásd a 6(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 6(g)-ben). A MANOVA klaszterből csak egy projektstruktúra (49) mutat releváns különbséget a megszorításokban (összevetve a 6(a) és 6(c) és ??(k) részábrákat).

### **Megvalósíthatóság kontra rugalmasság**

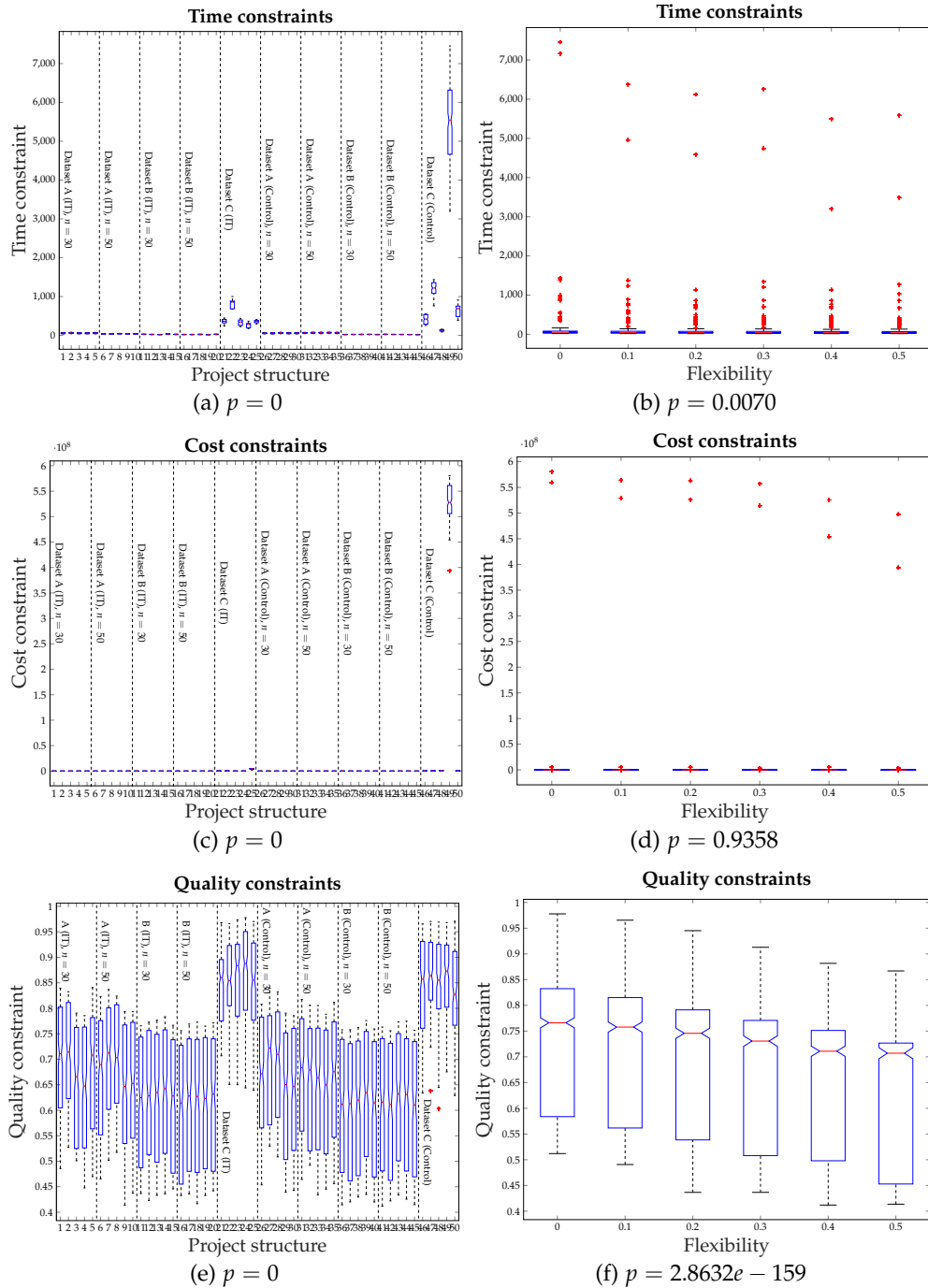
A

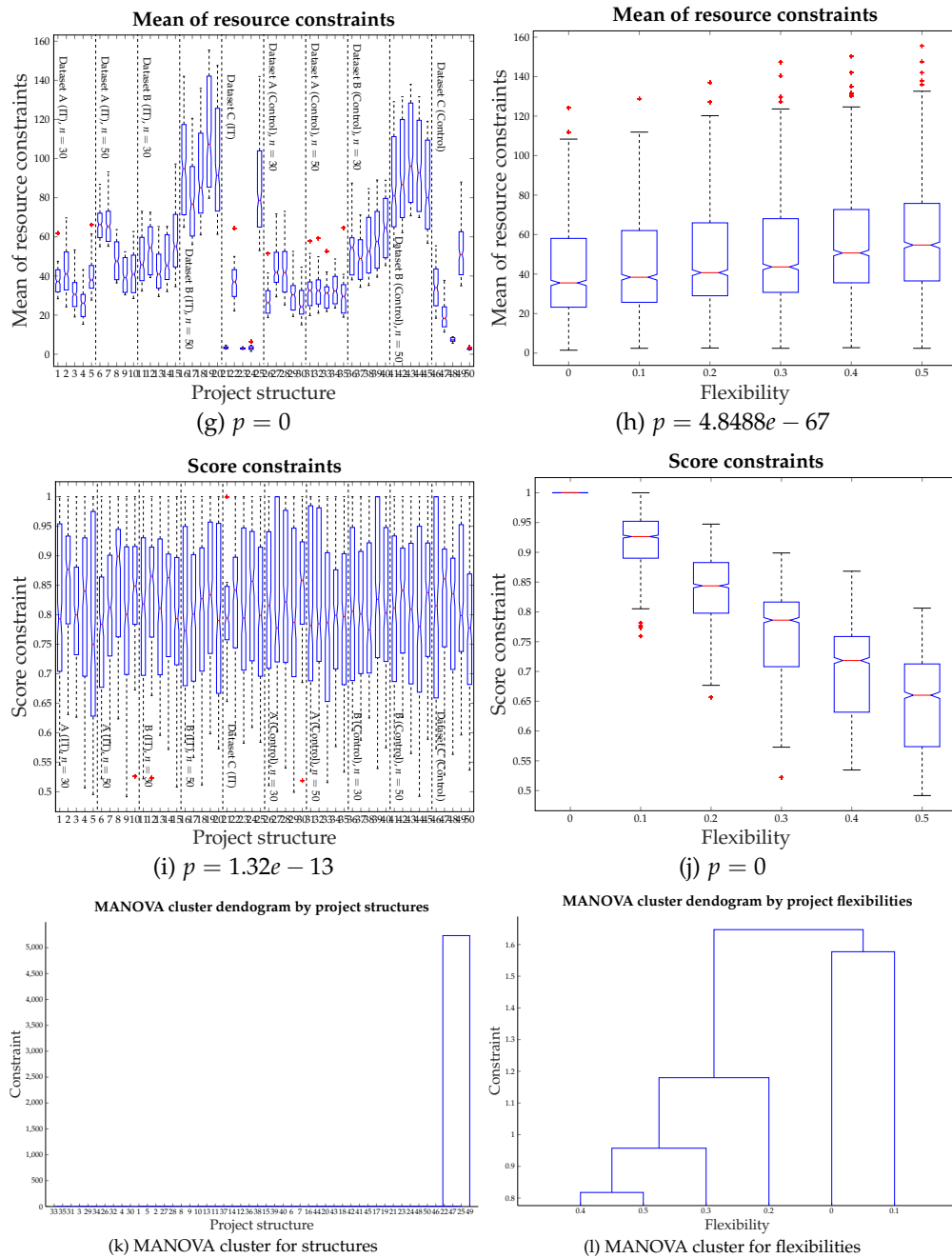
7.(a) ábra a projektmenedzsment ágensek megvalósíthatósági arányát mutatja szakaszonként és rugalmasságonként. A túlélési arány megadja az adott szakaszban a TPMa, APMa vagy HPMa által kezelt megvalósítható projektütemezési problémák arányát. Szakaszról szakaszra egyre kevesebb projekt éli túl a korlátok 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 7(b) ábrát) a TPMa érzékenyebb az igények változásaira, míg a rugalmas megközelítések általában kevésbé érzékenyek (lásd a 7(b) ábrát)), még akkor is, ha a rugalmassági arány magas (lásd a 7(a) ábrát).

A 6(d,f,h,j) ábrával összhangban a 7(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 7(a) ábrát).

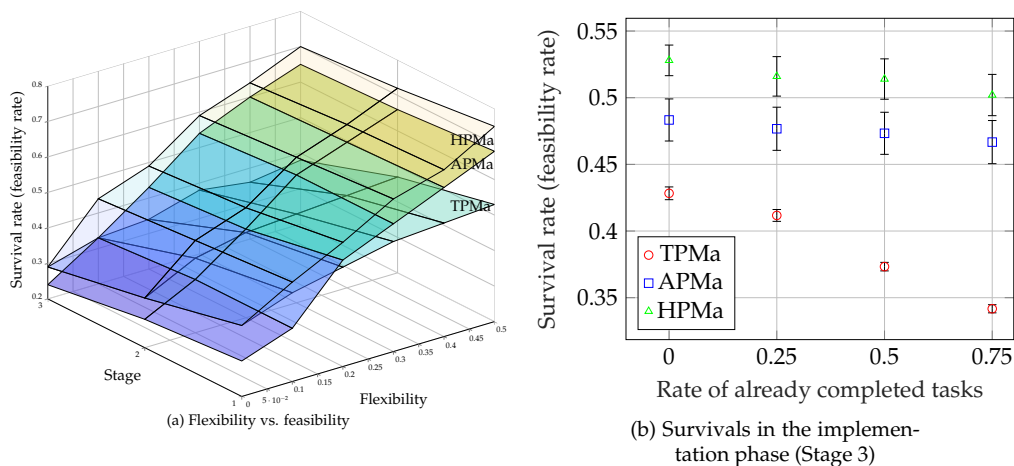
Érdekes eredmény, hogy a HPMa jobban kihasználta a rugalmasság adta lehetőségeket. A HPMa megvalósíthatóbb projekteket tesz lehetővé, mint az agilis megközelítés.

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





6. ábra. A (M)ANOVA eredményei a korlátokra, a projektstruktúrákra és a rugalmasságra vonatkozóan



7. ábra. Projektmenedzsment ügynökök megvalósíthatósági aránya a rugalmasság alapján

**KK3:** *Melyek az ütemezési probléma kockázati tényezői az ALM környezetben? Mely projekttervezési és ütemezési megközelítések enyhítik a legtöbb kockázatot az ALM környezetben? Hogyan befolyásolják az ALM-specifikus kockázati tényezők 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 kockázatok kezelése 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 termékéletről-folyamatot végigkíséri. 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 ad a potenciális kockázati tényezők megjelenésének. Sajnos az ALM kockázati tényezőit még alig kutatták, a szakirodalom elsősorban leszűkített ALM hatóköröket tartalmaz. 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 általános és bevezetéssel kapcsolatos, pl. ALM szervezetváltási szempontból releváns kockázatot, habár az ALM szervezeti adaptációhoz (Akgun és tsai., 2020; Tüzün és tsai., 2019), majd később a működéshez kapcsolódóan is jelentős kockázati tényezőket azonosítottak a szakirodalomban (Cheng, 2010).

A kockázatkezelési megközelítések eltérőek az ALM-környezetben is gyakran

használt agilis megközelítések esetében, mivel az iteratív hurkokkal rendelkező agilis ideológia célja, hogy „korán megbukjon”<sup>7</sup> a fejlesztési belső teszteseteken, és még mindig tudjon reagálni a felmerülő problémákra a fejlesztés sikeres befejeztéig. Buganová 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 korlátozottak, Choetkiertikul és Sunetnanta (2012) egy kockázatgyűjtő és -értékelési eszközt javasolt, amely többnyire az elosztott szoftverfejlesztéssel kapcsolatos kockázatokra összpontosít. Azonban többnyire 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 model értelmezés korlátozásaként, a releváns kockázatokat, például a projektkockázatokat kezeljük, amelyeknek megfelelőnek és elfogadhatónak kell lenniük az ALM modellünk számára. A korlátozás a jövőben az ALM hatókörének további tanulmányozásával feloldható.

Az 5. táblázatban látható a szakirodalomból a projektmenedzsmenttel és az ALM kockázati tényezőkkel kapcsolatos faktorokat, valamint, hogy mely akadémikusok vizsgálták a témákban. Az alábbiakban rövid betekintést adok arra is, hogy ezek hogyan kapcsolódnak az általános projektmegközelítésekhez, az SW-projektekhez és az ALM-hez. Mivel jelen disszertáció középpontjában a módszertani megközelítés áll, így a fő hangsúly az ALM-specifikus nem tervezett tevékenységek kidolgozásán és annak hatásain van.

*A Terjedelműzés (Scope Creep).* Komal és tsai. (2020) azt állítják, hogy a terjedelműzés főként SW-projektekben van jelen, és alapos irodalomkutatással vizsgálja ennek okait. Elmondásuk szerint a szakirodalom szoftvermérnöki és szoftverprojektmenedzsment-szakértői azt állították, hogy a tartalom túlterjedés és a szoftverprojektek kudarcához vezető tényező. Ezenkívül a kritikusok azt állítják, hogy ez szinte minden szoftverprojektben megnyilvánulhat, ami minőségi kompromisszumot, késleltetett ütemezést, megnövekedett költségeket és csökkent ügyfél-elégedettséget eredményez. Madhuri és tsai. (2018) a SW-cégeknél a projektek hatókörének elcsúszását is vizsgálja, és még egy lépést tesz annak megjelenítésére, és javaslatot tesz annak kezelésére egy matematikai modellezési perspektívával a vezető SW-cégeknél, illetve hasonlóképpen Ajmal és tsai. (2022)

<sup>7</sup>Az angol "fail early" agilis elv fordítása.

Kockázati Faktor	Projektmen. releváns	ALM releváns	Elsősorban ALM-ben
terjedelműzés	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

5. táblázat. Projektmenedzsment és Alkalmazás-életciklus menedzsment környezetben megjelenő kockázati tényezők

az építőiparban. Az ALM esetében kétségtelenül jelen van a nem tervezett feladat végrehajtásából adódó tartalmi kúszás, ahogyan Rossberg (2019) is kiemeli könyvében, mint egy ALM-specifikus tényezőt, amelyre az érintetteknek figyelniük kell és aktívan kell kezelniük. A Aiello és Sachs (2016) még az ALM agilis módszertanokat és a DevOps használatát is javasolja a hatókör-csúszással kapcsolatos kockázatok megelőzésére.

*Változás a követelményekben.* A projektmenedzsment a már a projekt elején elérhető követelményekre alapoz, azonban a hagyományos megközelítésben későbbi fázisokban való változtatások nem örvendetesek. A projektmenedzsmentben a gyakorlati követelménykezelés kulcsfontosságú tényező a projekteknél, amit Kossmann (2016) is leír és kifejti könyvében. A Venkatesh és Balani (2016) kiemeli, hogy a követelménykezelés a sikeres szoftverprojektek kulcsa. Tekintettel arra, hogy a nem tervezett, nagy eséllyel követelményváltozással is járó tevékenységek, az ALM-ben ez a kockázat is jelen van, még jóval magasabb előfordulási aránnyal is, mint a Chanda és tsai. (2013) és Rossberg (2019) által hivatkozott projektmenedzsment megközelítésekben.

*Költségkeret túllépés.* A klasszikus projektmenedzsmentben a megnövekedett költségeket leginkább a nem megfelelő tervezés befolyásolja, de a nem tervezett és nem kompenzált tevékenységek is, ami tulajdonképpen az ALM környezethez hasonló helyzetet teremt. Albtoush és Doh (2019) és Jackson (2002) az építőiparban a költségtúllépés kockázati tényezőinek ellenőrzésére és kezelésükre javasol módszert. A fő megállapításuk az, hogy a nem megfelelő változáskezelés és kockázatkezelés költségtúllépésekhez vezethetnek. Ebert (2013) általánosságban kiemeli az ALM rendszer előnyeit a termék- vagy szoftverfejlesztés hatékonyságának növelésében, így a költségvetés kezelésében is. Banjanin és Strahonja (2018) a kockázati tényezőkre az ALM-területen egy keretrendszer felállítását javasolják a költségvetési túllépések kockázatának csökkentésére már a portfólió kezelés szintjén.

*Ütemezési késések.* A Majerowicz és Shinn (2016) az ütemezési késések és a költségtúllépések közötti összefüggést vizsgálja összetett projektekben. Sok projektmenedzser általában egyetért abban, hogy a költségtúllépések közvetlenül összefüggenek az ütemezésbeli késésekkel, ez azonban nem felétlenül igaz. Hasonlóképpen, Park (2021) egy tanulmányt folytatott, amely a klasszikus projektmenedzsmentben vizsgálja a 113 építési projekt során fellépő ütemezési késések előfordulási arányát, mértékét és jellemzőit, olyan tényezőket azonosítva, mint például a nem tervezett tevékenységek jelentős hatással vannak a költségek túllépésére és a menetrendi késésekre. Az ALM előnyére Tudenhöfner (2011) felfedi, hogy míg a hagyományos projektekben az ütemezés késik, ha előre nem látható probléma lép fel, amit ki kell javítani, addig az ALM esetében a keretrendszer nagyobb rugalmasságot biztosít a tervezésben és integrált teljesítménymenedzsmentjével. Ezzel szemben a Deuter, Otte és tsai. (2019) párhuzamba állítja az ALM és a PLM integrációt az ütemezési tevékenységekhez, így két megközelítés integrálását javasolja.

*Erőforrás-korlátozások.* A legtöbb projektnek van valamilyen korlátja az

erőforrások rendelkezésre állásában, és az erőforrások hatékony kezelése költségérzékeny cél a klasszikus projektmenedzsment projektjei számára. Mishra (2020) az erőforrás-felhasználást vizsgálja útépitési környezetben a kényszerelmélet (Theory Of Constraint) és a kritikus lánc koncepciók segítségével, hogy javítsa az általános teljesítményt a hatékony ütemezés érdekében. A nem tervezett tevékenységekre a pufferstratégiával történő kezelést javasolja. Az erőforrás-korlátos projektek és a többprojekt-ütemezési problémák (RCSP-k és RCMSP-k) tanulmányozása kulcsfontosságú volt az elmúlt harminc évben. Mindkét probléma magában foglalja a tevékenységek lefutását, figyelembe véve azok sorrendjét és a rendelkezésre álló erőforrások korlátait (Issa és Tu, 2020). Szintén jelen dolgozat egy kiterjesztett RCSP mátrix reprezentációs módszert vizsgál az ALM probléma megoldására. Aiello és Sachs (2016) az ALM-nek írt könyvében arra hivatkozik, hogy az ütemtervben (Rossberg, 2019; Rossman, 2010) tisztázza a nem tervezett feladatokhoz szükséges erőforrásokat és azok hatását.

**Ütemezés megvalósíthatósága** Az erőforrás-korlátos projektütemezési probléma (RCSP) a projektmenedzsment ütemterv megvalósíthatósági elemzéséhez jól ismert terület. Számos akadémikus vizsgálta az alapprobléma kiterjesztését valós életszerű megközelítésekre (Issa és Tu, 2020) a zavarok vagy késések ütemezésére (Rahman és tsai., 2021), rugalmas struktúra (Beek és tsai., 2024; Van der Beek és tsai., 2022) vagy nem tervezett feladatok megjelenés (Koszyán és Szalkai, 2020b). Az ALM terület esetében a nem tervezett feladatok a problématerületen belül várhatóan dinamikusán kezelhetőek lesznek. Ez az ütemterv-adaptáció rugalmasságával, agilis módon (Aiello és Sachs, 2016) történik általában.

**Minőségbeli problémák.** Komal és tsai. (2020) már korábban is felhívték a figyelmet arra, hogy a terjedelmkúszás, lehet az eredeti projektcélok el nemérésének az oka, úgy mint ütemezési késések és költségnövekedések mellett a minőségbeli problémák forrása. Shafqat és tsai. (2022) egy olyan megközelítést dolgoztak ki, amely megmutatja, hogyan lehet figyelembe venni a nem tervezett tevékenységeket adott minőségi szintekkel. Az ő modelljük esetében a tervezési szakaszt iterációkra osztja, hogy a magas minőséget biztosítani tudja. Megvizsgálja továbbá, hogy az új termékek fejlesztésében részt vevő vállalatok hogyan kezelik a „proaktív kockázatkezelést” és a „reaktív gyors tanulást”. Wawak és tsai. (2020) egy szisztematikus irodalomkutatással vizsgálja az építőipar minőséggel kapcsolatos főbb tényezőit, és megállapította, hogy a legtöbb említett termékminőségi tényező megfelelési hatókörrel jellemezhető. Ez azt jelzi, hogy a nem tervezett tevékenységek jelentős hatással vannak a minőségi szintre.

Az ALM esetében Otibine és tsai. (2017) a minőségi összefüggés kérdését vizsgálta, azonban azt találta, hogy az ALM-megoldások előnyben részesítik a szoftverfejlesztési fázisok integrálását, de nem foglalkoznak megfelelően a minőség témájával. A minőség fogalma homályos maradt Akgun és tsai. (2020). szerint



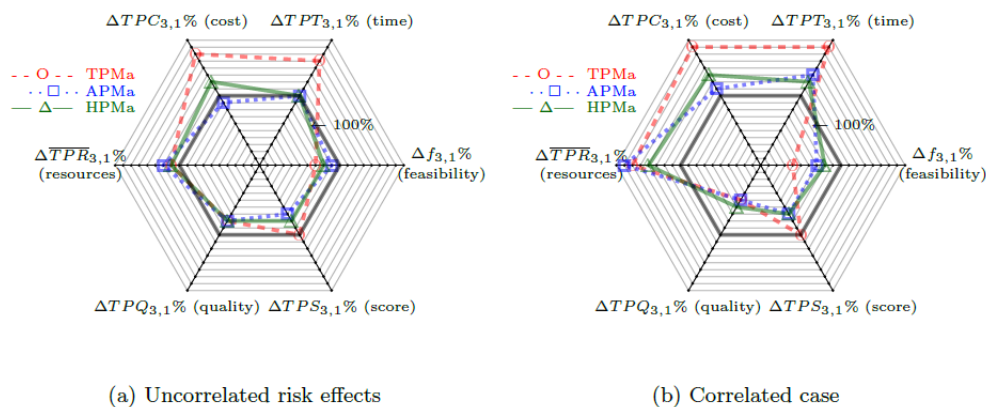
az ALM fő célja a szoftverminőség javítása. Közvetlen vita azonban a további nem tervezett feladatok értékelésének megjelenéséről sem itt, sem más ALM-mel kapcsolatos cikkekben nem szerepel a részletes vitákban.

**A nyomonkövethetőség hiánya.** Míg a tradicionális projektmenedzsmentben ez nem feltétlenül része a hatókörnek, addig az ALM esetében a nyomonkövethetőség kulcsfontosságú szempont (Corallo és tsai., 2020), egy szisztematikus irodalomkutatás segítségével hat iparágban (szoftver, gyártás, autóipar, automatizálás, repülőgép és légiipar) tárták fel a nyomonkövethetőség és az életciklus közötti kapcsolatot. Akgun és tsai. (2020) szerint az ALM fő előnyére világít rá, hogy ez a funkció a fejlesztési eszközökben ab ovo nyomonkövethetőséget biztosít.

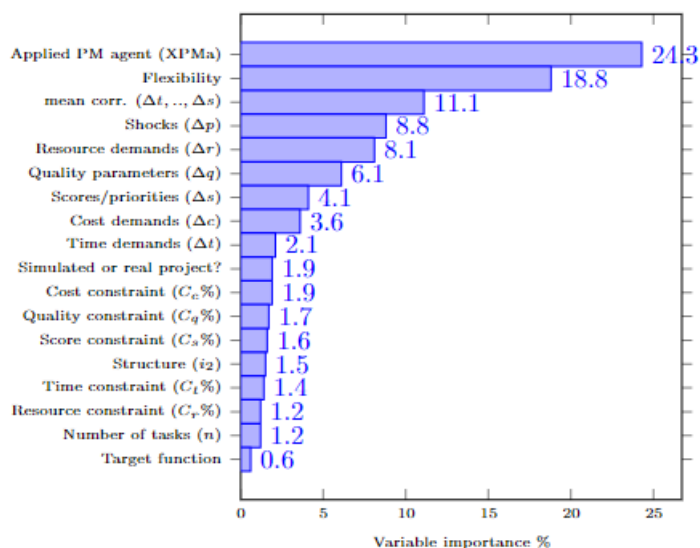
**Verziókezelési problémák.** A verziókezelés szorosan integrálva van az ALM-be, de nem feltétlenül része a klasszikus projekteknek (Kääriäinen és Välimäki, 2008). Az alkalmazás élettartama során számos verzió jelenik meg, amelyek szisztematikus ellenőrzést igényel a szoftverkiadások felügyeletéhez, az előre meghatározott állapotok és referenciaértékek megőrzéséhez a különböző összetevők között, és szükség esetén visszatérhetnek ezekhez az előre meghatározott állapothoz. A verziószabályozás koncepciója széles körben elismert, és a folyamatban lévő kutatások kiterjesztik a verziókezelés hatókörét a forráskód kezelésekre is (Pirklbauer és tsai., 2009). A verziókövető rendszerben megjelenő további feladat miatt szükség van a bevonásra is, amely lehetővé teszi a megfelelő munkatermék- és folyamatkövetést. A verziókezelés helytelen használata azonban ütközésekhez, adatvesztéshez vagy nem szándékos felülírásokhoz vezethet.

### A kockázatsökkentés teljesítménye

A szimulációból a 8. ábra mutatja a feltárt projektmenedzsment megközelítések kockázatsökkentési teljesítményét. Az ideális kockázatsökkentési stratégia minden projekttervet megvalósíthatónak tart.



8. ábra. Projektmenedzsment megközelítések kockázatsökkentési tevékenysége



9. ábra. Változók jelentősége a megvalósíthatóság szempontjából

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 terjedelm-csökkenést 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 módszerek és az 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.

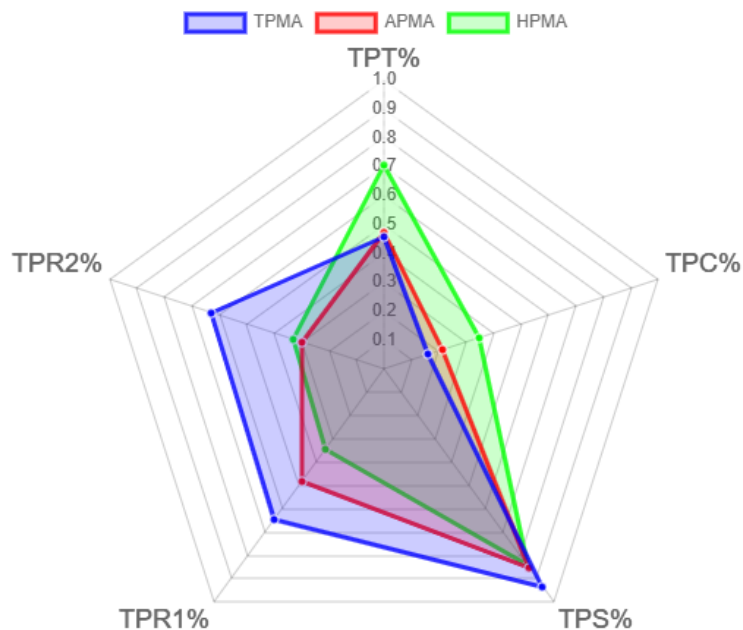
A 9. ábra a projektmenedzsment ágensek hatását mutatja; minden feltárt szerkezeti tulajdonság, mint például a projekt szerkezete és rugalmassága; alacsony szintű kockázati tényezők, mint például a költségek változása ( $\Delta c$ ), időtartam ( $\Delta t$ ), erőforrásigény ( $\Delta r$ ) stb.; és magas 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 projektmenedzsernek 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 kerül.

## Validáció

A szimulációs eredmények alátámasztása érdekében ALM-környezetben validálás készült egy jelenlegi probléma esetén. Az esettanulmányt egy 1871-ben alapított globális autóipari beszállítónál végezték el, amely fékrendszerek, belső elektronika, autóbiztonsági, hajtáslánc- és alvázalkatrészek, gumiabroncsok és különféle egyéb autóalkatrészek gyártására szakosodott vezető vállalat. A szervezet 58 országban működik, teljes árbevétele 33,8 milliárd euró, alkalmazottainak száma pedig megközelítőleg 190 000. Ebben az esetben az elektronikus fékrendszerek szoftveralkalmazása volt a fókuszban, ahol a vállalat csúcskategóriás beszállító, és jól ismert cégekkel versenyez, amely jól mutatja a szervezeti struktúrát és a legfontosabb adatokat az esettanulmányban. Az ALM-kezelés kihívása az évtized elején jelent meg, és a cég strukturális és módszertani szempontból agilis átállással 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



10. ábra. Radardiagram az ALM ágensek teljesítményéhez az esettanulmányban

erre az esetre. Amint az a 10. 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

6. táblázat. Összefoglaló táblázat az ALM ágensek teljesítményéről

A 6 táblázatban látható összefoglaló, amely utal az esettanulmányok eredményeire. Ezen eredmények és észrevételek bemutatásra és megvitatásra kerültek a vállalatvezetéssel és a szervezetfejlesztési projektek szakértőivel.

## 5 A kutatás tézisei

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

**T1:** Az egységes ALM definíció a szakirodalmi áttekintés alapján készült: **ALM egy holisztikus megközelítés a szoftveralkalmazások kezelésére azok teljes életciklusa során, a kezdetektől a kivonásig.**

Különbéféle tevékenységek és munkatermékek integrálásával és menedzselésével valósul meg, amelyek 3 ALM-funkcióhoz kapcsolódnak, úgymint a *kormányzás, fejlesztés és üzemeltetés, beleértve a karbantartást*. A kormányzás az ALM teljes élet-tartama alatt átfogó irányítási tevékenység, azonban jelentősége a kezdeti fázis-ban nagyobb a befolyási tényezője miatt. A fejlesztés leginkább a klasszikus SW

fejlesztési feladatokhoz kapcsolódik, amelyek a főbb K+F munkákat tartalmazzák. A működés és a karbantartás meglehetősen hasonlít egy szolgáltatáshoz. Egyedivé teszi azonban, hogy ebben a fázisban a hibajavítás mellett további nem tervezett feladatok is megjelenhetnek különböző méretekben.

Az ALM-nek három fő ALM-mérföldköve van: *Ötlet, Beüzemelés és Élettartam vége* és 7 fázis van, beleértve a *követelmények összegyűjtését, tervezését, fejlesztését, tesztelését, beüzemelését, karbantartását és visszavonást*.

Az ALM alapvető összetevői az életciklus támogatására szolgálnak olyan folyamatokkal és eszközökkel, mint a *verzióvezérlés, problémakövetés, folyamatos integráció és telepítési automatizálás*. Ezek az összetevők döntő szerepet játszanak az ütemezésben és az erőforrások elosztásában.

Ez az ALM definíció használható arra, hogy egy mátrix alapú projekttervezési modellt az alkalmazás életciklus-kezelési problémáinak ábrázolására tegyen lehetővé. Megfelel a megújuló és nem megújuló erőforrások, az idő, a költségek és a minőség igényeinek egy és több végrehajtási móddal.

**T2:** Megmutattam, hogy az ALM probléma egy kiterjesztett projektmenedzsment nézet, amely a fő fejlesztési fázis után nem tervezett feladatokat tartalmaz. A nem tervezett feladatok kezelését már a szerződéses részben meg kell határozni, hogy azonosítani lehessen a kezelés rugalmasságát, és ennek megfelelően melyik PM megközelítést alkalmazzuk. A feltételek alapján a következők javasolhatók felhasználásra:

1. Hagyományos PM-megközelítés: a teljesítmény további feladatok tervezésével inkrementális modellé válik. Több módú végrehajtás lehetséges, azonban a fix végrehajtási sorrend miatt további prioritások nem tarthatók be. Akkor érdemes alkalmazni, ha a kezdettől fogva tartalmaz végrehajtási puffert, különben a negatív hatások csak többmódusú megközelítéssel csökkenthetők.
2. Agilis PM megközelítés: a sprint szintű ütemezés nem változtatható, mivel a nem tervezett feladatok ezen a szinten nem engedélyezettek. Csak magasabb szintű tervezéssel a sprintek között lehetséges a következő tervezési munkamenet átrendezése a kijelölt prioritásokkal.
3. Hibrid PM megközelítés: a leginkább megengedhető esetben a több módú végrehajtás megengedett, és a nem tervezett feladatokhoz is lehet prioritásokat rendelni.
  - A, A nem tervezett feladatokat a rendszer változtatási kérésként kezeli, és közvetlenül kompenzálhatók és végrehajthatók.
  - B, A nem tervezett feladatokat keretszerződésben kezeljük, és ezen keretek között teljesítjük.

Összességében a hagyományos elemekkel rendelkező hibrid PM-megközelítés teljesít a legjobban.

**T3:** Az ALM kockázati tényezőket a kibővített modellfeldolgozás során az ütemezési módszertani szempontú szakirodalmi áttekintés során azonosítottam. A további feladatok arányának növelésével a megvalósíthatóság és az ütemezési teljesítmény megváltozik. Azonosítottam azokat a tényezőket is, amelyek befolyásolják az ütemező teljesítményét az ALM területen: A következő kockázatok szintén relevánsak az ALM-ben: tartalomkúszás, követelmények változása, költségvetés túllépése, ütemezési késések, erőforrás-korlátok, probléma megvalósíthatósága és minőségbeli problémák. A leginkább az ALM területen megjelenő kockázati tényezők, mint a nyomon követhetőség hiánya és a verziókezelési problémák, az egyedi beállítás miatt jelennek meg a nem tervezett feladatok megjelenésével.

Kutatási értekezés eredményei a **KT1**-hez felülvizsgálat alatt álló státuszban (2024. február) publikálás alatt van a *European Journal of Information Technology*-ban mint **Jakab, R.** & Kosztyán, Z. T. *The Evolution of Definition in Application Lifecycle Management – A Systematic Literature Review Article with a Critical Analysis*

A **KT2** tézis eredményei publikálásra kerültek, mint Kosztyán, Z. T., **Jakab, R.**, Novák, G., & Hegedűs, C. (2020). *Survive IT! Survival analysis of IT project planning approaches*. *Operations Research Perspectives*, 7, 100170.

## 6 Összefoglalás

A technológia gyors fejlődése megköveteli a szoftverprogramok vagy alkalmazások alapos megértését és hatékony kezelését, amelyek elengedhetetlenek a modern üzleti műveletekhez olyan iparágakban, mint az infokommunikáció, az autóipar, az egészségügy, a repülőgépipar és sok más terület. Az elmúlt évtizedekben páratlan eltolódást láthattunk a szoftveralapú gazdaság felé. A szoftverek létrehozásának és karbantartásának egyre nagyobb gazdasági hatása van. Az *Application Lifecycle Management* (ALM) keretet kínál egy ilyen megoldáshoz, mivel a teljes szoftverkezelési folyamatot a kezdetektől az alkalmazás fejlesztésén és karbantartásán keresztül egészen a kivonulásig menedzseli, így hosszú távú gazdasági megtérülést biztosít a szoftverbefektetések számára.

Összegzésként megállapítható, hogy a disszertáció kitűzött célja teljesült, hogy az ALM tudományos irodalom több eredménnyel bővüljön, amelyek több konferencián, könyvrészlet publikációban és cikk publikáció formájában is megjelentek. Az esettanulmány értékelési folyamatában részt vevő szakemberek a napi rutinton kívüli kérdések feltevésével és megválaszolásával is hangsúlyozták a pozitív hatást, segítette őket a munkájuk módjának újragondolásában, sőt, bizonyított tudományos adatokkal is alátámasztottam.

### 6.1 Hozzájárulás az irodalomhoz

Először is meg kell jegyezni, hogy az ALM-mel kapcsolatos szakirodalom továbbra is szükséges, és több szempontból is bővülésre számít, mivel döntően beszállítóvezérelt

területként indult, és egyértelmű vagy egybehangzóan elfogadott ALM-definíció sem létezik még.

1. A disszertáció első részében a szakirodalmi áttekintése fókuszált, tágabb értelemben az ALM részletesebb megismerésére, illetve inkább a meglévő ALM definíciók felkutatására, 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, pl. 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 létezett az ALM szakirodalomban. Az összetett szimulációk valós adatokat 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 a 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 felé.

## 6.2 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 projekteken eddig nem alkalmaztak, és kiterjesztette azt az ALM környezetre is. Az elemzés kimutatta, hogy minden módszernek nemcsak 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 Káosz jelentés (SGL, 2019) eredményeivel, ahol a hagyományos projektmenedzsment megközelítést követő úgynevezett vízvezés 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 hozhat létre.

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 megvalósíthatóakat, a kockázatelemzést követően pedig a leginkább túlélő projekttervekkel rendelkezőket adja meg, de ezek az értékek csak adott célfunkciókra a legjobbak.



A hibrid és különösen az APMA-k jobbak 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 ( $i_2$ ), 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 célfüggetlenséget 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.

A kilátások szerint az autóiipari esettanulmány adatai és információi értékes tudássá válhatnak az adott iparág szervezeti felépítéséhez és tevékenységeik ütemezéséhez. Az intelligens aktuátorok jövője és a SW termékké válásának kihívása (SWaaS) lépésről lépésre vezet az Application Lifecycle Management világához. Az ipar számára fontos a tudományos körök hozzájárulása az ütemezés optimalizálásához és más folyamatokhoz.

## 7. táblázat. A kutatás összegzése - I.

Elem	Ismertetés
<b>KK1:</b>	<i>Hogyan készíthető a rendelkezésre álló tudományos irodalom alapján olyan tervezési modell, amely az alkalmazás életciklus-menedzsment (ALM) problémáját reprezentálja, és amely felhasználható ütemezési feladatok módszertani fejlesztéséhez?</i>
<b>KF1:</b>	Létrehozható egy olyan modell, amely egyesíti a szakirodalomból származó különböző ALM-attribútumokat, amely teljesíti a rugalmas tervezési megközelítést az idő-, költség-, erőforrás- (megújuló és nem megújuló) és minőségi igények figyelembevételével, beleértve a nem tervezett feladatokat is.
<b>T1:</b>	<p>Az egységes ALM definíció a szakirodalmi áttekintés alapján készült: <b>ALM egy holisztikus megközelítés a szoftveralkalmazások kezelésére azok teljes életciklusa során, a kezdetektől a kivonásig.</b></p> <p>Különbféle tevékenységek és munkatermékek integrálásával és menedzselésével valósul meg, amelyek 3 ALM-funkcióhoz kapcsolódnak, úgymint a <i>kormányzás, fejlesztés és üzemeltetés, beleértve a karbantartást</i>. A kormányzás az ALM teljes élettartama alatt átfogó irányítási tevékenység, azonban jelentősége a kezdeti fázisban nagyobb a befolyási tényezője miatt. A fejlesztés leginkább a klasszikus SW fejlesztési feladatokhoz kapcsolódik, amelyek a főbb K+F munkákat tartalmazzák. A működés és a karbantartás meglehetősen hasonlít egy szolgáltatáshoz. Egyedivé teszi azonban, hogy ebben a fázisban a hibajavítás mellett további nem tervezett feladatok is megjelenhetnek különböző méretekben.</p> <p>Az ALM-nek három fő ALM-mérföldköve van: <i>Ötlet, Beüzemelés és Élettartam vége</i> és 7 fázis van, beleértve a <i>követelmények összegyűjtését, tervezését, fejlesztését, tesztelését, beüzemelését, karbantartását és visszavonást</i>.</p> <p>Az ALM alapvető összetevői az életciklus támogatására szolgálnak olyan folyamatokkal és eszközökkel, mint a <i>verzióvezérlés, problémakövetés, folyamatos integráció és telepítési automatizálás</i>. Ezek az összetevők döntő szerepet játszanak az ütemezésben és az erőforrások elosztásában.</p> <p>Ez az ALM definíció használható arra, hogy egy mátrix alapú projekttervezési modellt az alkalmazás életciklus-kezelési problémáinak ábrázolására tegyen lehetővé. Megfelel a megújuló és nem megújuló erőforrások, az idő, a költségek és a minőség igényeinek egy és több végrehajtási móddal.</p>

## 8. táblázat. A kutatás összegzése - II.

Elem	Ismertetés
<b>KK2:</b>	<i>A jelenlegi projektmenedzsment módszertanok (TPM, APM, HPM) eredményeznek-e megvalósítható megoldásokat az ALM környezetben? Hogyan teljesítenek az ALM problémák ütemezésében?</i>
<b>KF2:</b>	A projektmenedzsment megközelítésekhez (TPM, APM, HPM) kapcsolódó mátrix tervezési módszer kiterjeszhető, amely lehetővé teszi, hogy az ütemező ügynök megoldja a problémát, és megvalósítható megoldásokat eredményezzen az ALM környezetben. Az ALM-problémák ütemezhetőek, hogy közel optimális megoldásokat találjanak meghatározott korlátok mellett. A szimulációs keretrendszer létrehozható rugalmas függőségek és nem tervezett feladatok kezelésére.
<b>T2:</b>	Megmutattam, hogy az ALM probléma egy kiterjesztett projektmenedzsment nézet, amely a fő fejlesztési fázis után nem tervezett feladatokat tartalmaz. A nem tervezett feladatok kezelését már a szerződéses részben meg kell határozni, hogy azonosítani lehessen a kezelés rugalmasságát, és ennek megfelelően melyik PM megközelítést alkalmazzuk. A feltételek alapján a következők javasolhatók felhasználásra: <ol style="list-style-type: none"> <li>1. Hagyományos PM-megközelítés: a teljesítmény további feladatok tervezésével inkrementális modellé válik. Több módú végrehajtás lehetséges, azonban a fix végrehajtási sorrend miatt további prioritások nem tarthatók be. Akkor érdemes alkalmazni, ha a kezdettől fogva tartalmaz végrehajtási puffert, különben a negatív hatások csak többmódusú megközelítéssel csökkenthetők.</li> <li>2. Agilis PM megközelítés: a sprint szintű ütemezés nem változtatható, mivel a nem tervezett feladatok ezen a szinten nem engedélyezettek. Csak magasabb szintű tervezéssel a sprintek között lehetséges a következő tervezési munkamenet átrendezése a kijelölt prioritásokkal.</li> <li>3. Hibrid PM megközelítés: a leginkább megengedhető esetben a több módú végrehajtás megengedett, és a nem tervezett feladatokhoz is lehet prioritásokat rendelni. <ul style="list-style-type: none"> <li>• A, A nem tervezett feladatokat a rendszer változtatási kérésként kezeli, és közvetlenül kompenzálhatók és végrehajthatók.</li> <li>• B, A nem tervezett feladatokat keretszerződésben kezeljük, és ezen keretek között teljesítjük.</li> </ul> </li> </ol> <p>Összességében a hagyományos elemekkel rendelkező hibrid PM-megközelítés teljesít a legjobban.</p>

## 9. táblázat. A kutatás összegzése - III.

Elem	Ismertetés
<b>KK3:</b>	<i>Melyek az ütemezési probléma kockázati tényezői az ALM környezetben? Mely projekttervezési és ütemezési megközelítések enyhítik a legtöbb kockázatot az ALM környezetben? Hogyan befolyásolják az ALM-specifikus kockázati tényezők a megvalósíthatóságot és az ütemezési teljesítményt?</i>
<b>KF3:</b>	Vannak meglévő projekthez kapcsolódó kockázati tényezők, amelyek kiterjeszthetők az ALM ütemezési problémákra, hogy belefoglalják a nem tervezett feladatok megjelenését is. A nem tervezett többlettevékenységek magas aránya miatt ALM-specifikus kockázatok jelennek meg a projektmenedzsmenthez képest. A nem tervezett tevékenységek erőforrásokra, költségekre és időzítésre gyakorolt hatása befolyásolhatja a megvalósíthatóságot és az ütemezési teljesítményt.
<b>T3:</b>	Az ALM kockázati tényezőket a kibővített modellfeldolgozás során az ütemezési módszertani szempontú szakirodalmi áttekintés során azonosítottam. A további feladatok arányának növelésével a megvalósíthatóság és az ütemezési teljesítmény megváltozik. Azonosítottam azokat a tényezőket is, amelyek befolyásolják az ütemező teljesítményét az ALM területen: A következő kockázatok szintén relevánsak az ALM-ben: tartalomkúszás, követelmények változása, költségvetés túllépése, ütemezési késések, erőforrás-korlátok, probléma megvalósíthatósága és minőségbeli problémák. A leginkább az ALM területen megjelenő kockázati tényezők, mint a nyomon követhetőség hiánya és a verziókezelési problémák, az egyedi beállítás miatt jelennek meg a nem tervezett feladatok megjelenésével.

## 7 A szerző kapcsolódó publikációi

### Nemzetközi cikkek

- Koszttyán, 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
- Koszttyán, 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.**, Kosztyán, 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

- Koszttyán, 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)

## Hivatkozások

- 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 Šimičková, 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 (2020a). „Multimode resource-constrained project scheduling in flexible projects”. *Journal of Global Optimization* 76.1, 211–241. old.
- Kosztján, Zsolt T és Szalkai, István (2020b). „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, Hélène (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, Jörg és Pisla, 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 és 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 és 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.
- Rossmann, 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.
- SGI (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>.
- Van der Beek, T, Van Essen, JT, Pruyn, J és Aardal, K (2022). *Exact solution methods for the resource constrained project scheduling problem with a flexible project structure*.
- 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, Žanesa és Vukomanović, Mladen (2020). „Understanding the key quality factors in construction projects—A systematic literature review”. *Sustainability* 12.24, 10376. old.