

BUDAPESTI GAZDASÁGI EGYETEM

PÉNZÜGYI ÉS SZÁMVITELI KAR

SZAKDOLGOZAT

Korek György  
Nappali tagozat  
Gazdaságinformatikus  
Logisztikai informatikus

2021

BUDAPESTI GAZDASÁGI EGYETEM  
PÉNZÜGYI ÉS SZÁMVITELI KAR

Low code platformok szerepe az üzleti  
alkalmazásfejlesztésben

Belső konzulens: Mészáros György

Külső konzulens: Szigeti Szabolcs

Korek György

Nappali tagozat

Gazdaságinformatikus

Logisztikai informatikus

2021

# NYILATKOZAT

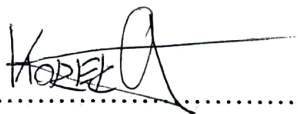
Alulírott **Korek György** büntetőjogi felelősségem tudatában nyilatkozom, hogy a szakdolgozatomban foglalt tények és adatok a valóságnak megfelelnek, és az abban leírtak a saját, önálló munkám eredményei.

A szakdolgozatban felhasznált adatokat a szerzői jogvédelem figyelembevételével alkalmaztam.

Ezen szakdolgozat semmilyen része nem került felhasználásra korábban oktatási intézmény más képzésén diplomaszerezés során.

Tudomásul veszem, hogy a szakdolgozatomat az intézmény plágiumellenőrzésnek veti alá.

Budapest, 2021. november 26.

A handwritten signature in black ink, appearing to read 'KOREK György', written over a horizontal dotted line.

hallgató aláírása

# Tartalom

1. Bevezetés.....	2
1.1 Témaválasztás.....	2
1.2 Szakdolgozat tartalma.....	3
1.3 Téma aktualitása .....	4
1.4 Téma jelentősége .....	5
2. Elméleti háttér .....	7
2.1 A Low Code fogalom ismertetése .....	7
2.1.1 A Low Code és a No Code közti fontos különbség.....	8
2.2 Low Code Development Platformok ismertetése .....	9
2.3 A Microsoft PowerPlatform bemutatása .....	12
2.3.1 A Microsoft PowerApps bemutatása röviden.....	13
2.3.2 A Microsoft PowerAutomate bemutatása röviden .....	16
2.3.3 A Microsoft PowerApps alkalmazások típusai .....	19
3. Szoftverfejlesztés a gyakorlatban.....	21
3.1 Rapid Application Development .....	21
3.2 Általános nehézségek szoftverfejlesztés közben .....	23
3.3 Milyen megoldást nyújt a nehézségekre a Low Code .....	26
4. Esettanulmányok .....	27
4.1 Esettanulmányok ismertetése röviden .....	27
4.2 Esettanulmány: GDPR nyilatkozatok kezelése - PowerAutomate .....	28
4.3 Esettanulmány: IT eszközmenedzsment alkalmazás - PowerApps .....	35
4.4 Esettanulmányok rövid összefoglalása .....	46
4.5 Low Code határai.....	47
5. Mélyinterjú .....	47
5.1 Interjú ismertetése.....	47
5.2 Interjúalanyok bemutatása .....	48
5.3 Interjúk eredményei, következtetés .....	49
5.4 Interjúk összefoglalása .....	52
6. Eredmények, összefoglalás .....	52
7. Irodalomjegyzék.....	55
8. Ábrajegyzék .....	59
9. Mellékletek.....	60

# 1. Bevezetés

## 1.1 Témaválasztás

Szakedolgozatom az üzleti alkalmazásfejlesztésről szól, amiben ismertetem korunk egyik új alkalmazásfejlesztési lehetőségét a low code development platformokat. Szakedolgozatom témáját személyes érdeklődésem és korábbi munkatapasztalataim ihlették. Több szervezetnél is dolgoztam már szakmai gyakornokként, ami által lehetőségem nyílt mélyebben megismerni, és összehasonlítani a szakedolgozatomban mélyebben taglalt alkalmazás és szoftverfejlesztést. Többek között szoftverfejlesztő gyakornokként dolgoztam egy multinacionális autóiipari beszállítónál, ahol főként .NET technológiákban fejlesztettem, valamint egy multinacionális gyógyszeripari vállalatnál, ahol IT és szoftverfejlesztő gyakornokként eltöltött idő során többnyire a Microsoft PowerPlatform adta low code keretek között fejlesztettem üzleti célmegoldásokat Európa több országába, mint többek között szervezeti munkavállalóknak szánt egyszerűsített CRM alkalmazást vagy összetett eszközzel rendelkező rendszert az országok IT csoportjai számára. Mindezen által megtapasztaltam két eltérő iparágban tevékenykedő multinacionális cég szoftvermegoldás igényeit, melyeket véleményem szerint érdemes elemezni, és egymással párhuzamba állítani. A különböző rendszerek egymástól eltérő eszközöket kínálnak, lehetőségeket biztosítanak, különféle szakmai tudást és erőforrást igényelnek. Mivel az informatika egy folyton változó szektor, ezért sokszor gondolkoztam azon, hogy melyik irányba érdemes több érdeklődést, energiát fektetni, így személyes érdeklődésem volt a további meghatározó tényező témám kiválasztásakor.

Szakedolgozatomban nem bocsátkozom mély technikai összehasonlításba a programozási nyelveket tekintve, valamint az összehasonlításom is a célfelhasználó, és a fejlesztő oldaláról megközelíthető különbségekre korlátozódik. A szakedolgozatom nem foglalkozik ezeken túlmenően részletesen a hagyományos programozási nyelvekkel, megoldásokkal. Továbbá értekezésem részét nem képezi a hagyományos programozási nyelvek multiplatform megoldásai és azok elemzése. Központi kérdésem, hogy az újszerű technológiának nevezhető low code platformok térnyerése minek köszönhető, illetve, hogy a különböző vállalatok eredményessége mennyire alapulhat ezen platformok használatára.

## 1.2 Szakdolgozat tartalma

A szakdolgozatom tematikáját tekintve először ismertetem mi is az a low code development platform, miben tér el a pro code megközelítéstől, megoldástól. Bemutatom a Microsoft PowerPlatformot részletesebben. Bemutatom a PowerApps, és PowerAutomate technológiát és publikus eredményeit, életútját, lehetséges kilátásait, illetve azt, hogy milyen hatással volt a vállalkozások digitalizációs átalakulására az, hogy egyszerűbben tudtak reagálni a sürgős igényekre. Ismertetem a Microsoft low code platformjának a PowerApps és PowerAutomate által alkotott megoldások nyilvánosan elérhető vállalati esettanulmányainak tanulságait. Továbbá véleményem szerint fontos kérdés, hogy milyen szoftverfejlesztésbeli akadályokba ütközik ma sok vállalat, és miért lehet segítség még a tapasztalt fejlesztőknek is egy-egy low code platform megoldás. Mélyebben ismertetem a low code platformok elméleti hátterét. Fejlesztői oldalról is megközelítem a PowerApps összehasonlítását a hagyományos pro code megközelítéssel. Szakdolgozatom második felében bemutatom korábbi munkatapasztalataim alatt elkészített, és Európa szerte használt kettő PowerApps és PowerAutomate megoldásomat részletesen egy-egy esettanulmány formájában, továbbá ismertetem személyes kutatásom eredményét, miszerint bemutatom, hogy miként látja a low code platformok jövőjét kettő különböző multinacionális vállalat senior középvezetője. Szakdolgozatom harmadik felében pedig a low code platformok jövőjéről és kilátásairól írok. Értekezésem fontos felvetése, hogy napjaink IT fejlesztő hiányában miért lenne fontos a low code platformok megismertetése az informatikai érzékkel vagy háttérismerettel rendelkező munkavállalók között szélesebb körben. Esetleg a low code fejlesztés lehet az új excel, amit mindenkinek érdemes lenne elsajátítani? Legvégül pedig röviden összegzem a szakdolgozatom tanulságát és lényegi tartalmát.

Szakdolgozatom az alábbi kiemelt kérdésekre keresi a választ:

- Mi az a Low Code Development, és kiknek nyújt megoldást?
- Megoldást nyújt-e a növekvő digitalizáció, és a globális, képzett szoftverfejlesztő hiány mellett a Low Code a szervezetek szoftveres igényeire?
- Miként változhat meg a szoftverfejlesztés világa az LCDP mellett, megfér-e a Pro Code, és a Low Code egymás mellett?
- Képzett IT szakemberek, fejlesztők miként vélekednek a LCDP-ről?

### 1.3 Téma aktualitása

A 2020 elején elindult koronavírus világjárvány alapjaiban forgatta fel a világunkat, kezdve a munkafolyamatok megszervezésével és elvégzésével. Cégek ezrei és munkavállalók milliói kényszerültek arra, hogy az addig végzett folyamataikat egy teljesen új módon lássák el. (Faraj, Renno, Bhardwaj, 2021)

Az állami intézkedések hatására rendkívül rövid idő leforgása alatt, mindösszesen néhány hét rendelkezésre állásával kellett munkavállalók millióinak világszerte különböző digitális platformokra költözniük, függetlenül attól, hogy az adott szervezet és a benne dolgozó emberek milyen szinten, és milyen képességekkel rendelkeztek a digitális megoldások terén. (Dr Géro, Polacsik, 2020).

2010-es évek második felétől az Európai bizottság új digitális stratégiát hirdetett Európa számára, melynek kiemelt célja, hogy a következő évtizedet 2020-tól Európa digitális évtizedévé formálja, melyben kiemelten támogatja az európai bizottság az új technológiák és a digitalizáció térnyerését. A digitális évtized céljai közé tartozik többek között, az EU-n belüli vállalkozások több mint 90%-át aktívan a digitális eszközök és megoldások felé terelni, valamint a felhőtechnológiákat széleskörűen támogatni. A bizottság tervei szerint a digitális törekvésekből az Európai Unió valamennyi állampolgára és vállalkozása részesülni fog. (Európai Bizottság, 2019).

Klaus Schwab 2017-ben megjelent könyvében (Fourth Industrial Revolution) részletesen taglalja a jelenleg zajló informatikai, technológiai forradalom mindennapi életünkre, munkavégzésünkre, valamint napjaink társadalmára gyakorolt nem elhanyagolható mennyiségű változásait. Mindezt olyan sebességgel következik be, hogy sokkal gyorsabban zajlik, mint a negyediket megelőző ipari forradalmak. Klaus Schwab szerint egy olyan változás küszöbén áll az emberiség, ami alapjaiban változtatja meg az életünket, munkavégzésünket, szokásainkat, valamint az emberek közötti viszonyt is. (Schwab, 2017)

Bojár Gábor a Graphisoft alapító elnöke azonban 2018-as értekezésében inkább tartotta célszerűbb megközelítésnek, hogy Schwab negyedik ipari forradalom megfogalmazása helyett a harmadik informatikai forradalom megfogalmazást használjuk. Ugyanakkor Schwab gondolattal Bojár sem vitatkozik, miszerint minden idők legjelentősebb átalakulásának lehetünk szemtanúi a következő években, évtizedekben. Azonban a Schwab által használt fogalom szerint a negyedik ipari forradalom még rengeteg beláthatatlan hatással és

következménnyel rendelkezik, amit napjainkban legfeljebb csak megbecsülni tudunk, pontos képpel, ismerettel, adatokkal nem rendelkezünk róla. (Bojár, 2018).

Mindezek alapján kijelenthető, hogy napjainkban, és a globális vírusveszély megérkezése idején egy digitalizációs informatikai fejlődés által fűtött korban élünk. Ezzel együtt a COVID-19 okozta átfogó lezárások miatt felértékelődött a digitális megoldások iránti igény, valamint az új digitális megoldások értéke és jelentősége is. A Low code development platformok egyik legfontosabb alappilére, hogy a szoftverfejlesztés idejét jelentősen csökkentve, sokkal elérhetőbbé és gyorsabbá teszi egyes alkalmazások, programok és célmegoldások elkészítését (Marvin, 2014).

Az eddig leírtak alapján láthatjuk, hogy a low code platformok, mint új technológiai megoldások, olyan megoldásokat kínálnak az alkalmazásfejlesztésben, ami tökéletesen szükséges napjainkban. COVID-19 alatt rengeteg munkáltatónak hetek alatt kellett áttérnie a megszokott folyamatokról egy digitális megoldásra, így a vírushelyzet miatt kikényszerített szükséghelyzet rengeteg cég figyelmét terelte a low code platformok felé, melyben megtalálhatták a megoldást arra, miként tudnak gyorsan és hatékonyan digitalizálni egyes folyamatokat, vagy digitális megoldást nyújtani több addig hagyományos „analóg” módon elvégzett munkára minél rövidebb idő alatt.

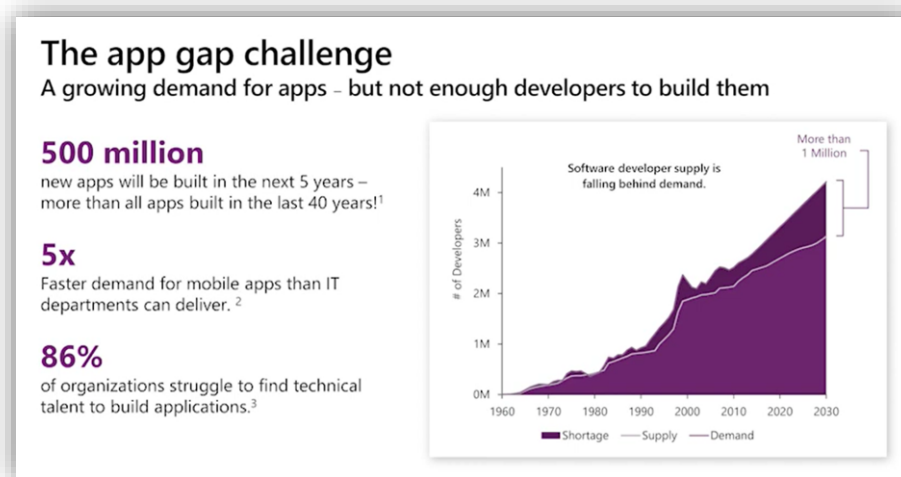
#### 1.4 Téma jelentősége

Az elmúlt években élesen nőtt a kereslet a különböző programozók, szoftverfejlesztők, szoftvermérnökök iránt, akik képesek olyan szoftvereket, alkalmazásokat, célmegoldásokat fejleszteni, amik a vállalatok problémáira nyújtanak modern digitális megoldást. Napjainkban, és az elmúlt években egyre több szervezet megy át digitális átalakulásra, ezt az irányvonalat a 2020-as COVID-19 vírushelyzet csak meredeken növelte. (Pham, 2021)

Ez a digitális átalakulás különböző képzett IT szakembereket igényel. A növekvő igényt nem elégíti ki a kínálat, így a CNBC 2019-es felmérése szerint csak Amerikában több mint 900 ezer betöltetlen, képzettséget igénylő IT állás van. (Liu, 2019)

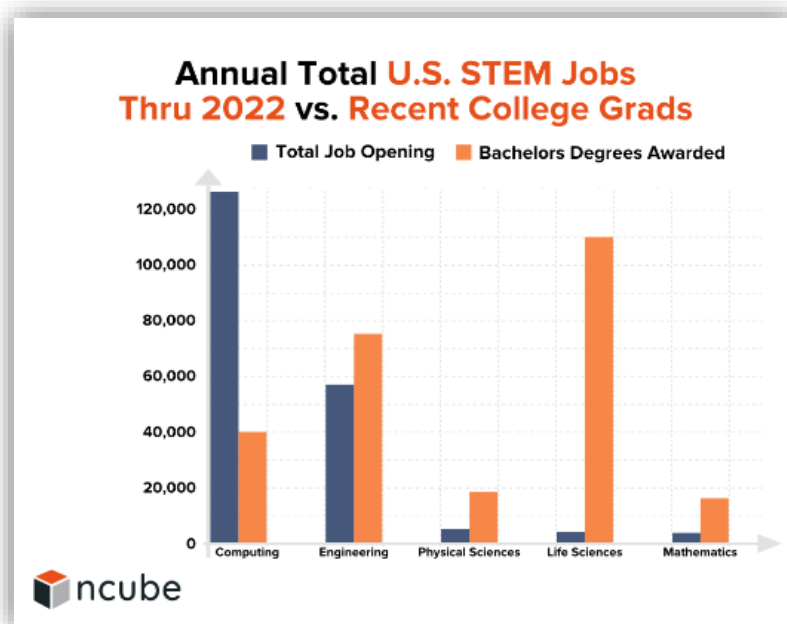
A Microsoft felmérése 1 millió körüli szakember hiányról számol be 2021-es internetes előadásában. (Boggan, 2021)





1. ábra, Növekvő igények a szoftverek iránt, fejlesztő hiány (forrás: Boggan, 2021)

A fentieket megerősítve a több mint 12 éve szoftverfejlesztéssel foglalkozó nCube Ltd. szintén 1 millió fő környékére teszi az USA szoftverfejlesztő hiányát, ez a hiány arányosan megjelenik a világ összes többi földrészén és országában, miközben mind az egyetemről mind az úgynevezett bootcampekről (célirányos programozó képzések) kikerülő fiatal IT szakemberek száma messze alulmarad a piac által igényelt létszámról. (Melnichuk, 2019)



2. ábra, Nyitott pozíciók és várható új munkaerő szektoronként (forrás: Melnichuk, 2019)

Az igények, és a modern szoftveres megoldások kereslete folyamatosan növekszik, a vállalatok nem tudnak elegendő IT szakembert munkába állítani, a Microsoft felmérése szerint a

szervezetek 86%-a küzd azzal az akadállyal, hogy megfelelő IT szakembert találjon különböző szoftverfejlesztési feladatok ellátására. (Boggan, 2021)

Ez a hiány pedig hosszútávon hátráltatja a cégeket abban, hogy folyamatosan fejlődjenek, és a legújabb technológiák mentén piacvezető szerepet tölthessenek be. (Sloyan, 2021)

A Low Code platformok pedig éppen erre nyújthatnak segítséget. Ezen platformok segítségével a meglévő fejlesztők gyorsabban, és hatékonyabban tudnak különböző igényekre megoldást fejleszteni, míg IT orientált, de nem kifejezetten fejlesztő munkakörben dolgozó munkavállalók számára is elérhetővé válik a célmegoldások fejlesztése, ami szükségszerű, mert a Microsoft becslései szerint a következő 5 évben több alkalmazás fog készülni összességében, mint az elmúlt 40 év alatt, beleértve a low code platformok által készített megoldásokat is. (Boggan, 2021)

## **2. Elméleti háttér**

### **2.1 A Low Code fogalom ismertetése**

A szoftverfejlesztésben az egyik legfontosabb tényező az idő és a sebesség. A vállalatok és a szoftverfejlesztéssel foglalkozó csapatok mindig keresik a lehetőségeket és megoldásokat, hogy a fejlesztés idejét lerövidíthessék. Ez a fajta törekvés különösen növekedett az elmúlt években, mind fejlesztői, mind felhasználói, ügyfél oldalról. Ez pedig a low code fejlesztés felé terelte a figyelmet. A low code development nem teljesen új elgondolás, csupán az elmúlt évtizedben kapott pontos megfogalmazást és elnevezést. A low code elképzelés alapelve, hogy lehetővé tegye a gyors fejlesztést, beüzemelést, miközben limitált mennyiségű kézi kódírással van szükség, a munkafolyamat másik fele vizuális felületeken keresztül történik. (Pegasystems, 2021)

A növekvő digitalizációs átalakulások miatt a Forrester elemzése szerint a low code fejlesztés a megoldás napjaink szoftverigényeinek növekedésére, ami egyszerre találkozik a szervezetek és a végfelhasználó igényeivel. A low code fejlesztés másik alapelve, hogy ne csupán tapasztalt fejlesztőknek legyen lehetősége használható célmegoldásokat fejleszteni, hanem olyan üzleti csoportoknak, munkavállalóknak is, akik rendelkeznek némi IT érzékkel, alaptudással. (Bratincevic, Koplowitz, 2021)

### 2.1.1 A Low Code és a No Code közti fontos különbség

Fontos elkülönítenünk a két fogalom és technológia közti különbséget, szakdolgozatom nem foglalkozik a No Code megoldásokkal, de ahhoz, hogy mélyebben elemezhessük a Low Code Platformokat, a technológiát és a részleteit, ahhoz meg kell említeni a No Code és a Low Code Platformok közti eltéréseket, különbségeket. Napjainkban sok vállalat és online portál tévesen, de Low Code alá sorolja a No Code platformokat és megoldásokat, pedig külön csoportokat képviselnek.

**No Code:** Előnye, ahogy a neve is tartalmazza, kódolás és programozási ismeretek nélkül lehetséges alkalmazások, megoldások felépítése. Legelterjedtebb példája a különböző előre felépített sablonokból felépíthető weboldalszerkesztők. Azonban mivel részleges egyedi kódfuttatásra nincs lehetőség, ezért jelentősen korlátozódik a megvalósítható alkalmazások tára. Jelentősen függ a platformon rendelkezésre álló eszköztártól az megvalósítható megoldások száma. (Jednaszewski, 2021)

**Low Code:** Az előre megírt eszköztár, és vizuálisan felépíthető alkalmazásrészek itt is adottak. Szintén nem feltétlen igényel szoftverfejlesztői tudást egyes alkalmazások felépítése ezeken a technológiákon, azonban ahogy a neve is mutatja, lehetőség van egy platform specifikus leíró nyelvben egyedi kódok, scriptek futtatására, ezzel pedig, habár vannak korlátai a Low Code Platformoknak, a No Code-al összevetve nagyobb szabadságot biztosít. (Forsyth, 2021a)

Egyszerű kódok, scriptek írása szintén nem igényel szoftverfejlesztői előismereteket, tudást, de előnyt jelenthet bizonyos komplex alkalmazások fejlesztése során, hiszen itt lehetősége nyílik a fejlesztőnek egyedi, előre nem definiált scriptek korlátozott futtatására. (Forsyth, 2021b)

**Pro Code:** A professzionális, hagyományos megközelítése a szoftverfejlesztésnek. Nevéből fakadóan egyedi kódírást és fejlesztést helyezi előtérbe. Nem tartalmaz előre meghatározott logikai mintákat, mint ahogy a low code és a no code. Leegyszerűsítve a hagyományos programozás egy pro code megközelítése a fejlesztésnek. (Garms)

A low code megközelítés olyan embereket is bevonzott a szoftverfejlesztés világába, akiket a pro code fejlesztések nem tudtak volna bevonzani. Pro code megoldás alatt a hagyományos szoftverfejlesztés értendő. Viszont a Low Code platformok különböző vizuális és „drag and drop” megoldásai szélesebb körben teszik lehetővé a fejlesztést. (Pegasystems, 2021)

A Gartner előrejelzései alapján 2023-ra több nem hivatásos fejlesztő fog alkalmazásokat fejleszteni, mint képzett IT szakember. Az együttműködés, és a visszajelzések feldolgozása is

gyorsabbá és hatékonyabbá válik a Low Code fejlesztés mellett, hiszen nem különül el feltétlen az üzleti oldal és a fejlesztők közötti kapcsolat. Az üzleti oldal ugyanúgy kiveheti a részét a fejlesztésből. (Wong, Iijima, Leow, Jain, Vincent, 2021)

## 2.2 Low Code Development Platformok ismertetése

Fontos megjegyezni, hogy több különböző low code development platform létezik, pont úgy ahogy több programozási nyelv is rendelkezésre áll. Számos különbség van közöttük, mindegyiknek megvan az erőssége és a gyengesége, de ezeket nem fogom mélyebben ismertetni. Szakdolgozatom bevezetésében a low code platformokra hivatkoztam, az egyszerűbb érthetőség kedvéért kiválasztottam egy platformot, amin keresztül mélyebben bemutatom a low code development működését. Választásom a Microsoft PowerApps és PowerAutomate megoldására esett, melynek okát a továbbiakban meg is indoklom. Személyes munka és fejlesztési tapasztalataimon túl figyelembe vettem több iparági elemzést is az elemzett platform kiválasztásakor.

Ezen elemzések közül egyik a Gartner Inc. 2021. augusztusi mágikus négyzet elemzése, ami 4 csoportba sorolja a Low Code Platformokat, 2 tulajdonság mentén.

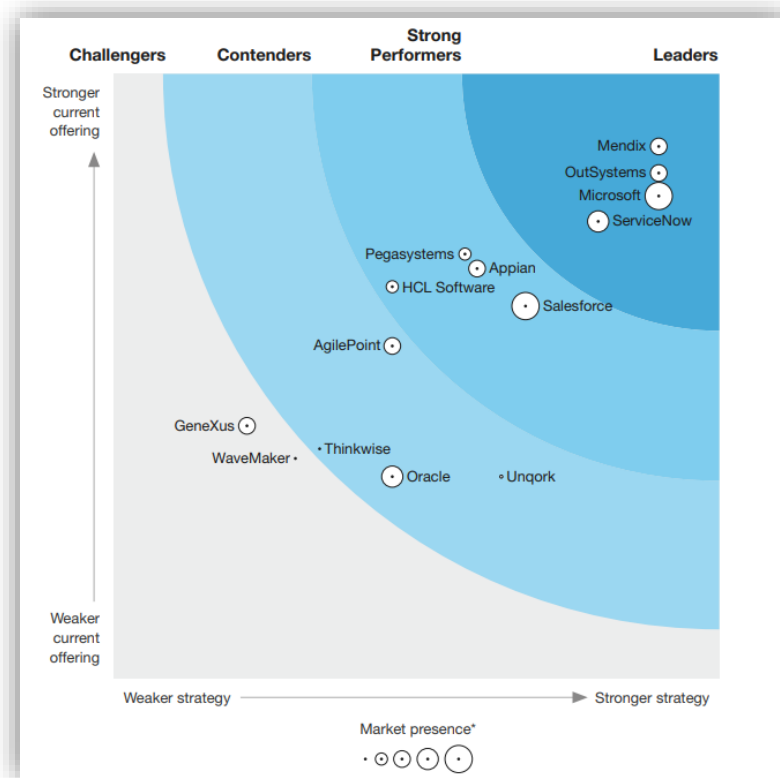


3. ábra, Gartner low code platform elemzése és pozicionálása (forrás: Wong et al., 2021)

Gartner minden évben közzéteszi ezt az elemzést, és az utóbbi években letisztulni látszik a low code platformok világa. Ahogy az ábra is szemlélteti, a Microsoft a vezető platformok között is a felső harmadot képviseli minden tulajdonságában, mind az elképzelés teljessége tekintetében, mind a végrehajtási képességben egységesen vezető pozíciót foglal el évről évre minden Gartner elemzésben. (Wong et al., 2021)

Gartner 2019-es hosszabb elemzésében 2024-re azt prognosztizálta, hogy az alkalmazásfejlesztés több mint 65%-a fog low code platformokon történni, ezzel jelentősen csökkentve a hagyományos pro code programozási technológiák terét. (Vincent, Iijima, Driver, Wong, Natis, 2019)

PowerApps-ról tett részletes elemzését később fogom ismertetni a Gartner jelentésből. Mindezek mellett a Forrester is évről évre pontozza, és minősíti a Low Code platformokat, mely elemzés szintén nem elhanyagolható súllyal bír. A Forrester elemzése hasonló a Gartneréhez. Szintén 4 csoportba sorolja a platformokat, valamint 2 tengely mentén kerülnek a tulajdonságaik pontozásra. A riportban értékelt, és a benne szereplő platformok részben megegyeznek a Gartner riportjának résztvevőivel, valamint a végeredmény is mutat hasonlóságokat, de eltéréseket is. (Bratincevic, Koplowitz, 2021)



4. ábra, Forrester low code platform elemzése és pozicionálása (forrás: Bratincevic, Koplowitz, 2021)

A Forrester riportjában szintén a vezető platformok közé került a Microsoft megoldása, és a vezető platformok között is jó eredménnyel rendelkezik. A Microsoft PowerPlatform megoldását mind stratégiailag, mind jelenlegi lehetőségei mentén kiemelkedő low code platformok közé sorolta a Forrester elemzése. A piaci jelenlétet, piaci kapitalizációt a platformok neve melletti buborék méretén jelzi a Forrester kimutatása. Ez alapján megfigyelhetjük, hogy a Microsoft nem csak technológiai szempontból tölt be vezető szerepet, de a legjelentősebb piaci jelenlét is hozzá köthető. (Bratincevic, Koplowitz, 2021)

Fontosnak tartom röviden megemlíteni, hogy miért foglalhat el vezető szerepet a platformok között a Microsoft megoldása. A Microsoft PowerPlatform több száz külső félhez köthető webszolgáltatás és adatforrás összekötését teszi elérhetővé. A PowerPlatform szervesen illeszkedik a Microsoft ökoszisztémába, így az a szervezet aki Office365 termékeket használ, ami napjaink egyik legelterjedtebb és legnépszerűbb irodai alkalmazáscsomagja, azok számára rendkívül leegyszerűsíti az üzleti alkalmazások fejlesztését, és implementálását. Rendkívül átgondolt és teljes a Microsoft ökoszisztémája a low code fejlesztés szempontjából is. A PowerPlatformmal kapcsolatban a Gartner piackutató cég ezt megerősítve, átlag feletti elégedettséget mért az általános tapasztalat, platform támogatás, és a platform által megvalósítható lehetőségek szempontjában is. (Vincent et al., 2019)

Összességében tehát a fenti részletes riportok, elemzések eredménye, és a saját tapasztalataim miatt esett a választásom a Microsoft PowerApps és PowerAutomate megoldására, mely egy vezető platform, ami komplex megoldást nyújt a legtöbb üzleti igényre. Részletesebben később ismertetem a Microsoft PowerApps megoldását. Ellenben, ahogy látható, vannak rajta kívül szintén vezető low code platformok, mint a Mendix a ServiceNow vagy az Outsystems. Fontos megjegyezni, hogy ezeknek a platformoknak is nagy szerepe van a low code platformok népszerűségének növekedésében és a technológiák fejlődésében, de értekezésemben ezen túl nem foglalkozok velük bővebben, legfeljebb megemlítésre kerülnek majd a továbbiakban. A Microsoft low code platformja a Microsoft PowerPlatform, ami magába foglalja az alábbiakat: PowerApps, PowerAutomate, PowerBi és Power Virtual Agents. Ahhoz, hogy átlássuk a PowerPlatform eszköztárát, ismertetem mindezt röviden a következő fejezetben, mert a Microsoft low code platformja szorosan támaszkodik a PowerPlatform-ot képviselő különböző termékek, megoldások közös erejére, összetettségére. (Microsoft -C)

Továbbá választásom alapját megerősítve, kiemelendőnek tartom azt, hogy több száz nyilvános esettanulmány érhető el több száz különböző cégtől, szervezettől, akik nyíltan vállalják, hogy a Microsoft Platformját aktívan használják a szervezeteiken belül. Többek között olyan nagy

cégek tették le voksukat a Microsoft PowerApps mellett, mint: T-Mobile, Raiffeisen Bank, ExxonMobil, Unilever, Dow Chemical, Toyota Motors, Shell, British Petrol, PayPal, Pepsi, H&M, Coca-Cola, ThyssenKrupp, KPMG, és a sor hosszan folytatható lenne. Valamint megfigyelendő, hogy olyan vezető tech vállalatokon belül is pozitív hatása és fontos jelenléte van a PowerApps platformnak, mint a PayPal vagy a T-Mobile, amely cégek alapvetően a termékpalettájukból fakadóan sok képzett szoftverfejlesztővel rendelkeznek. Viszont nem csak versenypiaci szereplők vezették be a PowerPlatform termékeket a szervezeteik mindennapjaiba, ez állami cégeknél, szervezeteknél is megfigyelhető. (FeaturedCustomers)

### 2.3 A Microsoft PowerPlatform bemutatása

Ahogy fenti bevezetőben már említettem, a Microsoft PowerPlatform napjainkban 4 jelentősebb terméket foglal magába, amik a következők: PowerApps, PowerAutomate, PowerBi, Power Virtual Agents. A Microsoft PowerPlatform komplexitása pontosan a termékek összhangján és működésükben rejlik. Szervesen illeszkednek a Microsoft üzleti ökoszisztémájába, és kiegészítik egymás funkcionalitását, külön is meghatározó termékek a szervezetek digitális transzformációjában, de együtt még több lehetőséget rejtenek magukban. (Moon, 2020)

Szakedolgozatom a Power Platformon belül a PowerApps és a PowerAutomate-re összpontosít, ennek egyszerű oka, hogy a hagyományos értelmében vett alkalmazásfejlesztéshez, és Low Code Developmenthez ennek a két terméknek van lényegében köze. A PowerBi egy adat analitikai platform, míg a Power Virtual Agents egy csevegőrobotok létrehozására alkotott felület. (Microsoft -B)

A szoftverfejlesztéshez hagyományos esetben több professzionális fejlesztő-, és fejlesztést támogató eszköz megvásárlása is szükséges lehet, ha egy szervezet hatékonyan és gyorsan szeretné az elkészítendő projekteket kezelni. A Microsoft PowerApps és PowerAutomate nem igényel egyéb fejlesztőeszközt, hiszen a licenc megvásárlását követően a fejlesztőnek elérhetővé válik a célmegoldások elkészítésére alkalmas fejlesztőkörnyezet, valamint a verziókövetést, felügyeletet támogató funkciók összessége is.

Ahhoz, hogy szervezeten belül elérhetőek, fejleszthetőek, és használhatóak legyenek a PowerPlatform termékek, érvényes licensszel szükséges rendelkeznie a felhasználónak a termékekre. Értekezésemben kizárólag a PowerApps és a PowerAutomate árazását emelem ki, mert ez a kettő Microsoft termék kategorizálható a low code development alá leginkább,

valamint e kettő termék nyújtja a legtöbb lehetőséget és teret a low code fejlesztésre. A következő árak tájékoztató jelleggel kerültek feltüntetésre.

#### **PowerApps (Microsoft-A):**

- 1. lehetőség:** 5\$/felhasználó/alkalmazás/hónap. 1 alkalmazás fejlesztésére és futtatására nyújt lehetőséget. A licenc halmozható. Kisebb cégeknek, kevés alkalmazással lehet megfelelőbb.
- 2. lehetőség:** 20\$/felhasználó/hónap. Korlátlan számú alkalmazás futtatására és fejlesztésére nyújt lehetőséget. Alapvetően megfelelőbb nagyobb szervezeteknek, ahol több alkalmazás fejlesztésére is sor kerül idővel.

#### **PowerAutomate (Microsoft-I):**

- 1. lehetőség:** 15\$/felhasználó/hónap. Korlátlan számú folyamat készíthető.
- 2. lehetőség:** 40\$/felhasználó/hónap. Korlátlan számú folyamat készíthető, kiegészítve mesterséges intelligencia és RPA – robotikus folyamatautomatizálási lehetőségekkel.

Kiemelendő, hogy az Office 365 licencszerződésbe bevonhatóak a PowerPlatform termékek, így ha egy vállalat már rendelkezik Office365 előfizetéssel, akkor jelentősen kedvezőbb árázással találkozhat a valóságban. Természetesen a fent ismertetett árakon és terveken felül a Microsoft bizonyos mennyiségi licencvásárlás esetén kedvezőbb feltételekkel teszi elérhetővé közepesebb-nagyobb szervezetek számára a termékeik licencszerződését, de ehhez egyedi árajánlat kérése szükséges, kiemelendő, hogy az árak országonként, régióként eltérhetnek. A Microsoft gyakran ingyenes kipróbálásra szánt hónapot is kínál a szervezetek, felhasználók számára, hogy eldönthessék megoldást nyújthat-e a problémáikra a PowerPlatform. A fenti árak a Microsoft hivatalos magyar tájékoztató oldaláról kerültek feltüntetésre. (Microsoft -A)

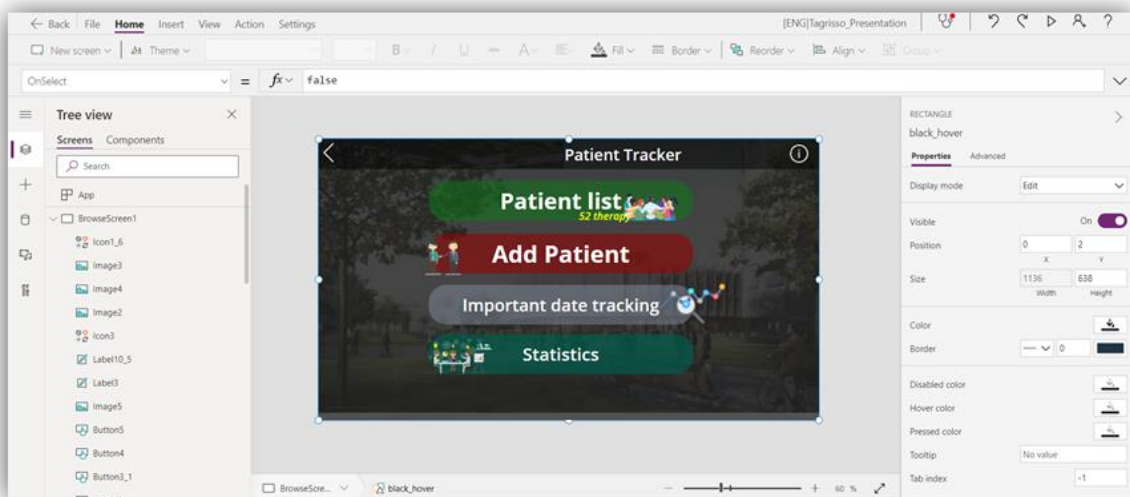
### **2.3.1 A Microsoft PowerApps bemutatása röviden**

A Microsoft PowerApps egy platform, ami lehetővé teszi, hogy a professzionális fejlesztőktől egészen a vállalat bármely dolgozójáig bárki számára elérhetővé, megvalósíthatóvá váljon üzleti alkalmazások fejlesztése számítógépre, telefonra, tabletre és bármely olyan eszközre, ami képes modern böngészőt futtatni, mint a Google Chrome, Mozilla, Safari és a Microsoft Edge. Tehát az alkalmazások tárolása, futtatása, fejlesztése, és életciklusának menedzselése is online történik, ez előnyökkel és hátrányokkal egyaránt jár. Előny, hogy az alkalmazások futtatása hardverfüggetlenné válik, bármely interneteléréssel rendelkező, böngészőt futtatni tudó, és kijelzővel rendelkező eszközre fejleszthető alkalmazás.



Az alkalmazás frissítése központilag menedzselhető, a frissítés feldolgozása, és aktiválása központilag történik, így jelentősen egyszerűsödik az alkalmazások frissítése, és az új funkciók bevezetése. Hátránya ugyanehhez kapcsolódik, hiszen aktív internetelérésre van szükség a PowerApps megoldások futtatásakor, így olyan munkakörnyezetben vagy szituációkban, ahol ez nem megvalósítható állandó jelleggel, nem nyújt teljeskörű megoldást. (Microsoft -A)

Ahogy fentebb megemlítettem, az alkalmazások fejlesztése is online történik, erre egy webes alapon futó studio felület áll a fejlesztési kész munkavállalók rendelkezésére. Ezt a webes stúdiót a Microsoft rendszeresen fejleszti, és implementál új funkciókat, lehetőségeket. Mivel a PowerApps a LCDP (low code development platform) elvet követi, így a fejlesztés jelentős része vizuális módon történik, kevés kódolással, azonban lehetőség van kódrészeket futtatni és írni egyes részekhez. Lényegében ez különbözteti meg a Low Code Platformokat a No Code Platformoktól. (Moon, 2020)



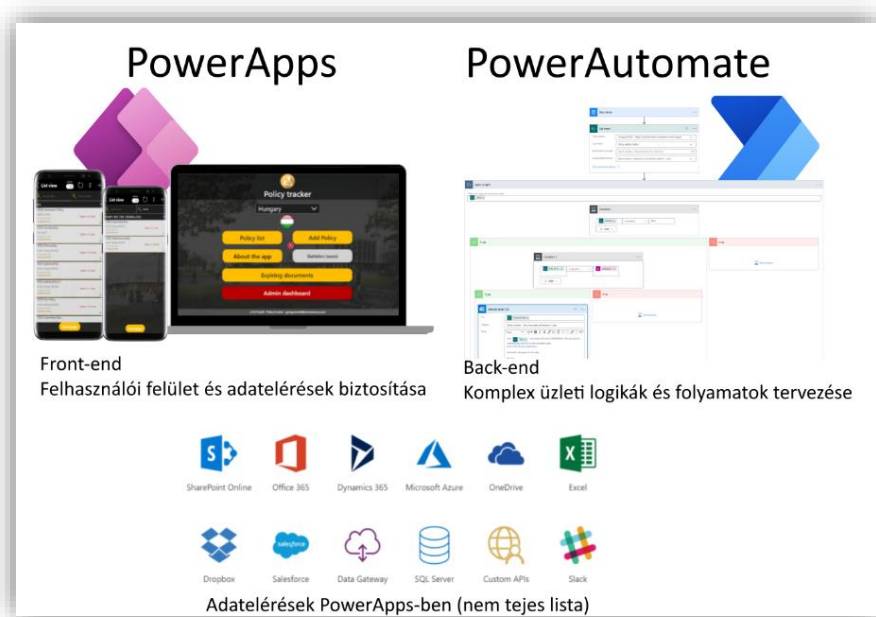
5. ábra, A PowerApps fejlesztői környezetéről és egy CRM alkalmazásomról (saját szerkesztés)

Az adatok tárolása, és elérése felhőmegoldások révén történik, mint például: Onedrive, Sharepoint, Dynamics 365, SQL adatbázisok, Microsoft Dataverse (korábban Common Data Service) által, és több harmadik féltől származó felhőmegoldást támogat a PowerApps, ha nem a Microsoft ökoszisztémájában képzelel el egy cég az adattárolást. (Microsoft -A)

Microsoft Dataverse kézenfekvő és átlátható adattárolási és kezelési lehetőséget nyújt a PowerApps fejlesztőknek arra, hogy a szükséges adatokat skálázhatóan, biztonságosan, dinamikusan és egyszerűen átláthatóan tárolja, és kezelje. Dataverse megkönnyíti az adatok kezelésével, adatbázisok beállításával, kapcsolatok meghatározásával, és megtervezésével járó

nehézségeket, és egy összetett megoldást nyújt a szervezeti igényekre, amely a PowerApps további funkcióival kihasználva egészen komplex üzleti alkalmazások fejlesztését teszi lehetővé. (Microsoft -E)

Lényegében tehát a PowerApps felel az alkalmazás vizuális megjelenésének meghatározó részéért, a felhasználói felület megjelenéséért, miközben kezeli az adateléréseket, adatkapcsolatokat, felhasználói bemeneteket, megjeleníti bizonyos folyamatok, back-end logikák végeredményét, és képes folyamatokat kezelni önmagában is, de erről részletesen később. PowerApps alkalmazásokban hatalmas potenciál rejlik, hiszen egészen komplex back-end logikákat lehet benne megjeleníteni, kezelni, viszont itt jön a képbe a PowerAutomate, ami hatalmas eszköztárral rendelkezik ahhoz, hogy az üzleti életben felmerülő valamennyi automatizálható, digitalizálható folyamatra megoldást nyújtson, kiegészítve a PowerApps eszköztárát és lehetőségeit. (Moon, 2020)



6. ábra, PowerApps és PowerAutomate integritásáról (saját szerkesztés, forrás:Microsoft-C)

Ábrámon a fent leírtakat szemléltetem vizuálisan, hogy miként kapcsolódik össze egy PowerApps alkalmazás és egy PowerAutomate folyamat. A felhasznált példát szakdolgozatom későbbi részében részletesen ismertetem, mely egyike az Európa szerte használt alkalmazásaimnak. A fenti példán is látható, hogy ameddig a PowerApps egy felület, ami képes a felhasználói interakciókat kezelni, adateléréseket megjeleníteni, adatokat szerkeszteni, menteni, frissíteni, felhasználói felület felhasználóbarát megjelenítését nyújtani, PowerAutomate integrációt és platformfüggetlen elérhetőségét biztosítani, addig komplex

folyamatokat a PowerAutomateben lehet összerakni, kezelni, és haladásukat monitorozni, figyelemmel követni. (Moon, 2020)

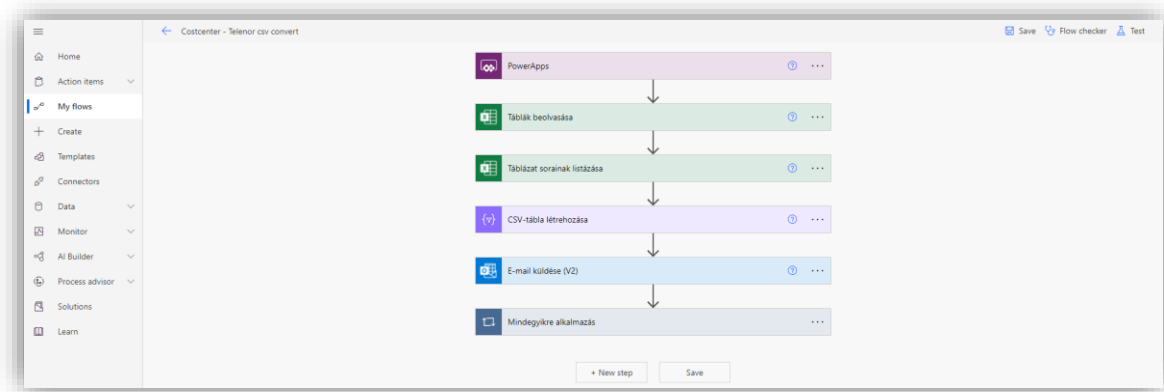
Fontosnak tartom kiemelni, hogy attól függetlenül, hogy a low code development szélesebb vállalati körben az utóbbi években kezdett elterjedni, napjainkban számos nagyvállalat mindennapi fejlesztésének, és folyamatainak szerves részét képezik a PowerApps és PowerAutomate megoldások. A Toyota Motors, ami világszerte több mint 360 000 embert foglalkoztat, és számos gyárral, irodaépülettel rendelkezik globálisan, bevezette a PowerApps-t, ami lehetőséget biztosított számára, hogy további fejlesztők felvétele nélkül a már meglévő munkavállalói közül megtalálhassa azokat a kreatív személyeket, akik képesek egyedül alkotni egy LCDP segítségével. Ennek eredménye, hogy 2020-ra több mint 400 egyedi PowerApps megoldás készült a Toyotán belül, amik egészen a pár főtől a pár ezer felhasználóig terjedő megoldást képviselnek. Chris Ingalls, a Business és Solution Architect szerint a Toyota számára – ami mindig is egy lean vállalatként üzemelt – jelentős akadályt okozott, hogy egyes üzleti egységek nem tudtak megoldást készíteni a saját problémájukra, ez összességében rontotta a vállalat agilitását. Erre való válaszként vezették be a PowerApps-t, amivel korai adoptálóként hamarabb fedezhették fel annak a jelentőségét, hogy a kreatív munkavállalók, és számos üzleti egység hatékony megoldást tud készíteni a saját problémájára, miközben nem kell aggódnuk olyan problémák miatt, mint a hálózati beállítások, szoftver architektúra, karbantartás, üzemeltetés, mert ezek mindegyikére egyszerűsített megoldást nyújt a bevezetett platform, szemben a hagyományos Pro Code által fejlesztett megoldásokkal. A Toyota különösen elkötelezett abban, hogy ezek az egyedi megoldások sok feleslegesen felhasznált papírtól vagy extra költségtől szabadították meg egyes üzleti egységeket, miközben a saját munkavállalóinak képességtára nőtt, ezzel egyidőben viszont a monoton, egyhangú munkafolyamatok száma jelentősen csökkent. Ez rendkívül fontos lépés, hogy egy vállalat a digitális transzformációt sikeresen teljesíthesse. (Toyota, 2020)

### 2.3.2 A Microsoft PowerAutomate bemutatása röviden

A Microsoft PowerAutomate egy folyamat automatizációs eszköz, ami önmagában is működőképes, de szervesen illeszkedik a Microsoft PowerPlatform ökoszisztémájába, és minden PowerPlatform megoldásba integrálható. (Canorea, 2021)

Rendelkezik Desktop alkalmazással, amivel lokálisan tudunk folyamatokat automatizálni a saját eszközünkön, azonban számunkra jelenlegi alkalmazásfejlesztési témában a webes studio

felülete mérvadó, ahol egy egyszerű, és átlátható felületen előre definiált kapcsolatokkal lehet vizuálisan felépíteni automatizált folyamatokat. (Microsoft -I)



7. ábra, A PowerAutomate felületéről, egy PowerApps-ba integrált PowerAutomate folyamatomról (saját szerkesztés)

Ahogy a 7-es számú ábrán is látható egy PowerAppsba integrált folyamat képe, egyszerű, és átlátható felülettel rendelkezik a PowerAutomate, egyértelmű ikonokkal és feliratokkal, ami nem ijeszti el az átlag felhasználót sem elsőre.

Az olaszországi székhellyel rendelkező illimnity elnevezésű digitális banknak, egy hosszú, sokáig email, és telefonhívás alapú hitelezési folyamat elbírálását sikerült automatizálnia a PowerAutomate segítségével. Az elbírálásokkal foglalkozó csapat igényenkénti 1 órás időszükségletét sikerült igényenként 20 percre csökkenteni, és átlátható digitális formába költöztetni. Az illimnity IT Department munkáit és leterheltségét jelentősen tudták csökkenteni a megfelelő digitalizált folyamatokkal és alkalmazásokkal. Francesco Zitelli az illimnity IT Managere véleménye szerint a Power Platformot minden egyetem első évében szükséges lenne oktatni. Az után jutott egy vezető erre a következtetésre, hogy megtapasztalta milyen gyorsan és egyszerűen tudnak fontos és értékes üzleti célmegoldásokat készíteni, amiknek jelentős hatása van a vállalatra és a folyamatok hatékonyságára. Chiara Sibia, szintén az illimnity Digital & ICT Operations Specialistája nem kevesebbet állít, mint, hogy a Power Automate és a PowerApps összessége teljesen megreformálta a munka végezésének a módját. (illimnity, 2020)

A Coca Cola szintén egy nagyon hosszú és sok kézi interakciót igénylő folyamatát tudta sikeresen automatizálni, és jelentősen csökkenteni a szükséges emberi beavatkozást. A régi folyamatban egy CRM ügynök fogadta a megrendelést, azt beírta egy excel állományba, SAP rendszerben megírta a hozzá tartozó PO-t, kibővítette az excelt az SAP alapján, kinyomtatta a

dokumentumokat, elküldte emailben a számlákat, és egy másik rendszerben szintén rögzítette a megrendelést. A kinyomtatott dokumentumokat aláírta és jóváhagyta. (Coca-Cola, 2020)

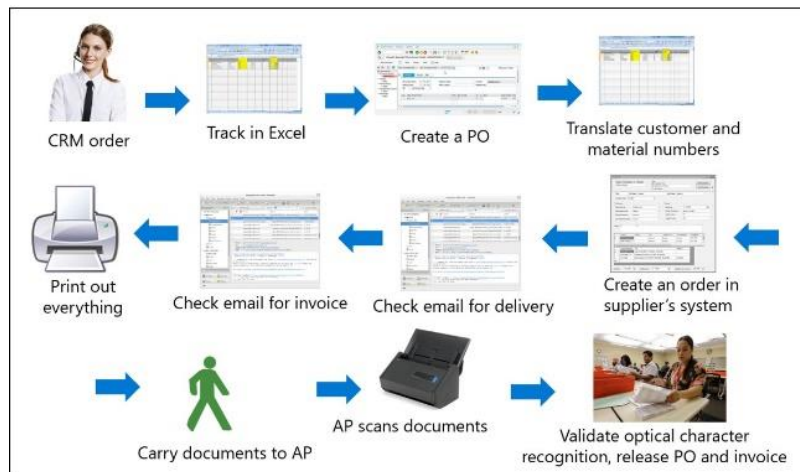


Figure 1. Manual Freestyle cartridge replacement before RPA. For larger versions of all the figures in this story, go to the Downloads section of the left-hand sidebar.

8. ábra, A Coca-Cola régi ügyfélkezelési folyamatáról (forrás: Coca-Cola, 2020)

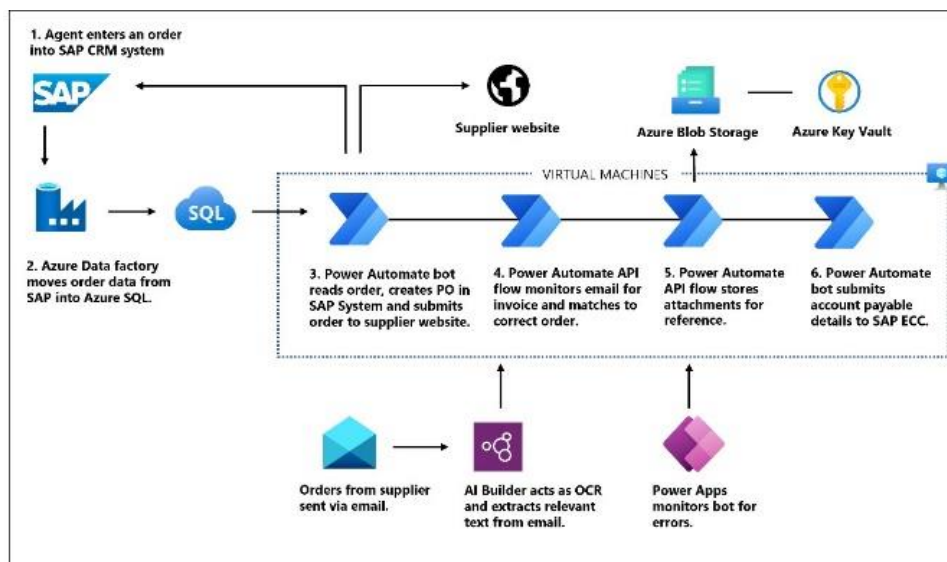


Figure 2. Bots built using Power Automate automate an order and invoice process with suppliers.

9. ábra, A Coca-Cola új, automatizált ügyfélkezelési folyamatáról (forrás: Coca-Cola, 2020)

Ez egy adminisztrációs rémálom, és jelentősen bekorlátozza az üzlet skálázhatóságát, egy ügynök maximum 50 ilyen folyamatot tudott egy nap elvégezni a Coca Cola szerint. A 8. ábrán a régi manuális folyamat, míg a 9. ábrán az új PowerAutomate vezérelte folyamat látható. A CRM ügynöknek kizárólag az SAP rendszerében kell rögzítenie a megrendelést, innentől minden automatizáltan történik, kezdve az adatok tárolásától, értesítésen át a bejövő csatolmányok feldolgozásáig. Ezzel a CRM ügynökök egyhangú adminisztráció alapú munkáját jelentősen csökkentette, és skálázhatóvá tette a megrendelések kezelését a Coca Cola. A felszabadult munkaerőt pedig egyéb kreatív projektekre tudtaallokálni a cég. Mindezek

mellett nem elhanyagolható a működési költség csökkenése, amit a nyomtatások kihagyásával elért a cég, nem utolsósorban pedig egy zöldebb megoldás. Az emberi hibatényező szintén jelentősen csökkent az automatizált folyamat által. (Coca-Cola, 2020)

### 2.3.3 A Microsoft PowerApps alkalmazások típusai

A PowerApps-en belül is jelenleg három féle alkalmazástípus fejlesztése között választhatunk, ezek egymástól eltérő alappillérekkel rendelkeznek, amit fontosnak tartok ismertetni a következő fejezet előtt. Fontos megjegyezni, hogy a három különböző alkalmazástípus különböző online stúdióval, szerkesztővel rendelkezik, így eltérő a három típusnak az eszköztára és a felhasználásra szánt célja is. (Microsoft -A)

**PowerApps Canvas apps:** A Canvas app fejlesztése támogatja leginkább a fejlesztő vizuális képességeinek kihasználását. Cégen belüli alkalmazások építésére ajánlott, és nagy teret ad a fejlesztőnek az alkalmazás interfészének személyre szabásában. A fejlesztés közben szem előtt tartandó cél a felület használhatósága kell, hogy legyen. Egyszerre több adatkapcsolatot, API-t tud kezelni, ha összetett back-end logikát szükséges bele építeni, azt a PowerAutomate-el együtt lehetséges benne kezelni. Legnagyobb UI eszköztárral rendelkezik a három típus közül, ebben készíthetők a legegyszerűbb megoldások. Az elkészült alkalmazás mindazonon túl, hogy böngészőben és hordozható eszközökön is futtatható, integrálható egyéb alkalmazásokba, mint például a Microsoft Teams-be és PowerBi-ba. Komplex verziókezelő, és monitorozó megoldással rendelkezik a stúdió része. (Microsoft -G)

**PowerApps Model-driven apps:** Ahogy a neve is mutatja, ezek modell-adatvezérelt alkalmazások, így a fókusz nem a felhasználói felületen és az UI eszköztár komplexitásán alapszik, hanem az adatokon és azok feldolgozásán. A modell vezérelt alkalmazások szintén belső céges felhasználásra megfelelőek. Olyan projektekhez nyújt megfelelő megoldást, ahol sokkal lényegesebb az adatokkal, adatbázistáblákkal való műveletek elvégzése és azok kapcsolata. Összetett adatokon alapuló üzleti folyamatok kezelésére alkalmas önmagában is, viszont nem lehetséges benne olyan egyedi front-end felhasználói felületet létrehozni, mint a canvas app típusnál, a felület alapvetően elég korlátozottan szerkeszthető, inkább összpontosít arra, hol milyen kimutatást, űrlapot vagy táblázatot szükséges megjeleníteni, végső simításokat önmaga által elvégzi az UI-val kapcsolatban. Nem integrálható más alkalmazásokba a canvas alkalmazásokkal ellentétben, valamint nem rendelkezik verziókövető lehetőséggel sem. (Microsoft -F)

**PowerApps Portal:** A legfontosabb különbség a Portal és az előző kettő megoldás között, hogy ameddig a modell vezérelt (eredeti megnevezés szerint model-driven) és a canvas típusok belső céges felhasználásra nyújtanak megoldást, addig a Portal alapvetően cégen kívüli felhasználásra nyújt fejleszthető megoldásokat. Lényegében modern rezponzív weboldalak elkészítésére alkalmas, amelyben szintén megjeleníthetünk kimutatásokat, riportokat aszerint, hogy mivel kötjük össze a weboldalunkat. Külső felhasználók azonosítására van lehetőség, így meghatározható a weboldalt elérő felhasználók köre is, LinkedIn, Google, Facebook azonosítási lehetőség is elérhető. PowerBi, Microsoft Dataverse és PowerAutomate összekapcsolás lehetséges, így a külső partnerekkel való együttműködés során is összetett megoldásokat készíthetünk. Több funkcionalitást kínál, mint egy bemutatkozó weboldal, és több lehetőség is rejlik benne. Studiója, szerkesztője inkább hasonlít egy leegyszerűsített HTML, CSS szerkesztőre, és az UI eszköztára is jelentősen korlátozottabb, mint a canvas app esetében. (Microsoft -D)

## PowerApps alkalmazások típusai

### röviden

 <p><b>Canvas</b></p> <ul style="list-style-type: none"> <li>-Több adatforrás kezelése</li> <li>-UI felett teljes kontrol</li> <li>-Fókuszban a felület használhatósága</li> <li>-Belső felhasználásra</li> </ul>	 <p><b>Model Driven</b></p> <ul style="list-style-type: none"> <li>-Fókuszban az adatok, adatbázis</li> <li>-Limitált UI szerkeszthetőség</li> <li>-Belső felhasználásra</li> <li>-Microsoft Dataverse fókuszban</li> </ul>	 <p><b>Portal</b></p> <ul style="list-style-type: none"> <li>-Külső felhasználásra</li> <li>-Limitált UI eszköztár</li> <li>-Weboldalszerű megoldás</li> </ul>
--	--	--

10. ábra, A PowerApps alkalmazástípusokról (saját szerkesztés, forrás: Microsoft -D)

A következőkben egy valós, megtörtént példát szeretnék bemutatni. Kobe városa, ami egy jelentős nagyváros Japánban közel 1.5 millió lakossal, olyan problémával találkozott a COVID-19 világjárvány alatt, hogy gyors, és hatékony megoldásra volt szükségük. A Japán kormány vészhelyzeti állami támogatás felvételét hagyta jóvá az állampolgárok számára, azonban az elbírálásokról való tájékoztatást a városoknak kellett megoldaniuk. Kobe városa volt, hogy egy nap 40,000 hívást fogadott ebben a témában, ami nehéz helyzetbe sodorta a város vezetését és beosztottjaikat. Összetett PowerPlatform megoldást hozott létre a város IT csoportja, aminek



jelentős részét képezte a Portal megoldásuk, ami lehetővé tette a város lakóinak számára, hogy egyszerűen lekérdezhessék a weboldalon a jelentkezésük állapotát. A Portal megoldás tartalmaz egy alkalmazás mögötti model-driven PowerApps megoldást, PowerAutomate folyamatokat, Power Virtual Agents chatbotokat és PowerBi dashboardokat, minden résznek megvan a maga szerepe a folyamatban, ami egy válsághelyzet alatt összetett és mégis gyors megoldást nyújthatott a városnak és a benne élő lakosoknak. (Yoshida, 2020)

Fenti példa jól szemlélteti a Microsoft Low Code Platform ökoszisztémának összetettségét, hogy a különböző PowerPlatform elemek miként integrálhatóak egymásba, hogy azok a végén egy összetett megoldást alkossanak együtt, mindezt egyszerűen, átláthatóan és gyorsan, akár egy globális krízishelyzet közepén. A PowerPlatform ismertetőjében is említettem, hogy a Microsoft LCDP hatékonysága a különböző PowerPlatform elemek együttműködésén alapszik. (Microsoft -C)

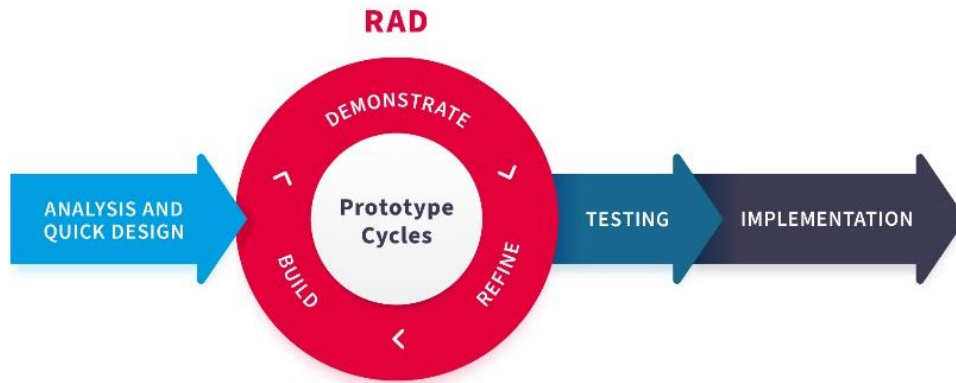
### **3. Szoftverfejlesztés a gyakorlatban**

#### **3.1 Rapid Application Development**

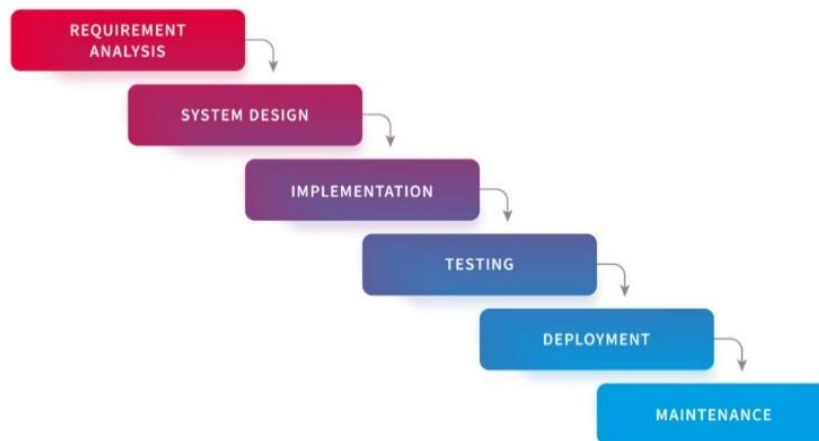
A Rapid Application Development, magyarul gyors alkalmazásfejlesztés alapelvét James Martin dolgozta ki az 1980-as években. A RAD, mint szoftverfejlesztési elv, a vízésésmodell módszertanra érkezett válaszul, ami alapvetően az építőiparból származott, és nem minden esetben nyújtott megfelelő struktúrát azon szoftverfejlesztési projekteknél, ahol az ügyfél nem tudta pontosan előre meghatározni az elvárásait és a céljait. Ezekben az esetekben a lineáris, visszacsatolással nem rendelkező vízésésmodell használatakor az elejétől kellett kezdeni egész folyamatot. Egy előre nem szigorúan meghatározott projektstruktúrában ezért a vízésésmodell, mint fejlesztési módszertan nem volt megfelelő. A RAD módszertan alapja a gyors prototípuskészítés, felhasználói visszajelzések begyűjtése, visszajelzések alapján az új fejlesztések, javítások implementálása, majd pedig ismét visszajelzések kérése a felhasználótól. A folyamat tehát egy gyors, de lehetőségekhez mérten minél alaposabb projektcél értelmezéssel kezdődik, ezt követi egy úgynevezett minimum viable product – MVP, magyarul minimálisan működőképes termék, prototípus létrehozása. Harmadik lépésben a fejlesztők alapszintű visszajelzéseket kérnek a felhasználótól, ügyféltől az MVP kapcsán. Ez leginkább használhatóságra, használati érzésre terjed ki. Negyedik lépésben az ügyfélnek prezentálásra kerül a minimum viable product. Az MVP-vel kapcsolatos visszajelzések alapján



további fejlesztés, hibajavítás lehet indokolt, majd pedig megkezdődik egy szélesebb szintű tesztelése a terméknek, hogy biztosan minden meghatározott projektcélt, és kliensigényt kielégít-e a termék. Végül soron a RAD, mint módszertan gyorsabb fejlesztést, igényeket jobban kielégítő és jobb, kevesebb hibával rendelkező végtermék, szoftver létrehozására nyújt lehetőséget. (Martin, 1991)



11. ábra, A RAD módszertan (forrás: Nix)



12. ábra, A vízésés módszertan (forrás: Nix)

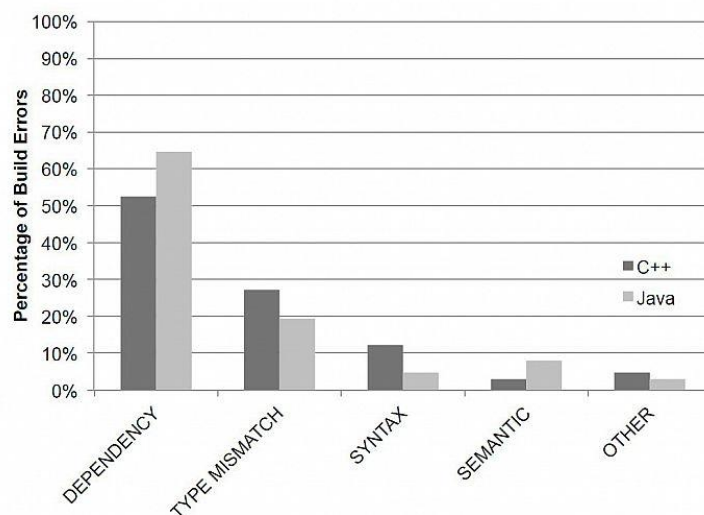
A RAD, mint módszertan élesen megmutatkozik a low code development keretei között, hiszen e technológiára fejlesztett alkalmazások, célmegoldások háttérében megbújó okok, projektek, és folyamatok sok esetben azért low code megoldással kerülnek kifejlesztésre, mert gyors, hatékony, és költségbarát megoldásra van szükség, mely projektekben a siker, és a hatékonyság sokszor úgy mérhető, hogy milyen gyorsan tudott reagálni az adott vállalat az adott problémára. Így tehát a low code development alatt megvalósul az 1980-as években lefektetett gyors alkalmazásfejlesztés alapelvei, és céljai. (Nix)

## 3.2 Általános nehézségek szoftverfejlesztés közben

A szoftverfejlesztés egy összetett szakma, ami naprakész felkészültséget, és tapasztalatot igényel, hogy megfelelő megoldásokat tudjon egy programozó fejleszteni céges környezetben. Szeretném ismertetni, nagy általánosságban milyen szoftverfejlesztésbeli nehézségekkel találkozhatnak fejlesztők és tanulók napról napra.

Nagyon gyorsan fejlődik napjainkban az eszközök hardvere, mind fejlettségben, sebességben, mind pedig architektúrában, vagy összetettségben. A hardveres fejlődéssel párhuzamosan folyamatosan jelennek meg új programozási nyelvek, új ökoszisztémák, technológiák, vagy pedig egy új verziója a már létező programozási nyelvnek, keretrendszernek, könyvtáraknak. Új programozási paradigmák jelennek meg, és régiéik tűnnek el egyik pillanatról a másikra. (Lowery, 2018)

Fentiekből tehát megállapíthatjuk, hogy egy jó szoftverfejlesztőnek naprakésznek kell lennie a szakterületén, ha évről évre ugyanazt a minőségű, vagy jobb munkát szeretné kiadni a kezei közül, ez a szakmát érintő folyamatos újratanulás, és fejlődés bizonyos esetekben kifejezetten nehezzé teszi a szakmában való kiteljesedést, helytállást. (Lowery, 2018)

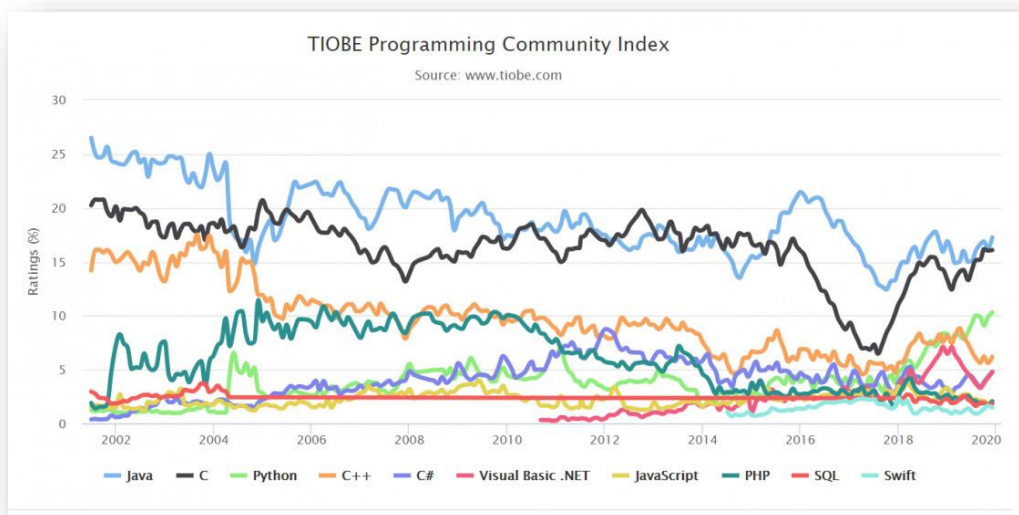


13. ábra, Google kutatása szerint elkövetett leggyakoribb hibák programozás közben (forrás: Seo, Sadowski, Elbaum, Aftandilian, Bowdidge, 2014)

A Google kutatást végzett a Java és C++ fejlesztők körében, ahol 56 millió adatot feldolgozva próbáltak képet alkotni arról, hogy milyen gyakori technikai hibákat ejtenek a fejlesztők a munkájuk során. (Seo, et al., 2014)

Meglepő eredményként a vezető hiba a függőségekhez, és azok kezeléséhez volt kapcsolható, tehát a fejlesztő olyan típusra, állományra, változóra hivatkozott, aminek az elérése nem volt biztosítva az adott forrásrészletben. Második helyen a típuseltérési hibák jelentek meg mindkettő programozási nyelvnél, tehát a deklarált változóhoz inkompatibilis műveletet társítottak. (Seo, et al., 2014)

A programozási nyelvek is folyamatosan változnak, ahogy azt már említettem. Viszont ami még kiemelendőbb, hogy a programozási nyelvek népszerűsége, használtsága, gazdasági jelzővel illetve „piaci kapitalizációja” is egy folyamatosan változó tényező, amit nem hanyagolhat el egy fejlesztő, ugyanis ha valaki leragad egy programozási nyelvnél, ami mellett elmegy a piac, és egyre kevesebb helyen használják, úgy ezzel együtt a fejlesztő lehetőségei is beszűkülnek a munkaerőpiacon. Gondoljunk csak arra, mi történhetett azzal a fejlesztővel, aki a pascal programozási nyelv után nem képezte tovább magát, hanem megmaradt annál. Természetesen minden programozási nyelv nehézsége más, ahogy a logikájuk is eltérő, így mindenki megválaszthatja a maga logikájához leginkább illeszkedő lehetőséget. Egyik programozási nyelv úgynevezett tanulási görbéje sem elhanyagolható, legyen akármilyen képzett és gyakorlott egy fejlesztő, ha új nyelv elsajátítása mellett dönt. Így a befektetett idő, mint tényező szintén jelentős. (Lowery, 2018)



14. ábra, Programozási nyelvek népszerűségéről (forrás: Tiobe, 2021)

A fenti ábrán láthatjuk hogyan alakult a vezető programozási nyelvek népszerűségének változása 2002 és 2020 között, szemmel látható, hogy a piac folyamatosan változik, a Swift például 2014-ben még nem is létezett, mint programozási nyelv, ahogy az is megfigyelhető,

hogy a 2005-ös években kifejezetten népszerű PHP és C++ 2020-ra szinte 10%-ot esett, amivel jelentősen csökkent a népszerűségük és a piaci jelenlétük. (Tiobe, 2021)

Ahhoz, hogy megérthessük mi vezetett, és mi vezet napjainkban a low code development és a platformjai népszerűségének folyamatos növekedéséhez szeretném röviden szemléltetni, hogy milyen általános nehézségekkel néz szembe egy szervezet, ha egy fejlesztői csoportot szeretne fenntartani különböző szoftverigényhez kapcsolódó célból. Miután a fentiekben felületesen, de megemlítésre került pár fejlesztői oldalról tapasztalható nehézség is.

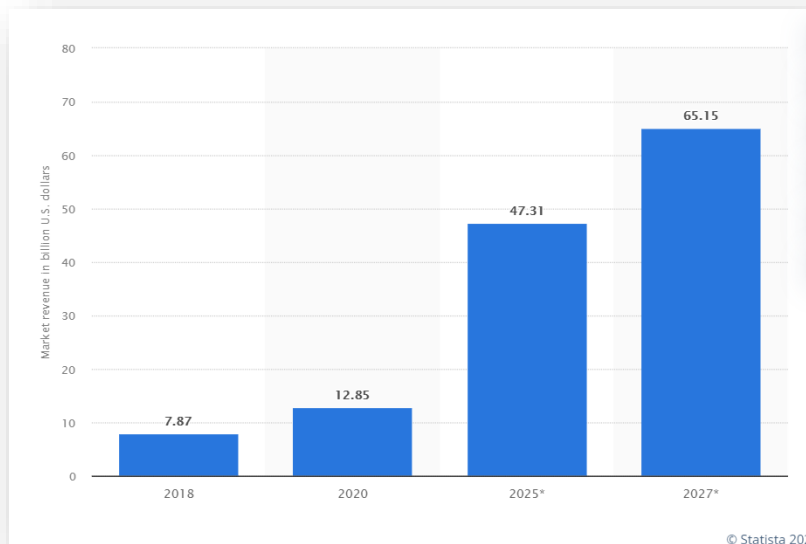
Szervezeti oldalról egyik kiemelt nehézség a megfelelően képzett, és tudással rendelkező szakember megtalálása és felvétele, későbbiekben pedig egy ilyen kiélezett szakemberhiány alakította helyzetben a felvett munkaerő megtartása. (Breux, Moritz, 2021)

Természetesen egy szoftverfejlesztő csapat fenntartása nem elhanyagolható összeg akkor sem, ha a későbbiekben fejlesztett megoldások mindezek mellett pénzt vagy időt spórolnak a szervezetnek. Mindenekelőtt a szervezetnek szükséges forrást elkülönítenie egy megfelelő csapat felépítésére, vagy pedig egy megbízható külső vállalkozással való szerződésre. Ugyanakkor felmerülő akadály, hogy a meglévő erőforrást minél hatékonyabban tudja kihasználni a szervezet, ez hozzáköthető a szakemberhiányhoz szintén, hiszen sok esetben a meglévő erőforrással kell egyre több munkát ellátnia a fejlesztőknek, ami kezeletlen körülmények között előbb-utóbb más problémákat eredményezhet, mint például az elkészített munkák minőségének leromlása, dolgozói elégedetlenség vagy be nem tartott határidők. (Habók, 2018)

A fentebb ismertetett általános szoftverfejlesztésbeli nehézségek, és felmerülő hibák jelentős része nem jelenik meg a low code platformokon való fejlesztés során. A low code platformok elsajátításának, betanulásának ideje sokkal rövidebb és egyszerűbb, így könnyebben sajátítható el több platform egyszerre vagy külön-külön. Ezzel pedig jelentősen csökken a lehetősége annak, hogy egy fejlesztő egy folyamatosan csökkenő népszerűségű, és kevésbé használt technológiában ragad. A low code platformok vizuális fejlesztőkörnyezetei pedig, a fejezet első felében ismertetett programozásban gyakran előforduló hibák jelentősen alacsonyabb előfordulására nyújtanak megoldást, hiszen a fejlesztés alatt a fejlesztőnek nem szükséges az alkalmazás legmélyebb részeit is megírnia, megteszi azt helyette a platform.

### 3.3 Milyen megoldást nyújt a nehézségekre a Low Code

Többféle piaci előrejelzés elérhető különböző piackutató szervezetektől, de abban mindegyik egyetért, hogy az alkalmazásfejlesztésre a következő években egyre inkább megnő a kereslet a szükséges digitális átalakulás miatt.



15. ábra, Low code platformok piacának növekedéséről (forrás: Mlitz, 2021)

A Statista piackutatással foglalkozó vállalat előrejelzése szerint 2027-re akár 5-6x nagyobb lesz a Low Code Development Platformok piaci bevétele, ami párhuzamba állítható elterjedésüknek növekedésével is. (Mlitz, 2021)

A szoftvermérnökök, képzett programozók iránti kereslet már napjainkban is jelentősen meghaladja a rendelkezésre álló munkaerő számát, így az első fejezetben ismertetett várható drasztikus digitális fejlődés, és transzformáció hatására megkerülhetetlen lesz a low code development azon szervezetek számára, akik szeretnének lépést tartani a folyamatosan fejlődő világunkkal. Képzett munkaerő híján pedig érthető, miért jósolnak drasztikus növekedést valamennyi low code development platform számára, bizonyos előrejelzések szerint pedig évről évre egyre általánosabbá fog válni a low code development gyakorlati ismerete. Nem véletlen az sem, hogy a legtöbb 10-es listára szűkített előrejelzés, ami a szoftverfejlesztés jövőjéről szól, tartalmazza a low code/no code development várható drasztikus térnyerését. (Martin, 2021)

A low code platformok folyamatosan növekvő térnyerése, és jelenléte tehát szinte egy konszenzussal elfogadott tény a piackutató szervezetek jelentéseiben. (Wong, et al. 2021)

A 14., programozási nyelvek népszerűségét bemutató ábrán látható, hogy mind a két, mai napig legnépszerűbb nyelv a Java és a C++, előbbi első kiadása az 1990-es évek elején, míg az utóbbi az 1980-as években jelent meg. (Ferguson, 2004)

Persze a programozási nyelvek azóta rengeteget változtak, és fejlődtek az idővel, de ahogy a piaci igény újabb, fejlettebb, és egyszerűbb programozási nyelveket hívott életre, úgy ugyaninnen eredeztethető a low code platformok megjelenése is, hiszen valamennyi low code development platform mögött is egy programozási nyelv húzódik meg, mint például, ahogy a Microsoft PowerApps mögött a Microsoft újonnan kifejlesztett nyílt forráskódú programozási nyelve felelős a folyamatokért, a PowerFx névre keresztelt programozási nyelv. (Microsoft -H)

## **4. Esettanulmányok**

### **4.1 Esettanulmányok ismertetése röviden**

Ahogy a bevezetésben említettem, 2 esettanulmány bemutatása a szakdolgozatom részét képezi. A később ismertetett alkalmazásokat, és folyamatokat egy multinacionális gyógyszeripari vállalatnál töltött munkatapasztalat keretei között készítettem és fejlesztettem. A bemutatott megoldások a Microsoft PowerPlatform-on belül a Microsoft PowerApps-en, és a PowerAutomate-en alapszanak. A későbbiekben szereplő képek, ábrák mindegyike a működő célmegoldások tesztelésre használt verziójából származnak, amik tesztelésre szánt nem valós adatokkal vannak feltöltve, ellátva. Fontos kiemelni, hogy mindennek ellenére logikailag, és technikailag nem különbözik a teszt alkalmazások felépítése a való életben is használt verzióktól. A személyes adatok védelme miatt láttam indokoltnak, hogy a szakdolgozatom bemutatása közben használt képi anyagok ne tartalmazzák harmadik félnek a valós adatait. Amennyiben egy-egy ábrában kitakart vagy átírt adatok láthatóak, azok szintén adatvédelmi szempontból kerültek kitakarása vagy átírásra. Kiemelendőnek gondolom, hogy fejlesztői munkatapasztalatom során több Microsoft PowerPlatform-ra épülő megoldást készítettem, például: célra szabott CRM alkalmazásokat, IT helpdesk alkalmazást, IT eszközzelők alkalmazást, eladási adatokat feldolgozó, és kimutatásokat készítő alkalmazást, szabályozáshoz kapcsolódó dokumentumok nyilvántartását, és követését segítő alkalmazást – policy tracker, pénzügyi adminisztrációs alkalmazást, adatvédelmi nyilatkozat digitalizált-automata folyamatot, stb.

Formai irányelvek miatt így az esettanulmányaimat, és megoldásaim bemutatását érintően 2 megoldás ismertetése mellett döntöttem melyek az alábbiak:

GDPR nyilatkozatokat kezelő automatizált PowerAutomate folyamat, illetve IT eszköznilyvántartó PowerApps alkalmazás PowerAutomate folyamatokkal. Az esettanulmányban bemutatott megoldások 100%-ban a saját fejlesztéseim, nem tartalmazzák külsős, harmadik fél megoldásait.

## 4.2 Esettanulmány: GDPR nyilatkozatok kezelése - PowerAutomate

A címben szereplő GDPR nyilatkozatok kezelése a régi hagyományos formában történt a szervezeten belül. Későbbiekben részletesen kifejtem, miért volt szükség a folyamat automatizálására, ezt miként lehetett megvalósítani, valamint milyen tanulságai, hozadékai, pozitívuma van az új automatizált GDPR folyamatkezelésnek a hagyományossal szemben.

### Régi, manuális folyamat



16. ábra, GDPR nyilatkozatok régi, hagyományos folyamatáról (saját szerkesztés)

A 16. ábrán a régi folyamat egyszerűsített leírását ábrázoltam. A folyamat az következők szerint zajlott. A területi munkavállaló kitöltte a GDPR dokumentumot a szervezet ügyfelével, ezt követően a kitöltött papír alapú dokumentumot valamilyen formában el kellett juttatnia eredeti példányban a vállalat budapesti irodájába, azon belül pedig az ezt a folyamatot

kezelő marketing menedzser asztalára. A marketing menedzser ellenőrizte a beküldött nyilatkozatot.

Helytelen kitöltés esetén manuálisan, emailben értesítette a területi munkavállalót a hibáról, és a folyamat kezdődött előlről.

Helyes kitöltés esetén egy irodai alkalmazottnak továbbította a dokumentumot, akinek be kellett scannelnie, digitalizálnia a nyilatkozatot, majd egy erre a célra létrehozott mappában tárolásra került a digitalizált nyilatkozat. A digitális nyilatkozatot ellátták egy iktatószámmal, amit szintén egy irodai alkalmazott manuálisan rögzített egy excel táblázatban ahol a nyilatkozatok adatait, és azonosítóját tárolták. Ezt követő lépésként az erre a célra létrehozott e-mail fiókon keresztül elküldték az IT csoportnak a nyilatkozatot, akik a további rendszerekben elvégezték a szükséges tennivalókat.

Helytelen kitöltés esetén sokkal hosszabb időbe telt tájékoztatni a nyilatkozatot beküldő munkavállalót a hibáról, így sokkal hosszabb idő telt el egy helytelen kitöltés utáni javított dokumentum-beküldés között. A dokumentumot legtöbb esetben csak a marketing menedzser ellenőrizte tartalmilag, így sokkal nagyobb súly nehezedett a dokumentumokat kezelő menedzser precíz munkájának tényére.

A leírásban is látszik, hogy több ember kezén ment keresztül fizikailag is a nyilatkozat, illetve rengeteg időt igényelt 1-1 dokumentum digitalizálása, és manuális rögzítése a megfelelő excel táblában, majd pedig a manuális e-mail kiküldése is. Olyan esetekben, amikor 1 nap 10-20-30-40 dokumentum érkezett be, az ezzel a feladattal megbízott irodai munkavállaló teljes napja is eltelhetett a nyilatkozatokhoz köthető munkák ellátásával. Ráadásul ezek egyhangú, kreativitást nem igénylő munkafolyamatok voltak, kevés munkavállaló szeret egész nap egy irodai nyomtató mellett scannelni dokumentumokat naphosszat.

Tehát rendkívül időigényes volt a nyilatkozatok kezelése, ami monotonitást tűrő alkalmazottat igényelt. Amikor a digitalizáció, és az IT értesítése megtörtént, akkor az IT-n dolgozó munkavállalónak egy e-mailen keresztül kapott PDF dokumentum szerint kellett a további rendszerekben való rögzítést elvégeznie, megnyitva az aktuális dokumentumot, majd pedig az abban kézírással szereplő adatok szerint eljárnia. Mivel minden emberi kézírás eltér, ezért ez további munkát igényelt az IT részéről, hogy megfelelően járhassanak el minden esetben.



## Automatizált digitális folyamat

Mivel a fentiekben ismertetve rendkívül időigényes, és lassú volt a folyamat kezelése, ezért felmerült a folyamat automatizációjának lehetősége. Mivel addigra kellő tapasztalattal, és ismerettel rendelkeztem a Microsoft PowerPlatform-on belül, ezért a PowerAutomate-t választottam megoldásom eszközeül.

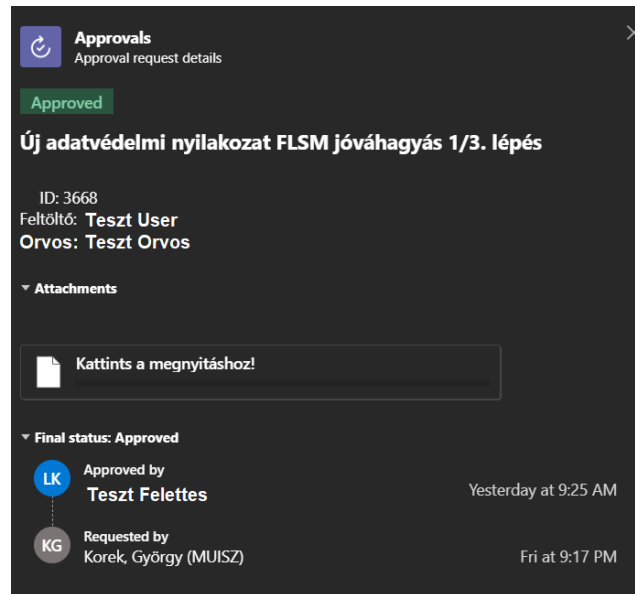


17. ábra, GDPR nyilatkozatok új, automatizált folyamatáról (saját szerkesztés)

A 17. ábrán a GDPR nyilatkozatok új automatizált folyamatának leegyszerűsítését vizualizáltam. A folyamat úgy kezdődik, hogy a területi munkavállaló feltölti egy sharepoint formon keresztül a digitálisan aláírt dokumentumot, ami egy elektronikus aláírásokat kezelő rendszeren keresztül történik papírintésen. Ezután a sharepoint formon kitölti a megfelelő mezőket, amik segítik a nyilatkozat digitalizációját. Amint befejezték a feltöltést az automatizált folyamat automatikusan elindul, és értesítést kap a feltöltő közvetlen felettese a nyilatkozat feltöltéséről e-mailben, és a felületen keresztül lehetősége van e-mailen keresztül a Microsoft Approvals megoldás segítségével elfogadni vagy elutasítani a dokumentumot.

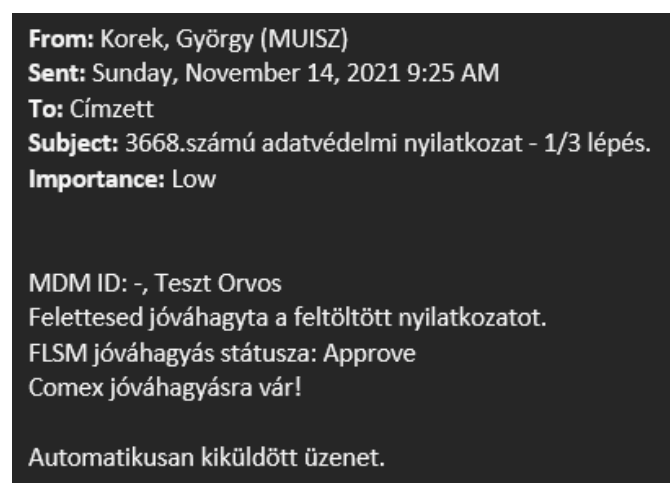
A 18. ábrán látható egy példa, hogy mit látnak a jóváhagyó személyek a Microsoft Teamsen belül. A PowerAutomate folyamat a Microsoft Approvals összeköttetését használja, ami lehetőséget nyújt arra, hogy a jóváhagyó személyek akár Microsoft Teamsen, akár Microsoft

Outlookon akár Microsoft PowerAutomate Approvals felületén is kezelhessék a rá váró jóváhagyásokat. Ez pedig szabadságot nyújt a jóváhagyóknak, hogy a nekik leginkább megfelelő felületen végezhesék a nyilatkozatok ellenőrzését. Elutasítás esetén indoklást is írhatnak, amiről automata tájékoztatást kap a feltöltő. Elfogadás esetén ugyanebben a formában tovább küldésre kerül a dokumentum és a form a marketing menedzser felé, aki szintén elutasíthatja, elfogadhatja a nyilatkozatot. Elfogadás esetén az IT csoport kap értesítést a dokumentumról, akik szintén elfogadhatják vagy elutasíthatják tartalmilag a nyilatkozatot.



18. ábra, Példa a jóváhagyás felületéről Microsoft Teamsben (saját szerkesztés)

Minden elfogadásról, és elutasításról automata értesítést kap a feltöltő, így azonnal láthatja, hogy milyen lépésnél tart vagy bukott el a nyilatkozata. Látható, hogy kézi interakció csak a feltöltéshez, és a jóváhagyásokhoz szükséges.



19. ábra, Példa a jóváhagyás utáni automata e-mailről (saját szerkesztés)

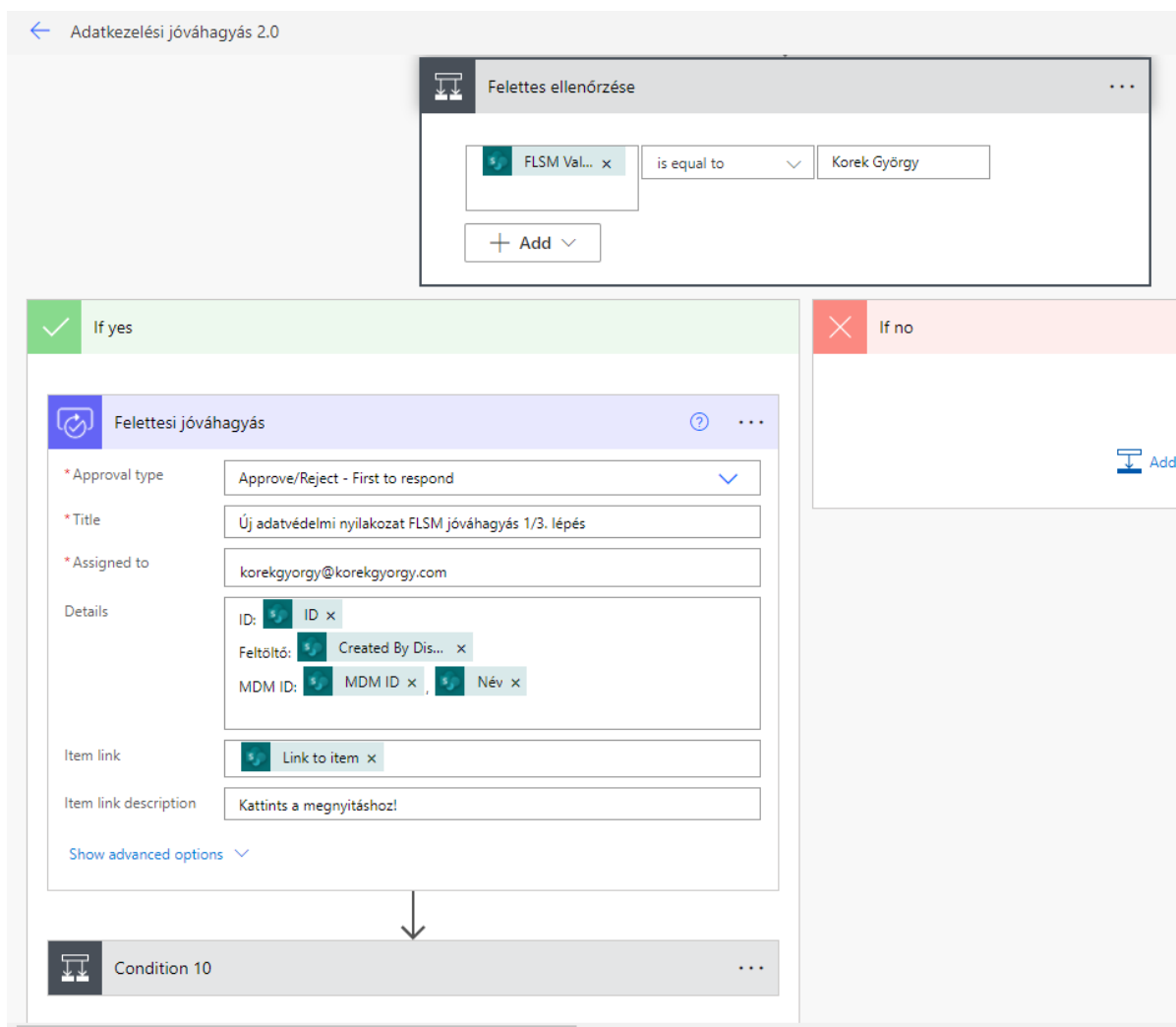
A 19. ábrán egy példa látható a 3 lépésből álló folyamat első lépésének elfogadásáról szóló automata tájékoztatásról. A dokumentum digitalizálása már a feltöltéskor megvalósul, és az egész folyamat átláthatóan, és egyértelműen követhető az erre kialakított sharepoint felületen keresztül. A feltöltő bármikor láthatja melyik nyilatkozata melyik fázisnál tart, ki fogadta el, esetleg miért utasították el. Nem szükséges a folyamatban résztvevő személyek közül senkinek sem az e-mail fiókjában keresnie a megfelelő levelezést, és nem szükséges e-mailben vagy telefonon érdeklődni sem a területi munkavállalóknak a feltöltött nyilatkozatuk állapotáról.

Aláírás dátuma	Attachments	FLSM jóváhagy...	Jóváhagyás/Elu...	FLSM elutasítás...	COMEX jóváhá...	COMEX jóváhá...	COMEX elutasit...	Kimaszkolás stá...	Kimaszkolás dá...
11/10/2021	0	Jóváhagyva	11/10/2021	-	Jóváhagyva	11/12/2021	-	Kimaszkolva	11/12/2021
9/20/2021	0	Jóváhagyva	11/9/2021	-	Jóváhagyva	11/12/2021	-	Kimaszkolva	11/12/2021
7/2/2020	0	Jóváhagyva	11/8/2021	-	Jóváhagyva	11/12/2021	-	Kimaszkolva	11/12/2021
11/2/2021	0	Jóváhagyva	11/3/2021	-	Jóváhagyva	11/12/2021	-	Kimaszkolva	11/12/2021

*20. ábra, Példa (részlet) a GDPR nyilatkozatok folyamatának követéséről sharepoint listában (saját szerkesztés)*

Az 20. ábrán látható a sharepoint lista egy részlete, melyen az követhető, hogy az FLSM jóváhagyás-felettes, Comex jóváhagyás-Marketing menedzser, és a kimaszkolás státusza-IT jóváhagyás mikor történt meg. Ezen túlmenően teljesen megszűnt a scannelésre, és excel rögzítésre delegált irodai alkalmazott része a folyamatban, mivel az iktatószám adás automatikusan történik a feltöltés pillanatában. Ennek következtében lehetőség nyílt a felszabadult humán munkaerőt más kreatívabb folyamatokra átcsoportosítani. A végső rögzítéseket elvégző IT pedig digitálisan láthatja a formanyomtatvány számára lényeges elemeit, amit a feltöltéskor adott meg a területi munkatárs egy sharepoint űrlapon keresztül, a benne szereplő adatokat pedig az IT előtt már 2 személy is hitelesítette a feltöltőn kívül. A 3 lépcsős jóváhagyással pedig jelentősen csökkent a hibásan kitöltött nyilatkozatok száma, és az utolsó lépcsőig jóval kevesebb hibás dokumentum érkezett az IT számára. Ez a fajta nagyobb többlépcsős jóváhagyás egyrészt nagyobb hangsúlyt fektet az adatvédelmi nyilatkozatok ellenőrzésére, és ezáltal lehetősége nyílt a szervezetnek úgy kezelnie precízebben a folyamatot, hogy közben minden résztvevőtől kevesebb idő, és energia ráfordítást igényel a GDPR folyamat. Ezekon túlmenően viszont, még volt egy fontos tényező, ami sürgette a hagyományos folyamat átalakítását, ez pedig a COVID-19 világjárvány volt, amikor ugyanis a szóban forgó szervezetnél a preferált munkavégzés az otthoni munkavégzés, úgynevezett home office volt.

Így a dokumentumok fizikai ellenőrzésére, és beküldésére majd a hozzá kapcsolódó manuális digitalizációjára sürgős megoldást kellett keresni, ami idő, és helyfüggetlenné teszi a nyilatkozatok kezelését. A megvalósított rendszerben bármelyik területi munkavállaló helytől, és időtől függetlenül feltöltheti a nyilatkozatot, amit a 3 lépcsős jóváhagyásban szereplő személyek szintén helytől és időtől függetlenül ellenőrizhetnek, és fogadhatnak vagy utasíthatnak el bármikor, miközben nem történik semmilyen fizikai interakció a dokumentumokhoz kapcsolódóan. Ez pedig kiemelt fontossággal bírt a személyes érintkezések csökkentésében is.

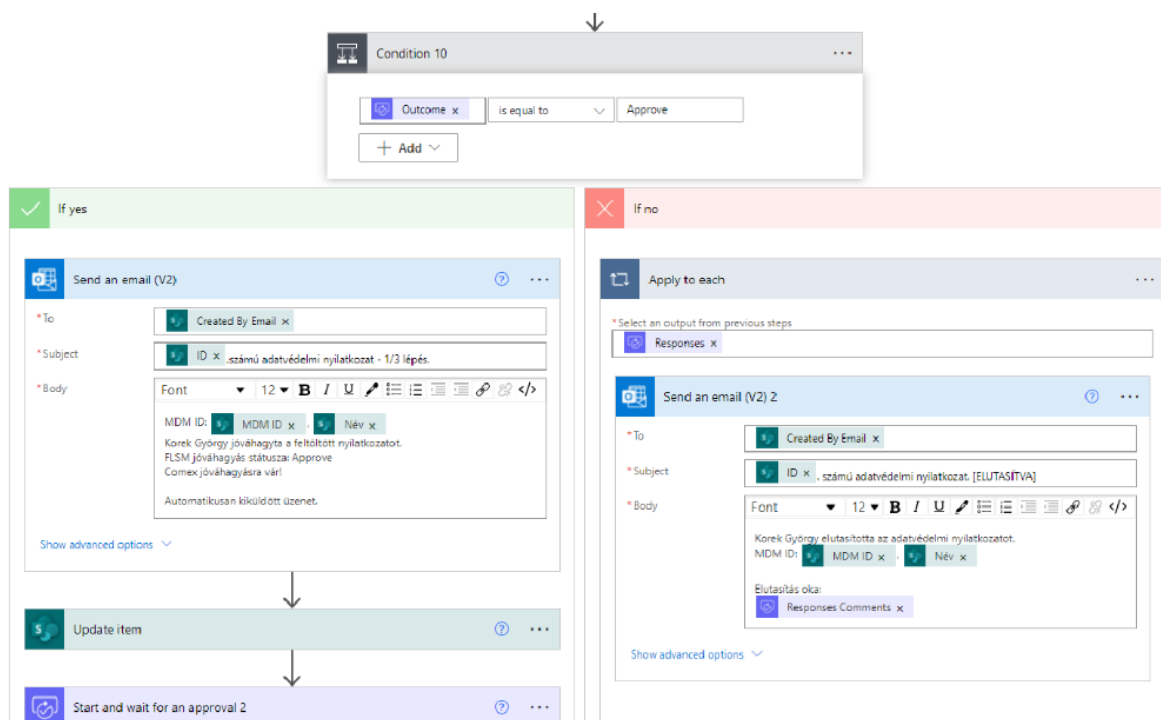


21. ábra, A PowerAutomate felületről és részlet az automatizált jóváhagyó folyamat felépítéséről (saját szerkesztés)

Az 21. ábra egy részlet a fent ismertetett PowerAutomate által automatizált folyamatból, és a PowerAutomate kezelőfelületből, ahol a folyamatok összeállítása, és megírása történik. Formai, és technikai megkötések miatt nincs lehetőségem a szakdolgozatomban teljes egészében szerepeltetni a fent ismertetett PowerAutomate folyamatot, hiszen egy nagy szabadon

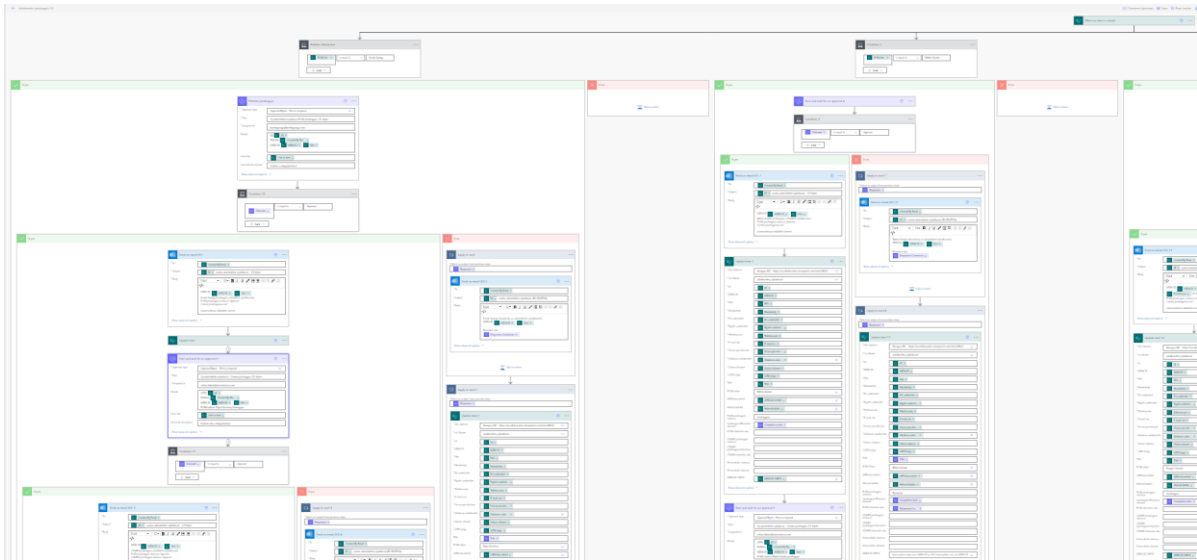
mozgatható felületen van lehetőség a folyamat összeállítására, szerkesztésére, melyet teljes egészében nem lehetséges egy A4-es lapon olvashatóan, és érthetően szerepeltetni. Ezen korlátok nyomán egyes részeit mutatom csak be külön a folyamatnak. A 21. ábrán az alábbiak látszanak:

A feltöltést követően ellenőrzi, hogy ki a feltöltő felettese (jelen esetben a Korek György ág látható). Ezután, a képen látható paraméterek mentén elindítja a jóváhagyási folyamatot majd pedig halad tovább a következő részhez.



22. ábra, A PowerAutomate felületéről és részlet az automatizált e-mail küldő folyamat felépítéséről (saját szerkesztés)

A fenti 22. ábra az 21. ábrának a folytatása. Amennyiben a jóváhagyás végeredménye – képen 'Outcome' a nyilatkozat jóváhagyása tehát 'yes' abban az esetben értesíti a feltöltőt a felettesi elfogadásról, a képen látható paraméterek mentén, majd pedig frissíti a sharepoint lista elemét a jóváhagyó személyével, időpontjával. Elutasítás esetén lényegében ugyanez a folyamat zajlik le, csak az elutasítás tényéről tájékoztatja a feltöltőt illetve az elutasítás részleteivel frissíti a sharepoint lista elemét. A további kettő jóváhagyási folyamat lényegében ugyanezen módszereken, megoldásokon és logikán alapszik így azokról nem mellékelek külön ábrát.



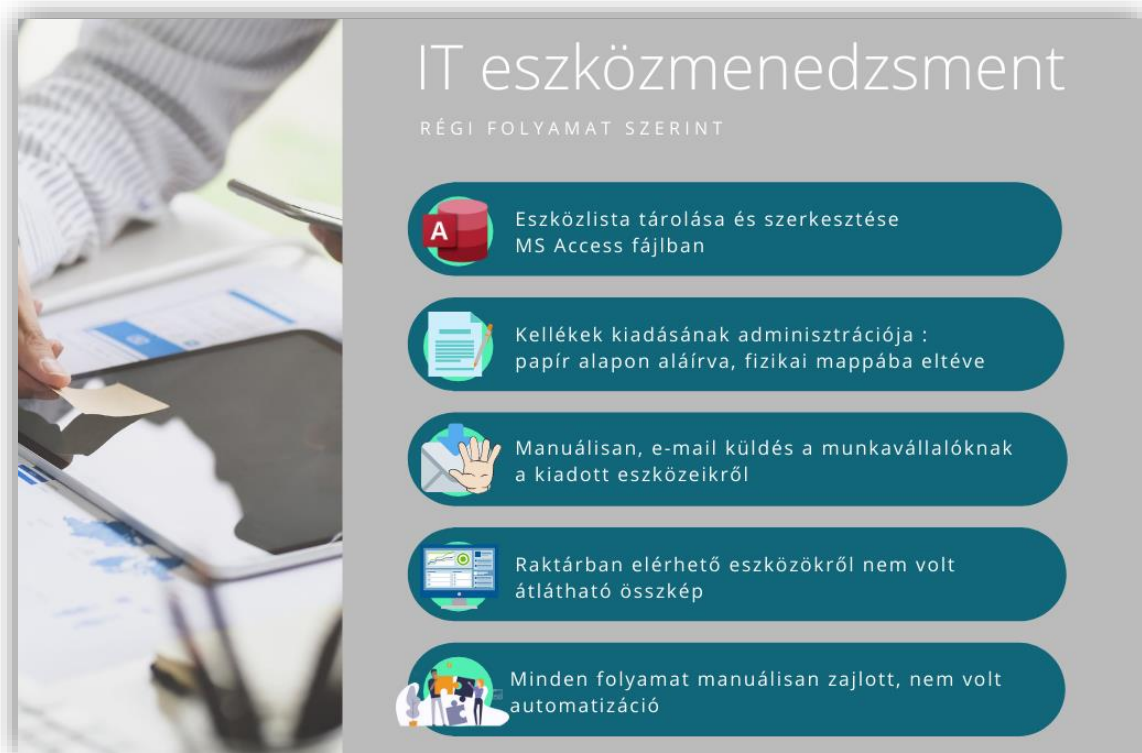
23. ábra, Részlet a GDPR folyamat összetettségéről PowerAutomateben (saját szerkesztés)

A 23. ábrán látható a PowerAutomate összetett kezelőfelülete, valamint szemlélteti a fent ismertetett digitalizált folyamat komplexitását, hogy a fenti ábrán a folyamat körülbelül 10%-a sem látható, ennél kisebb értéken pedig nem enged lehetőséget a böngésző szemlélni az oldalt. Viszont kitűnően kivehető az ábrán a feltételek több szálon való kezelésének lehetősége, valamint a folyamatot alkotó elemek egymásra épülése.

#### 4.3 Esettanulmány: IT eszközmenedzsment alkalmazás - PowerApps

Második bemutatandó esettanulmányom az IT eszközmenedzsmentjéhez és raktárkészlet kezeléséhez kapcsolódik. Az IT csoport számára minden szervezetben belül meghatározó feladat, és folyamat az eszközökhöz kapcsolódó ügyintézés és azok kezelése.

Az eszközökhöz kapcsolódó ügyintézés több szempontból is elavult és régi hagyományokat követett, amit elengedhetetlen volt digitalizálni, hogy az egyre növekvő IT feladatok mellett ne okozzon terhelést a régi folyamattal járó manuális lépések ellátása.



24. ábra, A régi, hagyományos IT eszközmenedzsmentről (saját szerkesztés)

### Régi, hagyományos folyamat

A 24. ábrán vizuálisan pár lépésben szemléltettem a régi folyamat egyes tényezőit. Az eszközöket egy Microsoft Access fájlban adminisztráltunk, és az ezekhez kapcsolódó adminisztrációt, tulajdonosváltást, új eszköz felvitelét, stb. ebben rögzítettük.

Az access állományban nem rögzített eszközöket, amiket kelléknek neveztünk külön átadásvételi lapon 2 példányban kerültek átadásra, melyből 1 példány az átvevőhöz 1 példány az IT fizikai irattartójába került. Kelléknek minősül az adott szervezeten belül a fejhallgató, egyéb kábelek, töltők, stb.

Bármilyen ügyintézés esetén az IT munkatárs manuálisan küldte el a listát az érintett munkavállalónak az eszközeiről, kiemelve a változásokat. A raktárban elérhető eszközöket legtöbb esetben egyszerűbb volt megnézni magában a raktárban, mint az access állományt megnyitva különböző szűréseket végezni a táblázaton. Az access fájl további nehézségeket okozott az adminisztrációban, mint például, hogy egyszerre csak 1 ember szerkeszthette a fájlt, ellenkező esetben duplikáció keletkezett a Onedrive felhőtárhelyen a fájlból ami kettő különböző verzió párhuzamos jelenlétét eredményezte.

Többször nehézséget okozott visszakövetni ezekben az esetekben a változtatásokat, és azokat egyesíteni az eredeti állományban. További akadályt okozott az access állomány lassú megnyílása, és felépítése, ami nem tette könnyebbé az adminisztrációs lépéseket.

A kellek kiadásának papír alapú adminisztrációja mind az IT, mind az átvevő részéről lassan zajlott, valamint összességében sok papír felesleges kinyomtatását, és tárolását eredményezte, melyeknek visszakeresése rendkívül időigényes folyamat volt az irattároló rendezettségétől függetlenül. Amikor egy munkavállaló, vagy annak felettese bármi okból kíváncsi volt az IT által jegyzett eszközeinek listájára, akkor egy e-mailt küldtek az IT részére azzal a kéréssel, hogy az IT küldjön egy e-mailt, ami tartalmazza a munkavállaló eszközeinek összesítését.

Összességében tehát látható, hogy minden lépésnél meghatározó szerepe volt a manuális munkavégzésnek, aminek következtében egy folyamatosan növekvő vállalatnál egyre több időt tett ki a nem hatékonyan végzett adminisztratív jellegű folyamatok ellátása.

Másik fontos tényező szintén a COVID-19 világjárvány volt. Noha az eszközkészítéshez kapcsolódó folyamat modernizálása régóta szerepelt az IT csoport napirendjén, a végső löketet és a gyors megvalósítást szintén a globális világjárvány sürgette meg. Ebben az időszakban minden olyan megoldásra szüksége volt az IT-nak ahol időt tud spórolni, és azt allokálni más tevékenységek, igények fedezésére. Valamint nem volt opció többé a papír alapú átadás-átvétel a személyes kontaktok csökkentése érdekében.

A globális chiphiány okozta problémák is érintették az IT csoportot olyan szempontból, hogy sokkal fontosabbá vált a raktárban elérhető eszközök elérhetőségének, raktárkészlet vizuális formában való áttekinthetősége. Ez azt jelentette, hogy az IT munkatárs egy egyszerűen és gyorsan átlátható dashboardon keresztül bármikor megtekinthesse milyen eszközből mennyi érhető el a szervezet raktárában, és a dashboard szerint miből van jelenleg alacsony készlet.

Sokkal több előrelátásra és pontos készletismeretre volt szüksége az IT-nak, hogy a chiphiány következtében fellépő megnövekedett szállítási idő mellett is ki tudja elégíteni az újonnan belépő és a régebb óta állomány részét képező munkatársak eszközigényeit.



## Az új digitalizált és automatizált folyamat

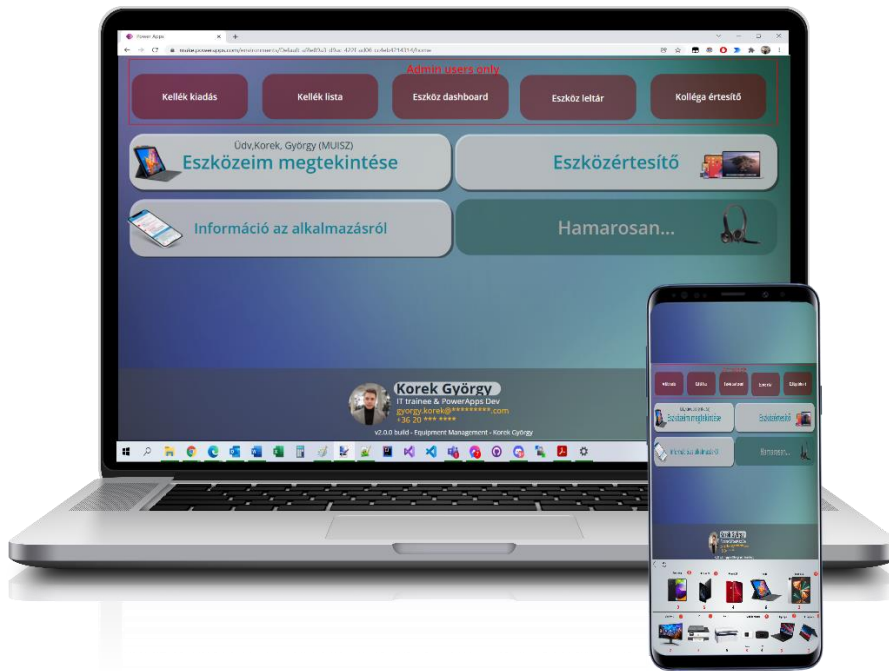
Az új folyamat alapjait a Microsoft PowerApps és a PowerAutomate közösen alkotják.



25. ábra, Az új, digitalizált IT eszközmenedzsmentről (saját szerkesztés)

A 25. ábrán látható röviden összefoglalva néhány része az átalakított eszközmenedzsment folyamatnak. Tehát elsődleges lépés volt az MS Access állományt és annak tartalmát migrálni egy felhőalapú megoldásba, ennek háttéréül pedig ebben az esetben is a sharepoint listát választottuk.

A felület megjelenéséért a PowerApps felel, miközben több automata folyamat logikája PowerAutomate segítségével került megvalósításra. Egy olyan megoldásra volt szükségünk, ami nem csak az IT-nak hasznos, hanem a szervezet munkavállalói számára is funkcióval bír. Az alkalmazás felülete úgy lett kialakítva, hogy hordozható eszközökön, tableten, mobilon is megjeleníthető, és használható legyen, de a fő platform az a számítógép/laptop maradt, mert ezt az eszközt használják a szervezeten belül legtöbbször a munkához kapcsolódó ügyintézésre jelenleg.



26. ábra, A PowerApps alapú eszközmenedzsment alkalmazásról lapon és telefonon (saját szerkesztés)

Az 26. ábrán az eszközező alkalmazás kezdőlapja látható számítógépen és telefonon egyaránt. Látható, hogy van egy elkülönített 'admin user' rész, amit kizárólag az IT csoport meghatározott munkavállalói érhetnek el, a szervezet többi tagja ezekhez a menüpontokhoz és funkciókhoz nem férhet hozzá. Az alkalmazás az alábbi funkciókat képes kezelni, és ellátni. A kellékek kiadásának adminisztrációja szintén az alkalmazáson keresztül történik, ami automata e-mailt küld az átvevő félnek a változtatás rögzítésekor, a kiadott kellékeknél pontosan követhető, hogy mikor várja az IT annak visszahozatalát, vagy, hogy az elmúlt időben milyen kellékek lettek kiadva a munkavállaló számára.

```

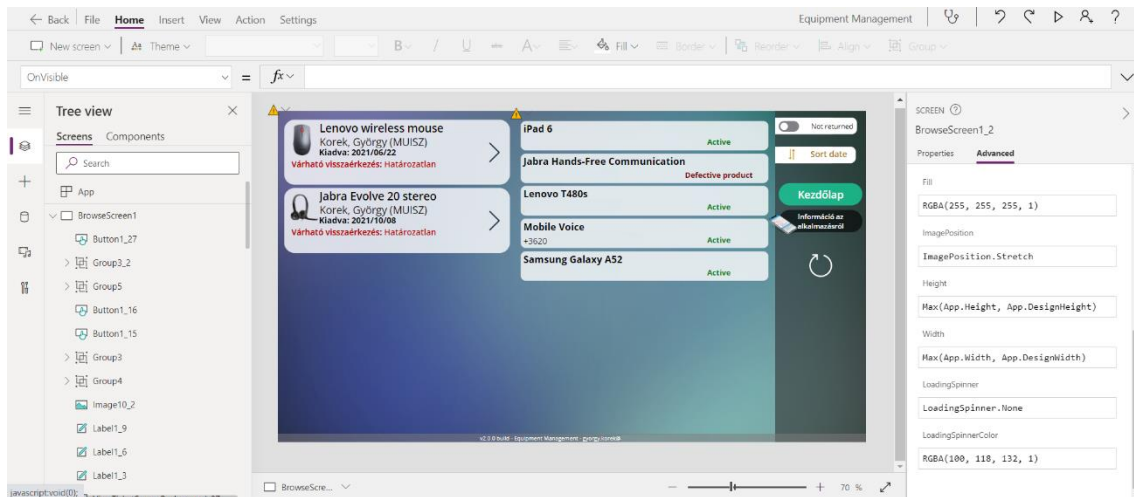
fx Office365Outlook.SendEmail(
    DataCardValue10.Selected.Email,
    "Információ a kiadott eszközökről - Equipments",
    "Kedves Kolléga!

Az alábbi kiegészítő eszköz(öke)t adtuk ki számodra a mai nappal: " & DataCardValue7 & ". " & Char(10) & "Szériaszámok: " &
DataCardValue9 & ". " & Char(10) & "Eszköz visszaérkezési határideje: " & DataCardValue11 & Char(10) & Char(10) & "A levélben szereplő
eszközök kiadását és átvételét elfogadottnak tekintjük. Bármilyen eltérés esetén kérlek jelezz a +HUTB IT (HUTBIT@astrazeneca.com)
címen. " & Char(10) & "Üdvözléssel,
Hungary IT: IT@ .com",
    {Cc: ' IT@ .com'}
);
SubmitForm(EditForm1);
Navigate(BrowseScreen1_1);

```

27. ábra, Példa a low code scriptre PowerApps alkalmazáson belül, automata e-mail küldés adminisztráció után (saját szerkesztés)

A 27. ábrán az a kódsor látható, ahol paraméterezésre került a PowerApps-en belül az automata e-mail küldése kellékanyag kiadásának pillanatában. Ezen az ábrán megfelelően tudom szemléltetni, hogy mitől low code platform a Microsoft PowerApps, hiszen látható, hogy saját kódok írására, és futtatására van lehetőség, nem csak építőelemekből való építésre.



28. ábra, A PowerApps fejlesztői felületéről és az alkalmazás munkavállalók számára elérhető részéről, a saját eszközlista oldalról (saját szerkesztés)

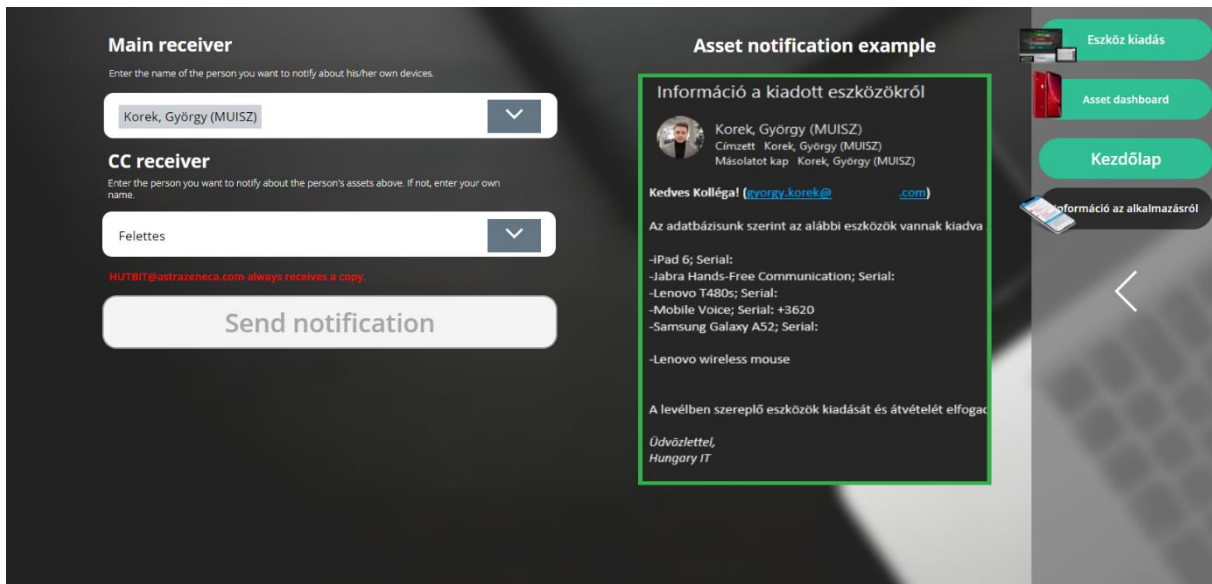
A 28. ábrán nem csak a PowerApps kezelőfelületét szeretném röviden bemutatni, hanem a fent bemutatott kellékiadáshoz kapcsolódó felhasználói felületet, amit az átvevő munkavállaló bármikor tételesen megtekinthet, milyen eszközök vannak a neve alatt, azok mikor kerültek kiadásra, és van-e velük bármi teendője.

Mellette az állandó eszközök is láthatóak, mint a laptop, telefon, telefonszáma, ahol további tájékoztatásokra tehet szert, például a mobilszáma PIN kódjának megtekintésére. Tételesen látható bármikor bárhol, hogy milyen eszközök vannak a munkavállaló neve alatt, ezzel sikerült azt a folyamatot is digitalizálni, amikor egyes okok miatt munkavállalók listát kértek az általunk a neve alatt jegyzett eszközök listájáról, az alkalmazásban ezt bármikor megtekinthetik a kellék eszközökkel, mint a fejhallgató és egérrel együtt.

A kellékekről sok esetben leterheltség vagy egyéb okból nem is tudtunk a régi folyamat szerint a papír alapú ügyintézés miatt tájékoztatást adni, azonban ez is megoldásra került az alkalmazáson segítségével.

Természetesen felmerültek korábban olyan igények is, hogy a munkavállaló főnöke, vagy a szervezet HR osztálya vagy egyéb munkavállalója kaphasson listát egy személy eszközeiről és kellékeiről. Ezek az igények többnyire egyes kilépéshez kapcsolódó folyamatokhoz voltak

köthetőek, amikor a COVID-19 vilájárvány alatt preferált home office munkavégzés alatt nem mindig volt bent az irodában IT munkatárs, hogy átvegye a kilépő eszközeit. Így kialakításra került egy rendszer ahol az IT munkatárs megadhatja a személyt, akiről listát szeretne generálni, valamint egy CC-t aki másolatot kap a munkavállaló eszközeiről. Ez a funkció bizalmas adatok miatt IT Admin jogosultsággal érhető csak el az alkalmazásban, nem kérheti le senki saját magán kívül más adatait az IT beavatkozása nélkül.

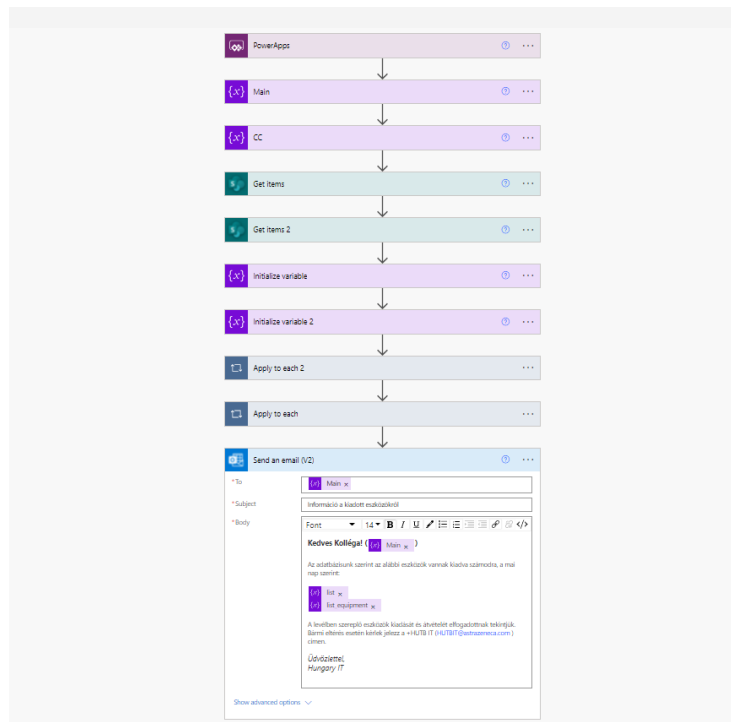


29. ábra, A PowerApps alapú eszközmenedzsment alkalmazásom kolléga értesítő felületéről (saját szerkesztés)

A 29. ábrán látható az IT Admin jogosultsággal elérhető kolléga értesítő felület. Itt példaként az látható, hogy Korek György adatairól generál a PowerAutomate folyamat egy listát az eszközeiről és kellekeiről amit e-mailben elküld automatikusan Korek Györgynek, és a megadott másolatot kapó második személynek (ábrán a „Felettes” mező jelöli ezt a részt), valamint az IT -nak. Az ábra jobb oldali részén zöld keretben látható egy példa e-mail, ilyen lista küldésére van lehetőség.

Tapasztalataink szerint ez a funkció is jelentősen csökkentette az IT leterheltségét az olyan kilépési folyamatok alatt, ahol harmadik félnek volt szükséges elküldeni a kilépő munkavállaló eszközlétét, hogy az a személy tisztában lehessen arról, milyen eszközöket kell pontosan átvennie. Régi folyamat szerint ez bonyolult MS access szűrést igényelt, majd kézi másolást és e-mail küldést Outlook-on keresztül. Ez az új megoldással 3 kattintásra egyszerűsödött, valamint szintén tartalmazza a kellekek listáját is az eszközök után. Természetesen ennek háttéréül egy PowerAutomate folyamat felel, ami egy előre paraméterezett PowerApps gomb megnyomása esetén kerül meghívásra. A 29. ábrán például a „Send notification” gomb

megnyomása meghívja az eszközértésítéshez kapcsolódó PowerAutomate folyamatot, és átadja a 2 paramétert a felületről, a main és a CC felhasználót, majd a folyamat ezeket az adatokat használva kerül lefutásra.



30. ábra, Részlet a munkatársértesítő e-mail PowerAutomate részéről. (saját szerkesztés)

A 30. ábrán látható a fent ismertetett folyamathoz kapcsolódó PowerAutomate folyamat. Mivel ismét technikai korlátai vannak annak, hogy egy teljes folyamatot megjeleníthessek egy A4-es lapon, ezért csak egy részletét tartalmazza az ábra. Viszont látható, hogy egy viszonylag egyszerűnek tűnő folyamat is több összetett elemből épül fel. Valamint kitűnő példája annak, miként lehet a PowerApps és a PowerAutomate lehetőségeit kihasználni azok együttműködésével és integrációjával. Előző fejezetekben ismertetett elv, miszerint a PowerApps a front-end megjelenítésért, és a PowerAutomate a back-end logikáért felelős.

A fenti kolléga értesítő folyamat elérhető olyan formában is, hogy a munkavállaló e-mailben lekérhesse a saját listáját, ezt elvégezheti az alkalmazásban saját maga is, viszont kizárólag a saját listáját kérheti le magának, más személy listáját nem kérheti le e-mailben.

Az eszköz dashboard felület szintén egy digitális megvalósulása az IT raktárában elérhető eszközök vizuális megjelenítésének. Mint az esettanulmány elején ismertettem, sokszor problémát okozott, hogy az eszközök pontos elérhető számáról nem volt könnyen átlátható áttekintés. Erre fejlesztettem megoldásul az alkalmazáson belül IT Admin jogosultsággal elérhető eszköz dashboard felületet, ami megoldást nyújt a korábbi áttekintés problémájára.



31. ábra, Eszközmenedzsment alkalmazásom eszköz dashboard felülete. (saját szerkesztés)

A 31. ábrán látható az alkalmazásból elérhető eszköz dashboard felület, amin kiemelt eszközökre bontva teszi áttekinthetővé raktárkészlet állapotát. Felkiáltójellel jelzi az alkalmazás, ha az alkalmazás előre paraméterezett beállítása szerint feltehetően kevés az elérhető raktárkészlet az adott eszközből. A felület mindig az aktuális készletet mutatja valós időben.

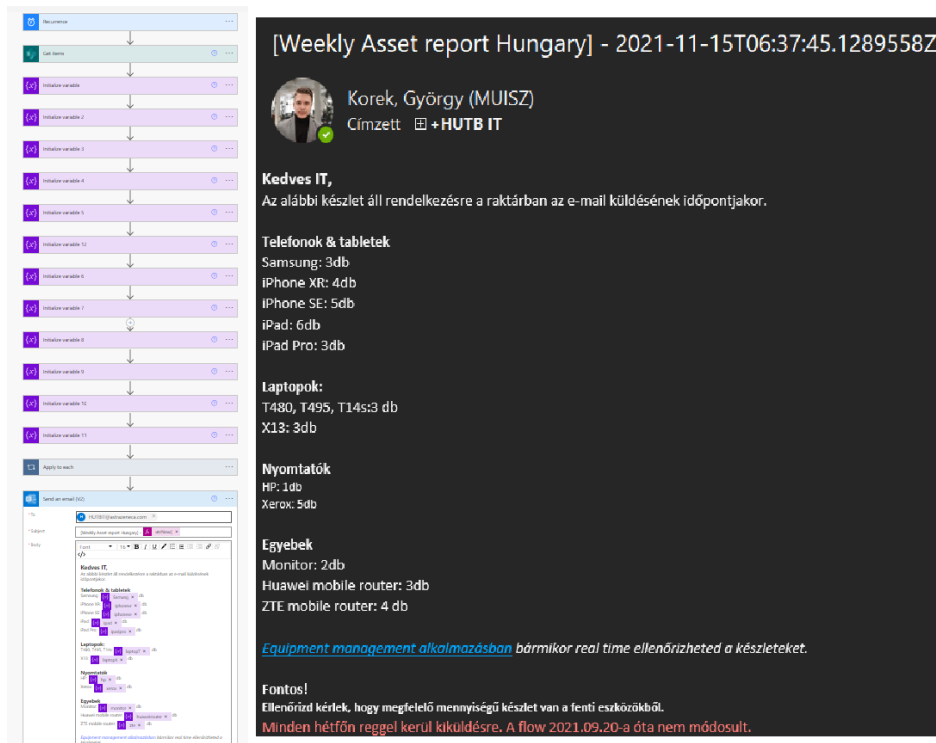
```
CountIf('Asset List - Hungary', Title = "iPad Pro (2nd generation)" && Owner.DisplayName = "VistaHU, MC" && Condition.Value = "Active" || Title = "iPad Pro" && Owner.DisplayName = "VistaHU, MC" && Condition.Value = "Active" )
```

32. ábra, Példa a low code scriptek lehetőségére az eszköz dashboard felület eszközszámláló funkciójában. (saját szerkesztés)

A 32. ábrán szintén a saját kódok futtatásának lehetőségét szeretném kiemelni, ez különbözteti meg a low code platformokat a no code platformoktól, ahogy a bevezetőben ismertetésre került a különbségük. Az ábrán az egyes eszközök számának a lekérdezése látható.

A dashboard részét képezi egy PowerAutomate által vezérelt automata értesítő e-mail, ami minden hétfőn reggel egy összesített listát küld a közös IT e-mailcímre a raktárban elérhető eszközökről, melyben figyelmezteti adott esetben az IT csoportot, ha egy eszköztípusból úgy ítéli a paraméterezett PowerAutomate folyamat, hogy kevés a raktárkészlet. Így hét elején szintén egy gyors betekintést nyújt e-mail formájában a raktárkészlet alakulásába, ezzel jelentősen csökkent annak a hibaforrásnak a lehetősége, hogy egy eszköztípusból kifogy a készlet. Így pedig hatékonyabbá vált a raktárkészlet kezelése.

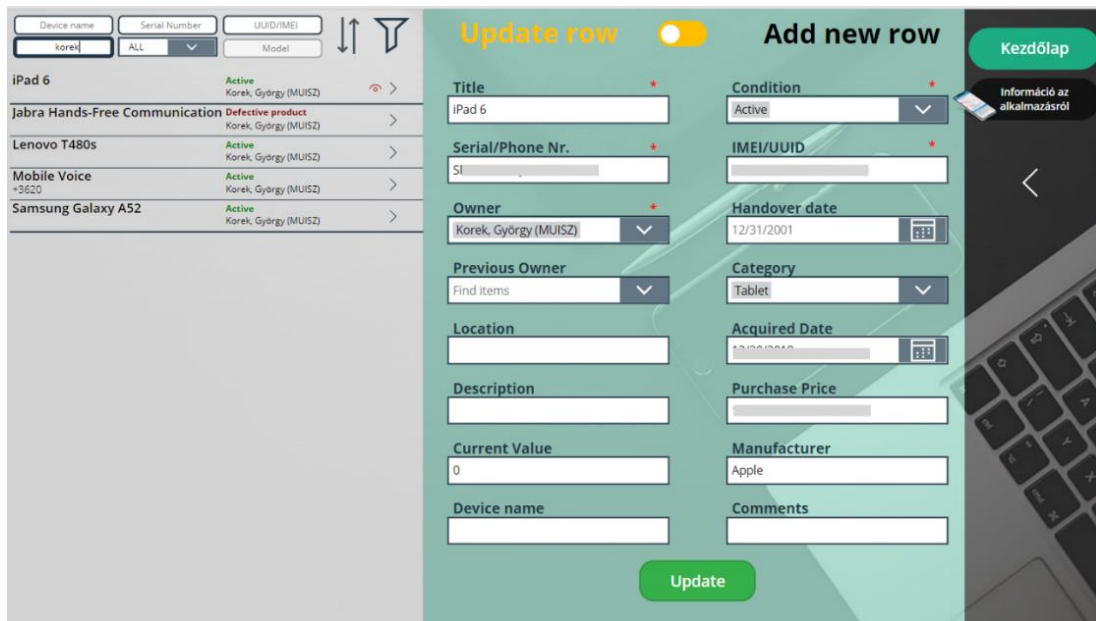




33. ábra, Részlet a heti eszközösszegző készletinformációs automata e-mailről és a PowerAutomate részéről. (saját szerkesztés)

A 33. ábra bal oldalán látható a hétfőnkénti automataértesítő PowerAutomate folyamatom. Jobb oldalán a példa e-mail amit minden hétfőn reggel kap az IT csoport készletinformációként. A dashboard, illetve az automata e-mail értesítéssel sikerült megoldást találni a korábban felmerülő készletkezelési problémáinkra.

Végére hagytam az alkalmazásom bemutatásában a felületet, ahol a lényegi adminisztráció történik IT Admin oldalról. Ahogy az esettanulmányom bevezetésében ismertettem, a korábban használt MS Acces állomány kiváltására kerestünk megoldást olyan formában, ami minden addig felmerült problémára megoldást nyújthat. Így a jelenlegi felületet egyszerre több IT felhasználó is használhatja, párhuzamosan szerkeszthetnek elemeket, és adhatnak hozzá eszközöket a listához egyszerre többen is anélkül, hogy bármilyen problémába ütköznenek. Az alkalmazás hátteréül szolgáló sharepoint lista alapú adatszerkezet gyors elérést tesz lehetővé az alkalmazáson keresztül. Az MS Access minden addig használt funkcióját megvalósítottam a PowerApps alkalmazásban, hogy az IT munkatársak ne érezzenek semmilyen szempontból sem visszaesést a korábban használt megoldáshoz képest. A felületet úgy alakítottam ki, hogy egy egyszerű és átlátható formában legyen szűrhető az eszközlista több szempont alapján, mint például: tulajdonos, szériaszám, eszköz állapot, IMEI azonosító, eszköznév.



34. ábra, Eszközmenedzsment alkalmazásom adminisztrációs felülete. (saját szerkesztés)

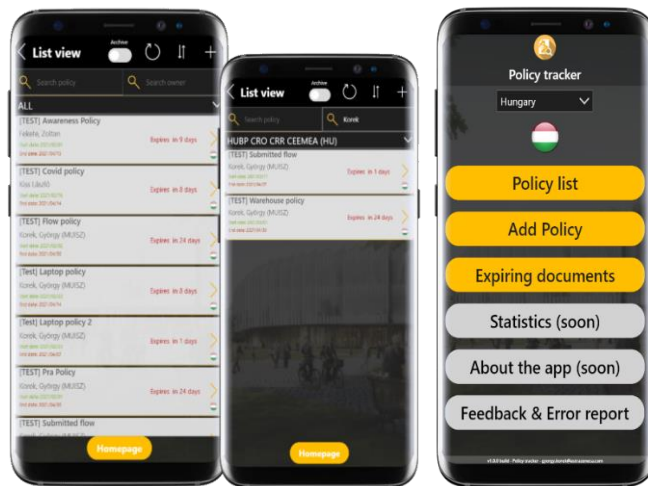
A 34. ábrán látható az eszközlista szerkesztésére vagy eszközök hozzáadására kialakított felület, ami az adminisztrációs folyamatok lehetővé tételéért felelős. Kiemelt szempont volt, hogy a felület az MS Access vagy egyéb excel alapú megoldásokhoz képest átláthatóbb, logikusabb és könnyebben kezelhető legyen. Bal oldalon látható a tételes eszközlista a szűrésnek megfelelően, az ábrán a saját nevemre szűkítettem az eszközök listáját. Látható melyik eszköz került kiválasztásra, és a jobb oldali felületen egyszerűen szerkeszthetőek az adatok. Adatvédelmi okokból egyes sorokat kitakartam az ábrámon.

Az alkalmazás tehát több, mint egy egyszerű eszközmenedzsment szoftver. Adminisztráción túl a munkavállalók megtekinthetik a szükséges információkat. Az adminisztrációs folyamat értesítéshez kapcsolódó részei automatizáltan történnek, kollaboráció lehetőségét sem akadályozza az alkalmazás, ha egyszerre több IT munkatársnak van eszköz adminisztrációs feladata. A régi hagyományos folyamatban feltárt minden ésszerűtlen, nem hatékony lépést egy hatékony automatizált vagy minimum jelentősen egyszerűsített lépés, funkció váltotta. Ezzel a megoldással nem csak az IT csoport munkája lett jelentősen megkönnyítve, de a nyomtatáshoz kapcsolódó ökológiai lábnyoma is csökkent a munkavégzésnek. Miközben gyorsan és hatékonyan tudunk reagálni a COVID-19 által megváltozott munkavégzéshez és feladatokhoz a szervezet, és a munkavállalók igényeit szem előtt tartva.



## 4.4 Esettanulmányok rövid összefoglalása

A fent bemutatott két általam fejlesztett low code megoldás lehetőséget nyújtott arra, hogy szakdolgozatomban egyedi gyakorlati példákon mutathassam be az értekezésem első részében elméleti szinten ismertetett témakört. A fenti esettanulmányok jól vizsgáztak, a gyakorlati tapasztalatom alapján, gyors, hatékony és hasznos összetett alkalmazások fejlesztésére van lehetőség a low code platformok segítségével. Ezt nem csupán az interneten nyilvánosan elérhető többi esettanulmányra tudom alapozni, hanem a saját szakmai tapasztalatomra is. A fenti esettanulmányok mellett a szakdolgozatom írásáig az említett multinacionális gyógyszeripari vállalatnál összesen 11 darab PowerApps alkalmazást és 30 automatizált PowerAutomate folyamatot készítettem, melyekből több célmegoldás európai szinten van használatban.



35. ábra, Policy tracker PowerApps alkalmazásom mobilos felülete. (saját szerkesztés)



36. ábra, CRM PowerApps alkalmazásom felületei, különböző platformokon. (saját szerkesztés)

## 4.5 Low Code határai

Ahogy a bevezetésben említettem vannak határai a low code platformoknak, mindegyiknek más az erőssége vagy a technikai megoldása. Ezért összegzésképpen, anélkül, hogy az összes low code platform erősségeiről és hátrányairól külön írnék, a következőket tudom kiemelni tapasztalataim alapján. Olyan munkafolyamatokra, ahol rendkívül fontos az alkalmazás és az adatátvitel precíz és gyors adatátvitele, ahol kiemelten fontos az alkalmazás végrehajtási ideje, például nagy gyártóüzem gyártási adatbázisát kezelő szoftver esetében, nem nyújt megoldást a napjainkban ismert low code platformok technológiája. A low code platformok eltérő előnyökkel, hátrányokkal és lehetőségekkel rendelkeznek, ezért sem a platformok diverzitása, sem a technológia folyamatos fejlődése miatt, egyértelmű, és általános határait nem lehet meghatározni a low code development platformoknak.

Ahogy a bevezetőben, az ismertetett nyilvános esettanulmányok és a saját példáimon keresztül is bemutattam, a low code platformok megoldásai nem váltják ki a hagyományos értelemben vett szoftverfejlesztést, hanem olyan folyamatok digitalizálására, megújítására nyújtanak lehetőséget, amikre a napjainkban tapasztalható szoftverfejlesztőhiány esetében nem jutna erőforrása sok szervezetnek. Így nem lehet egymás riválisainak kijelenteni a low code, és a pro code technológiákat, ezek napjainkban egymást kiegészítve segítik a világunk digitalizációját és fejlődését. (Józsa, 2018)

## 5. Mélyinterjú

### 5.1 Interjú ismertetése

Ahhoz, hogy még részletesebb képet kaphassak a low code platformok szerepéről az üzleti alkalmazásfejlesztés területén, kutatásom eszközüül mélyinterjút készítettem kettő hazai, különböző szektorokban tevékenységet folytató cégnél dolgozó tapasztalt senior IT középvezetővel. Kutatásom során legfőképpen külföldi szakirodalomra támaszkodhattam, mert csekély számú magyar forrás volt fellelhető a témában. Ahhoz, hogy még diverzifikáltabb képet kaphassak a témában, fontosnak éreztem bevonni az értekezésembe a hazai felfogás és gondolkodásmód egy részét. Az interjúk a COVID-19 vírushelyzet 2021. novemberi aktív jelenléte miatt Microsoft Teams-en keresztül kerültek lebonyolításra. A személyeket a következő fejezetben ismertetem részletesen, de anonim módon.

Az interjú szerkezetét és a fő kérdések felépítését az 1. számú melléklet tartalmazza.

Röviden ismertetve az interjú felépítését, az alapvetően 3 részre tagolható. Az interjú első blokkjában az interjúalanyok szakmai életútjáról beszélgettünk, végzettségükről, korábbi és jelenlegi munkatapasztalataikról, fejlesztésben töltött múltjukról, valamint a betöltött pozícióikról. Második blokkban a low code development platformok kerültek előtérbe, így az ezeket a technológiákat érintő kérdésekről beszélgettem interjú partnereimmel. Többek között, hogy mikor találkoztak ezekkel a platformokkal és hogyan, mi a véleményük ezekről a platformokról mik a személyes tapasztalataik velük. Harmadik, egyben lezáró blokkban pedig arról kérdeztem interjúalanyaimat, hogy a jelenlegi piaci trendeket, digitalizációs fejlődést figyelembe véve milyen jövőt jósolnak ezeknek a platformoknak és milyen veszélyeket látnak bennük.

Az elvégzett kutatásomnak az alábbiak a legfontosabb központi kérdései:

- Miként látja a low code development platformok jövőjét kettő hazai IT vezető?
- A hagyományos programozást kiváltja vagy kiegészíti a low code development platformok megoldásai a megkérdezettek szerint?

## 5.2 Interjúalanyok bemutatása

Ahogy az interjú ismertetésében említettem, két senior IT középvezetővel készítettem mélyinterjút. Ahhoz, hogy a későbbi elemzésben és értékelésben egyszerűen tudjak hivatkozni a személyekre, az alábbi megnevezéseket használom a továbbiakban.

**„A” interjúalany:** Jelenleg egy multinacionális gyógyszeripari vállalat közép európai régiójának IT menedzsere. Nemzetisége szerint magyar és villamosmérnöki végzettséggel rendelkezik, egy rövid ideig dolgozott is a végzettsége szerinti pozícióban egy magyar vegyipari vállalatnál. Ezt követően tesztmérnökként dolgozott egy IT eszközök gyártásával foglalkozó cégnél, ebben a pozíciójában ismerkedett meg a programozással és a fejlesztéssel a Borland Delphi programozási nyelv által. Felfedezett igények alapján Delphi-ben fejlesztett kisebb alkalmazásokat a cég számára. Következő szakasza a szakmai életútjának egy kisebb magyar szoftverfejlesztő cégnél folytatódott, ahol többnyire már Java-ban fejlesztett egy meghatározó telekommunikációs cég számára. Ezután dolgozott Németországban és Angliában egy indiai háttérű multinacionális információtechnológiai cégen keresztül Delphi fejlesztőként, valamint már project management funkciókat is betöltve. Dolgozott Magyarországon is aktív tevékenységet folytató banknál vezető fejlesztőként, főként Java környezetben. A bankszektori állást követően pedig egy multinacionális informatikai cégnél látott el több különböző first line

manager pozíciót. Végül pedig a jelenlegi munkahelye következett, ami a fent ismertetett multinacionális gyógyszeripari vállalat regionális IT vezetése.

**„B” interjúalany:** Jelenleg egy multinacionális autóiipari vállalat egyik termékcsoportjának a Development Leader pozícióban dolgozó vezetője. Magyar nemzetiségű, végzettsége „A” személyhez hasonlóan villamosmérnöki, eszerint kezdett el dolgozni a jelenlegi munkahelyén több mint 13 évvel ezelőtt különböző mérnöki pozíciókban. Több különböző munkakört is betöltött már a munkahelyén az itt töltött ideje alatt, és szemtanúja volt a digitalizáció elterjedésének egy multinacionális vállalatban belül. Programozással és fejlesztéssel is mérnöki munkái során találkozott. Kezdetben Visual Basic-ben fejlesztett, később a C# (C Sharp) népszerűségének emelkedése miatt (14. ábra) kipróbálta az utóbb megnevezett programnyelvet, és a későbbiekben ebben fejlesztett különböző megoldásokat, ami a mérnöki munkák megkönnyítéséhez hozzájárultak. Nem utolsó sorban szükségesnek érezte tartani a lépést a terjedő technológiákban, ez pedig a C# és a fejlesztés felé is vezette részben.

### 5.3 Interjúk eredményei, következtetés

**„A” interjúalany** jelenlegi munkáltatójánál a Microsoft PowerPlatform bevezetésre került, ez a cég által kötött Microsoft Office365 licencszerződés keretében jelent meg a használható eszközök között, amikor a gyógyszeripari vállalat pár évvel ezelőtt mélyebben implementálta a Microsoft ökoszisztémáját a működésébe. Fejlesztői tapasztalatai miatt felkeltette az érdeklődését a PowerApps, mert a korábbi fejlesztői tapasztalataiból fakadóan kicsit a Delphi-hez hasonlóan vélte elsősre a technológiát, annak egyszerűségét és sokszínűségét látva benne. IT vezetőként tisztában volt azokkal a folyamatokkal és használt régi, nem támogatott technológiákkal, amiknek a kiváltása időszerűvé vált, ennek érdekében régóta keresett megfelelő megoldásokat a régiak kiváltására. Elmondása szerint teljes mértékben az igény hozta el azt, hogy mélyebben megismerkedjen a PowerPlatform adta lehetőségekkel. Meggyőző volt számára a fejlesztés egyszerűsége és gyorsasága. A napainkban népszerű agilis vagy RAD módszertanokba abszolút illeszkedik szerinte, teljes mértékben kiszolgálja a minimum viable product (minimális működőképes termék) elvét, ami a gyors, de előre kevésbé megtervezett (11. ábra) fejlesztést támogatja a szigorú megtervezéssel (12. ábra) szemben, így véleménye szerint a napjainkban egyre jobban terjedő módszertanok gyakorlati megvalósulásához kitűnő táptalajt szolgáltathatnak a low code platformok.

Előnyeihez is többek között ezt sorolja, azon emberek számára, akik rendelkeznek informatikai érzékkel, olyan eszközt kapnak ezen platformokon keresztül a kezükbe, amivel olyan

megoldásokat hozhatnak létre, amivel a saját vagy a csoportjuk munkáját jelentősen megkönnyíthetik. Azt az elképzelést és gondolatot távolinak tartja, hogy bármelyik irodai munkavállaló képes legyen ezen eszközök használatára fejlesztőként. Így azzal a szlogennel, ami a Microsoft PowerApps oldalán megtalálható, miszerint: *„Lehetővé teheti minden tag számára, hogy alkalmazásokat készíthessenek olyan fejlett funkciókkal, amelyek korábban csak hivatásos fejlesztők számára voltak elérhetőek”* (Microsoft -A) nem ért egyet. Ezen technológiákon való fejlesztéshez is szükséges egy valamilyen szintű informatikai előismeret, és nyitottság az új dolgok megismerésére.

A low code platformok hátrányainak véli, hogy megfelelően kell felmérni a fejlesztendő projekt volumenét. Fontosnak tartja elkülöníteni, hogy melyik projektekben válassza a cég a hagyományos fejlesztés és technológiák által nyújtott megoldást, és mikor válasszon low code platform adta szolgáltatást. Nem megfelelően felmért projektek esetében létrejöhetnek olyan, kezdetben kicsi volumenűnek indult low code megoldások, amik túlnőnek magukon és a lehetőségeiken. Példája szerint, egy vállalatirányítási rendszert nem kezdene el low code platformon fejleszteni, arra egy hagyományos professzionális megoldás szükséges.

A COVID-19 világjárvány alatt több olyan elkészült low code megoldás segítette a munkavégzés zavartalan működését tapasztalatai szerint a vállalatban, amiket e platformok segítségével nélkül nem lehetett volna ilyen gyorsan megvalósítani.

Szoftverfejlesztési hatékonyságban szerinte szerepe van a low code platformoknak, mert lehetővé teszi a hivatásos fejlesztőknek, hogy a kisebb volumenű projekteket gyorsabban és hatékonyabban végezhessenek el, ezután pedig több idő juthasson a hagyományosan fejlesztett projektekre. Ez pedig napjainkban tapasztalt képzett IT munkaerő hiányában fontos tényező véleménye szerint. Ide kapcsolódik az is, hogy a hagyományos programozás, és a low code fejlesztés nem fogják szerinte kiváltani egymást, nem versengenek egymással, ezek a technológiák egymást kiegészítve képesek hatékonyan működni egy szervezet életében gondolatai szerint.

Véleménye szerint a low code platformok jövője azon dőlhet el, hogy a szervezetek és azok vezetői képesek lesznek-e pontosan megérteni a low code platformok lényegét, ugyanis ha egy szervezet rosszul használja fel ezeket a technológiákat, akkor tévesen rossz eszköznek és technológiának fogja minősíteni ezeket a low code platformokat. Miközben, ha jó helyen jól használják, akkor véleménye szerint sokáig életben tudnak maradni a low code platformok, miközben egyre inkább épülnek majd bele a szervezetek mindennapjaiba. A technológia jövője

szempontjából kiemeltnek tartja a helyes felhasználást segítő kommunikációt, ezen platformokat fejlesztő és árúsító vállalatoktól, mert ezek nagyszerű eszközök, rengeteg mindenre alkalmasak, de nem mindenre, és ez fontos.

**„B” interjúalany** szervezete szintén az elmúlt években váltott a Microsoft Office365 megoldásaira a korábban használt IBM Lotus Symphony-ról. Az Office365 licenc magában hordozta a szervezeténél a PowerApps és PowerAutomate termékeket. Egy külső konzultációs cég hatására pedig a szervezeten belül bevezették a Mendix nevű low code platformot is.

Előnyei közé sorolta, hogy ezeken a platformokon valóban gyors és hatékony a fejlesztés és viszonylag könnyű ezeket a platformokat elsajátítani azoknak, akik alapvetően is IT fejlesztéssel foglalkoznak, vagy rendelkezik informatikai affinitással, ami a fejlesztéshez szükséges. A különböző fejlesztői csoportok összedolgozását szerinte elősegítik ezek a low code platformok, ahol gyorsan és nagy tudásbázist lehet összegyűjteni a különböző technológiájú low code megoldásokról. Azok számára tartja jó eszköznek a low code platformokat, akik szeretnék haladni a korrallal mindig, akik erre nem nyitottak, számukra kevésbé lesz befogadható a low code technológia. Kifejezetten hasznosnak látja azt, hogy úgy lehet fejleszteni egyes megoldásokat, hogy nem szükséges a fejlesztőnek a legmélyebb szintig belemerülnie az alkalmazás fejlesztésébe.

Fontosnak tartja azt, hogy egy csoport tisztában legyen a megfelelő technológia kiválasztásával, kiemelte, hogy ezek a platformok eltérnek, mindegyik másban erős és másra alkalmas, nem lehet minden projektet egyetlen technológia segítségével elkészíteni. Eleinte elmondása szerint, ez a tudásbázis kiépítése lehet nehezebb, ameddig egy vezető, vagy IT szakember annyira képbe kerül a technológiákkal, low code platformokkal, hogy képes legyen az előtte álló projekteken mindig a megfelelő eszközt kiválasztani. Jó technológiáknak tartja a low code platformokat, de egyik platform sem alkalmas mindenre. Szerinte ezek kiismerése leginkább tapasztalat útján lehetséges, ami eleinte időigényes lehet. Elmondása szerint vannak olyan problémák, amikre nem alkalmas egy low code megoldás, ilyen lehet például egyes termékek gyártásához kapcsolódó specifikus folyamatok, amiket nem állna neki low code megoldással megoldani.

Véleménye szerint várható a jelentős terjedése ezeknek a technológiáknak, amik ugyan elmondása szerint még gyerekcipőben járnak és a vállalatok jelenleg a tanulófázisban tartanak ezekben a technológiákban.

## 5.4 Interjúk összegzése

Összegezve a két interjúalanyom gondolatait, meglátásait, kijelenthető, hogy mind „A” mind „B” személy egyetért abban, hogy ezek a platformok hozzájárulnak napjaink digitális fejlődéséhez. Észrevehető, hogy mindkettejük szervezetében a Microsoft licencszerződéssel együtt megjelentek ezek a termékek, így a Microsoft vezető jelenléte elengedhetetlen ezen platformok terjedésében és népszerűségük növekedésében, ahogy ez mind a Gartner (Wong, et al., 2021), mind pedig a Forrester (Bratincevic, Koplowitz, 2021) piackutató cég elemzéseiben is megjelenik.

Hasonlóan gondolkozik mindkettő vezető abban a kérdésben, hogy fontos meghatározni, melyik eszköz mire jó és mire érdemes azt használni, ezért egyértelműen szükséges meghatározni melyik projektet melyik eszköz segítségével célszerű elvégezni. Egyikőjük sem gondolja, hogy ezek a low code platformok versengenek a hagyományos pro code megközelítésű programozással és technológiákkal, sokkal inkább egymást kiegészítve segítik a folyamatos digitális fejlődését a szervezeteknek.

A low code platformoknak mindkettő vezető további piaci és népszerűségbeli növekedést jósol, ugyanis jó eszköznek tartják ezeket a platformokat, ami sok további csoport és szervezet működésére lehet pozitív hatással.

## 6. Eredmények, összefoglalás

A bemutatott nemzetközi és saját esettanulmányokon keresztül láthatjuk, hogy a low code development a jelen korunk egyik ígéretes technológiája, ami a COVID-19 járvány alatt rengeteg szervezetnek segített a gyors és hatékony reagálásban. Mindez adott körülmények között, hagyományos megoldásokkal nem lett volna megoldható, függetlenül az adott szervezet méretétől, berendezkedésétől. Ugyanúgy használták a low code megoldásokat aktívan a több millió lakossal rendelkező városok fejlesztő csoportjai mind a több tízezres multinacionális vállalatok és a pár száz fős kis és középvállalkozások vagy leányvállalatok. Mindezek alapján belátható, hogy a low code development platformok napjainkban minden olyan szervezetnek segítséget nyújtanak, akik a digitalizáció fűtötte korban további fejlődésre, hatékonyságnövelésre törekcsenek. Ezeket megerősítette a két hazai senior IT vezetővel folytatott interjúm eredménye is.

A szoftverfejlesztőhiányt több módon képes enyhíteni az ismertetett technológia, hiszen a fejlesztést bizonyítottan hatékonyabbá és gyorsabbá teszi rengeteg esetben, így a hivatásos fejlesztői erőforrás kihasználása sokkal hatékonyabban kerülhet megszervezésre.

A mélyinterjúm eredménye és a saját véleményem szerint is a hagyományos programozás és a low code fejlesztés nem egymással versengő technológia. Egymást kiegészítve alkotják napjainkra az alkalmazásfejlesztés eszköztárát. A low code technológia a hagyományosan pro code elvű programozást kiegészítve alkalmas arra, hogy a folyamatosan növekvő digitalizációs törekvéseket és növekvő szoftverigényeket kielégíthessék.

A piaci trendeket kutató és előre jelző Gartner (Wong, et al.,2021) és Forrester (Bratincevic, Koplowitz, 2021) is a low code platformok további elterjedését és jelenlétük megszilárdulását jósolja. Ezeket a gondolatokat erősíti meg mindkét mélyinterjúm eredménye, valamint a saját fejlesztői véleményem és tapasztalatom. Ugyanakkor fontos hangsúlyozni, hogy ezen technológiák elterjedésének még mindig a korai fázisában járunk.

Részletes bemutatásra került kettő saját low code megoldásom, ami kettő különböző technológia segítségével különböző vállalati problémákra nyújtott gyors, hatékony és modern megoldást. Ezen alkalmazások, célmegoldások fejlesztéséből szerzett tapasztalatom és véleményem által merem kijelenteni, hogy a low code platformok a jövőben egyre szélesebb körben teszik majd lehetővé az informatikai érzékkel rendelkező munkavállalók között, hogy saját vagy csoportjuk számára digitális megoldásokat készítsenek, miközben technológiák eszköztára folyamatosan fejlődik.

Véleményem és tapasztalatom szerint rengeteg olyan folyamat, régi elavult megoldás újítható meg a low code platformok segítségével, ami gyorsan és drasztikusan változtatja meg a módot, ahogyan dolgozunk. A low code development napjaink és a jövő egy technológiája, ami mellett nem szabad elmenni tétlenül akkor sem, ha nem IT fejlesztéssel foglalkozik valaki, ugyanis nem szükséges a legmélyebb részletéig ismernie a fejlesztés alatt az alkalmazás technikai jellegű működését és felépítését egy low code fejlesztőnek. Olyan megoldások készíthetők el gyorsan és hatékonyan, amire előtte több fős fejlesztői csapatra volt szüksége egy szervezetnek.

Kiemelten fontosnak tartom a low code platformok oktatásának lehetőségét a jövőre nézve. További kutatandó területnek gondolom, miként lehetne hatékonyan akár már az általános iskolai szinttől integrálni az informatikai tananyagba az IT fejlesztést a low code platformokon keresztül. Gyors és látványos vizuális haladás érhető el ezen platformok segítségével, ami sok fiatalat vonzhat be az IT fejlesztés világába később, vagy csak adhat egy általános informatikai



előképzettséget az egyedi megoldások készítéséhez, mindezt izgalmas módon. Piacvezető informatikai vállalatoknak, mint például a Microsoft, ebben külön szerepe lehet, ahogyan összességében a low code platformok elterjedésében is.

Szakedolgozatom forráskutatását érintő problémák közé sorolnám a mélyinterjúm alatt megerősített gondolatot, miszerint a low code platformok nagyon újszerű technológiák, amik szélesebb körben csak az utóbbi pár évben kezdtek elterjedni, ezért arányaiban relatívan kevés hazai szakirodalom érhető el a témában. Ezzel együtt egyelőre külföldi szakirodalom is kevés érhető el a technológia és a téma fiatalságából fakadóan, azonban mivel nagy fejlődés várható ezen platformok előtt, így az idő múlásával nem csak a népszerűségük, hanem az elérhető szakirodalom is bővülni fog véleményem szerint.

## 7. Irodalomjegyzék

Boggan P. (2021): Microsoft Webinar - Develop, Test and Deliver Apps with GitHub Actions for Microsoft Power Platform - <https://www.youtube.com/watch?v=XwR02QvNFdc&t> (letöltve 2021.11.03)

Bojár G. (2018): Negyedik ipari vagy harmadik informatikai forradalom? Az információ sok ezer éves hatalma, Magyar Tudomány 179, 37–46 - [https://mersz.hu/dokumentum/matud\\_96](https://mersz.hu/dokumentum/matud_96) (letöltve 2021.10.20.)

Bratincevic J., Koplowitz R. (2021): Forrester - The Forrester Wave™: Low-Code Development Platforms For Professional Developers, Q2 2021 - <https://powerapps.microsoft.com/en-us/blog/microsoft-is-a-leader-in-the-forrester-wave-low-code-development-platforms-q2-2021/> B link: <https://www.forrester.com/report/The-Forrester-Wave-LowCode-Development-Platforms-For-Professional-Developers-Q2-2021/RES161668> (letöltve 2021.10.30.)

Breaux T., Moritz J. (2021): The 2021 Software Developer Shortage Is Coming - <https://cacm.acm.org/magazines/2021/7/253461-the-2021-software-developer-shortage-is-coming/fulltext> (letöltve 2021.11.02.)

Canorea E. (2021): Plainconcepts - What is Microsoft Power Automate? What about Microsoft Flow? - <https://www.plainconcepts.com/what-is-microsoft-power-automate/> (letöltés 2021.10.27.)

Coca-Cola (2020): Coca-Cola Bottling Company United dispenses streamlined order management with RPA in Microsoft Power Automate - <https://customers.microsoft.com/en-us/story/845187-coca-cola-bottling-company-united-consumer-goods-power-automate> (letöltve 2021.10.20.)

Dr. Géro I., Polacsik G. (2020): Karantén tanulságok: Hogyan alakítja át a felgyorsuló digitális forradalom gazdaságainkat, ISSN: 2063-501X, 3-6 - <http://real.mtak.hu/122033/1/Karanten-tanulsagok-Hogyan-alakitja-at-a-felgyorsulo-digitalis-forradalom-gazdasagainkat.pdf> (letöltve: 2021.11.01)

Európai Bizottság (2019): A digitális korra felkészült Európa - [https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age\\_hu](https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age_hu) (letöltve 2021.11.02.)

Faraj S., Renno W., Bhardwaj A. (2021): Unto the breach: What the COVID-19 pandemic exposes about digitalization - <https://www.sciencedirect.com/sdfe/reader/pii/S1471772721000038/pdf> (letöltve: 2021.11.01.)

FeaturedCustomers: Publikus PowerApps esettanulmányok - <https://www.featuredcustomers.com/vendor/microsoft-powerapps> (letöltve 2021.11.08.)

Ferguson A. (2004): A History of Computer Programming Languages - [https://cs.brown.edu/~adf/programming\\_languages.html](https://cs.brown.edu/~adf/programming_languages.html) (letöltve 2021.10.30.)

Forbes (2020): 14 Benefits Of Low-Code App Development That Tech Pros Love - <https://www.forbes.com/sites/forbestechcouncil/2020/03/24/14-benefits-of-low-code-app-development-that-tech-pros-love/?sh=69f3d2fd48d5> (letöltve: 2021.11.01.)

Forsyth A. -A (2021): Outsystems - Top 5 Benefits of Low-Code: <https://www.outsystems.com/blog/posts/benefits-of-low-code-platforms/> (letöltve: 2021.10.15.)

Forsyth A. -B (2021): Outsystems - Low-Code and No-Code: What's the Difference and When to Use What? - <https://www.outsystems.com/blog/posts/low-code-vs-no-code/> (letöltve 2021.10.15.)

Garms C.: Neptune-software - Low-Code vs Pro-Code: Why They are not Mutually Exclusive - <https://www.neptune-software.com/resource/why-low-code-and-pro-code-are-not-mutually-exclusive/> (letöltve 2021.11.16.)

Habók L. (2018): HWSW - Kapacitásproblémákkal küzdenek a szoftverfejlesztő csapatok - <https://www.hwsw.hu/hirek/58683/codingsans-state-of-software-development-szoftverfejleszttes.html> (letöltve 2021.10.30.)

illimity (2020): illimity Bank simplifies loan process and saves 15 hours a month with Microsoft Power Automate <https://customers.microsoft.com/en-us/story/821782-illimity-bank-banking-power-automate> (letöltve 2021.10.20.)

Jednaszewski M. (2021): Mendix - Low-Code vs. No-Code: Differences, Similarities & Use Cases - <https://www.mendix.com/blog/understand-no-code-vs-low-code-development-tools/> (letöltve 2021.11.02)

Józsa A., (2018): Low code és no code reneszánsza - <https://edit.elte.hu/xmlui/static/pdfjs/web/viewer.html?file=https://edit.elte.hu/xmlui/bitstream/handle/10831/38895/Szakedolgozat.pdf?sequence=1&amp;isAllowed=y> (letöltve 2021.11.10.)

Lowery J. (2018): Gitconnected - Why is Software Development Difficult? - <https://levelup.gitconnected.com/why-is-software-development-difficult-81a2a041d35e> (letöltve 2021.11.11.)

Liu J. (2019): CNBC - The US has nearly 1 million open IT jobs—here's how much it can pay off to switch industries into tech - <https://www.cnbc.com/2019/11/06/how-switching-careers-to-tech-could-solve-the-us-talent-shortage.html> (letöltve 2021.10.30.)

Marvin R. (2014): SD Times - How low-code development seeks to accelerate software delivery - <https://sdtimes.com/application-development/low-code-development-seeks-accelerate-software-delivery/> (letöltve 2021.10.25.)

Martin J. (1991): Rapid Application Development, Macmillan Coll Div kiadó, ISBN 0023-76775-8

Martin J. (2021): BairesDev -Top 12 Software Development Trends for 2021 - <https://www.bairesdev.com/blog/2021-software-development-trends/> (letöltve 2021.11.03.)

- Melnichuk A. (2019): Ncube - Software Engineer Shortage in the World - <https://ncube.com/blog/software-engineer-shortage> (letöltve 2021.10.29)
- Mlitz K. (2021): Statista - Low-code development platform market revenue worldwide from 2018 to 2025 - <https://www.statista.com/statistics/1226179/low-code-development-platform-market-revenue-global/> (letöltve 2021.11.03.)
- Microsoft -A: PowerApps hivatalos dokumentációja - <https://powerapps.microsoft.com/en-us/> (letöltve 2021.10.05.)
- Microsoft -B: PowerBi hivatalos dokumentációja - <https://powerbi.microsoft.com/en-us/what-is-power-bi/> (letöltve 2021.11.01.)
- Microsoft -C: PowerPlatform hivatalos dokumentációja - <https://powerplatform.microsoft.com/hu-hu/> (letöltve 2021.10.05.)
- Microsoft -D: PowerApps alkalmazástípusok hivatalos dokumentációja - <https://powerapps.microsoft.com/hu-hu/build-powerapps/> (letöltve 2021.10.05.)
- Microsoft -E: Dataverse hivatalos dokumentációja - <https://powerplatform.microsoft.com/hu-hu/dataverse/> (letöltve 2021.10.05.)
- Microsoft -F: PowerApps model-driven alkalmazástípus hivatalos dokumentációja - <https://docs.microsoft.com/en-us/powerapps/maker/model-driven-apps/model-driven-app-overview> (letöltve 2021.10.05.)
- Microsoft -G: PowerApps canvas alkalmazástípus hivatalos dokumentációja - <https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/getting-started> (letöltve 2021.10.05.)
- Microsoft -H: PowerFx low code programozási nyelv hivatalos dokumentációja - <https://docs.microsoft.com/en-us/power-platform/power-fx/overview> (letöltve 2021.10.30.)
- Microsoft -I: PowerAutomate hivatalos dokumentációja - <https://powerautomate.microsoft.com/en-us/> (letöltve 2021.10.30.)
- Moon H. (2020): Implementation of a mobile business application built in Microsoft Power Platform - [https://www.theseus.fi/bitstream/handle/10024/339582/HeejinMoon\\_thesis\\_Final.pdf?sequence=2&isAllowed=y](https://www.theseus.fi/bitstream/handle/10024/339582/HeejinMoon_thesis_Final.pdf?sequence=2&isAllowed=y) (letöltve 2021.11.12.)
- Nix: The Ultimate Guide to Rapid Application Development - <https://nix-united.com/blog/the-ultimate-guide-to-rapid-application-development/> (letöltve 2021.11.07.)
- Pegasystems (2021): Embracing low-code development, e-book - [https://www.pega.com/system/files/resources/2021-09/low-code-ebook-v4-en.pdf?\\_rid=YT0xOntzOjc6ImNvbnRfaWQiO3M6OToiQ09OVVC04MDg4Ijt9](https://www.pega.com/system/files/resources/2021-09/low-code-ebook-v4-en.pdf?_rid=YT0xOntzOjc6ImNvbnRfaWQiO3M6OToiQ09OVVC04MDg4Ijt9) (letöltve 2021.11.10.)

- Pham T. (2021): Forbes - Analyzing The Software Engineer Shortage - <https://www.forbes.com/sites/forbestechcouncil/2021/04/13/analyzing-the-software-engineer-shortage/?sh=2b2727e3321c> (letöltve 2021.10.25.)
- Seo H., Sadowski C., Elbaum S., Aftandilian E., Bowdidge R. (2014): Google - Programmers' Build Errors: A Case Study , ACM 978-1-4503-2756-5/14/06 - <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/42184.pdf> (letöltve 2021.11.07.)
- Singh A. (2019): Capterra - What Is Rapid Application Development (RAD)? - <https://blog.capterra.com/what-is-rapid-application-development/> (letöltve 2021.11.07.)
- Sloyan T. (2021): Forbes - Is There A Developer Shortage? Yes, But The Problem Is More Complicated Than It Looks - <https://www.forbes.com/sites/forbestechcouncil/2021/06/08/is-there-a-developer-shortage-yes-but-the-problem-is-more-complicated-than-it-looks/?sh=251f356c3b8e> (letöltve 2021.10.29)
- Schwab, K. (2017): The Fourth Industrial Revolution: Penguin Books kiadó, ISBN: 1524758868
- Tiobe (2021): TIOBE Programming Community Index - <https://www.tiobe.com/tiobe-index/> (letöltve 2021.10.30.)
- Toyota (2020): Toyota improves efficiency and speeds innovation with Microsoft Power Apps - <https://customers.microsoft.com/en-us/story/766054-toyota-manufacturing-power-apps> (letöltve 2021.10.20.)
- Vincent P., Iijima K., Driver M., Wong J., Natis Y. (2019) - Gartner, Magic Quadrant for Enterprise Low-Code Application Platforms, ID G00361584 - <https://www.gartner.com/doc/reprints?id=1-1ODHN7R7&ct=190812&st=sb%201/35> (letöltve 2021.10.10.)
- Wong J., Iijima K., Leow A., Jain A., Vincent P. (2021): Gartner, Magic Quadrant for Enterprise Low-Code Application Platforms - <https://powerapps.microsoft.com/en-us/blog/microsoft-is-a-leader-in-the-2021-gartner-magic-quadrant-for-enterprise-low-code-application-platforms/> (letöltve 2021.11.08.)
- Yoshida T. (2020): City of Kobe - Supporting citizens with Microsoft Power Platform <https://powerapps.microsoft.com/en-us/blog/city-of-kobe/> (letöltve 2021.11.08.)

## 8. Ábrajegyzék

1. ábra, Növekvő igények a szoftverek iránt, fejlesztő hiány (forrás: Boggan, 2021) .....	6
2. ábra, Nyitott pozíciók és várható új munkaerő szektoronként (forrás: Melnichuk, 2019) .....	6
3. ábra, Gartner low code platform elemzése és pozicionálása (forrás: Wong et al., 2021).....	9
4. ábra, Forrester low code platform elemzése és pozicionálása (forrás: Bratincevic, Koplowitz, 2021).....	10
5. ábra, A PowerApps fejlesztői környezetéről és egy CRM alkalmazásomról (saját szerkesztés) .....	14
6. ábra, PowerApps és PowerAutomate integritásáról (saját szerkesztés, forrás:Microsoft-C) .....	15
7. ábra, A PowerAutomate felületéről, egy PowerApps-ba integrált PowerAutomate folyamatómról (saját szerkesztés) .....	17
8. ábra, A Coca-Cola régi ügyfélkezelési folyamatáról (forrás:Coca-Cola, 2020).....	18
9. ábra, A Coca-Cola új, automatizált ügyfélkezelési folyamatáról (forrás:Coca-Cola, 2020).....	18
10. ábra, A PowerApps alkalmazástípusokról (saját szerkesztés, forrás: Microsoft -D).....	20
11. ábra, A RAD módszertan (forrás: Nix) .....	22
12. ábra, A vízésés módszertan (forrás: Nix) .....	22
13. ábra, Google kutatása szerint elkövetett leggyakoribb hibák programozás közben (forrás: Seo, Sadowski, Elbaum, Aftandilian, Bowdidge, 2014) .....	23
14. ábra, Programozási nyelvek népszerűségéről (forrás: Tiobe, 2021).....	24
15. ábra, Low code platformok piacának növekedéséről (forrás: Mlitz, 2021).....	26
16. ábra, GDPR nyilatkozatok régi, hagyományos folyamatáról (saját szerkesztés) .....	28
17. ábra, GDPR nyilatkozatok új, automatizált folyamatáról (saját szerkesztés).....	30
18. ábra, Példa a jóváhagyás felületéről Microsoft Teamsben (saját szerkesztés) .....	31
19. ábra, Példa a jóváhagyás utáni automata e-mailről (saját szerkesztés).....	31
20. ábra, Példa (részlet) a GDPR nyilatkozatok folyamatának követéséről sharepoint listában (saját szerkesztés) .....	32
21. ábra, A PowerAutomate felületéről és részlet az automatizált jóváhagyó folyamat felépítéséről (saját szerkesztés) .....	33
22. ábra, A PowerAutomate felületéről és részlet az automatizált e-mail küldő folyamat felépítéséről (saját szerkesztés) .....	34
23. ábra, Részlet a GDPR folyamat összetettségéről PowerAutomateben (saját szerkesztés) .....	35
24. ábra, A régi, hagyományos IT eszközmenedzsmentről (saját szerkesztés) .....	36
25. ábra, Az új, digitalizált IT eszközmenedzsmentről (saját szerkesztés) .....	38
26. ábra, A PowerApps alapú eszközmenedzsment alkalmazásomról laptopon és telefonon (saját szerkesztés) .....	39
27. ábra, Példa a low code scriptre PowerApps alkalmazásomon belül, automata e-mail küldés adminisztráció után (saját szerkesztés).....	39
28. ábra, A PowerApps fejlesztői felületéről és az alkalmazás munkavállalók számára elérhető részéről, a saját eszközlista oldalról (saját szerkesztés).....	40
29. ábra, A PowerApps alapú eszközmenedzsment alkalmazásom kolléga értesítő felületéről (saját szerkesztés) .....	41
30. ábra, Részlet a munkatársértesítő e-mail PowerAutomate részéről. (saját szerkesztés).....	42

31. ábra, Eszközmenedzsment alkalmazásom eszköz dashboard felülete. (saját szerkesztés).....	43
32. ábra, Példa a low code scriptek lehetőségére az eszköz dashboard felület eszközszámláló funkciójában. (saját szerkesztés).....	43
33. ábra, Részlet a heti eszközösszegző készletinformációs automata e-mailről és a PowerAutomate részéről. (saját szerkesztés) .....	44
34. ábra, Eszközmenedzsment alkalmazásom adminisztrációs felülete. (saját szerkesztés) .....	45
35. ábra, Policy tracker PowerApps alkalmazásom mobilos felülete. (saját szerkesztés).....	46
36. ábra, CRM PowerApps alkalmazásom felületei, különböző platformokon. (saját szerkesztés).....	46

## 9. Mellékletek

### Mélyinterjú fő kérdései:

1. Jelenleg milyen pozícióban dolgozik, a munkáltató szervezete mely iparágba sorolható?
2. Eddigi szakmai pályafutása során milyen méretű, nemzetiségű, cégeknél dolgozott, milyen pozícióban, ezek a szervezetek mely iparághoz tartoztak?
3. Eddigi munkái során milyen programnyelvek kerültek előtérbe?
4. Ön eddigi munkái során fejlesztett Low Code Development platformokban? Ön eddigi munkái során felügyelte-e már Low Code Development platformokban történő fejlesztést? Hogyan találkozott az LCDP-vel? Mi volt az első benyomás?
5. Ön szerint a low code platformoknak milyen előnyeik vannak?
6. Ön szerint a low code platformoknak milyen hátrányaik vannak?
7. Ön szerint a low code development platformok segítették-e a jelenleg munkáltató szervezetének digitalizációját és a COVID-19 világjárvány alatti reagálást, amennyiben igen, hogyan?
8. Ön szerint a megnövekedett szervezeti digitalizáció korában hogyan tud segíteni a Low Code fejlesztés?
9. Ön szerint mi az amiben nem tudja helyettesíteni egy Low Code megoldás (PowerApps/PowerAutomate) a hagyományos programozási nyelv, környezet által fejlesztett megoldást?
10. Ön szerint a low code platformok milyen segítséget nyújtanak a jelenleg tapasztalható szoftverfejlesztő hiányban?
11. Ön szerint a hagyományos programozás és a Low Code fejlesztés miként egészíthetik ki egymást?
12. Ön szerint a low code platformoknak milyen jövő várható?

