

Egyedi vállalatra szabott bérléskezelő rendszer fejlesztése

Székely Dániel

Mérnökinformatikus MSc

2025

Tartalomjegyzék

1. Bevezetés	1
2. Projekt áttekintése és részletes bemutatása	3
2.1. A vállalat és az üzleti igények	3
2.2. A projekt célkitűzései	4
2.3. A projekt összetettsége és kihívásai	4
2.4. Technológiai háttér	4
3. Projekt élelciklusa	6
3.1. Projektindítás (Initiation)	7
3.2. Tervezés (Planning)	7
3.3. Megvalósítás (Execution)	7
3.4. Ellenőrzés és irányítás (Monitoring és Controlling)	8
3.5. Projekt lezárás (Closure)	8
4. Kockázatkezelés és problémamegoldás	9
4.1. Felmerült problémák	9
4.2. Kockázatok azonosítása és priorizálása	9
4.3. Megoldási stratégiák	10
4.4. Tanulságok a kockázatkezelésből	10
5. Tanulságok és szakmai összegzés	11
5.1. Projektmenedzsment tapasztalatok	11
5.2. Fejlesztési és technológiai tanulságok	11
5.3. Összegzés: a gyakorlati tanulságok	12
Ábrák jegyzéke	13
Irodalomjegyzék	13

1. fejezet

Bevezetés

A mai vállalati világban az idő és az erőforrások hatékony kezelése nem csupán optimális, hanem létfontosságú versenyelőny. A hagyományos, papíralapú vagy szigetszerűen kezelt folyamatok lassúak, átláthatatlanok és hibákra hajlamosak. A vállalatoknak ezért exponenciálisan igénye van olyan rendszerekre, amelyek gyorsítják a munkafolyamatokat, csökkentik a hibalehetőségeket és automatizálják a mindennapi működést.

Ebben a környezetben vállaltam egy önálló projektet a vállalatnak, ahol már évek óta tevékenykedem. Egy **egyedi bérletskezelő rendszer (Rental Management System – RMS)** fejlesztését, amely a vállalat minden projektéhez kapcsolódó folyamatait digitális formába önti. A cél nem csupán a szoftver létrehozása volt, hanem annak teljes életciklusú menedzselése, a tervezéstől és ütemezéstől kezdve a kockázatok azonosításán át a megvalósításig és az átadásig és karbantartásig.

A dolgozat célja a projekt részletes bemutatása: ismertetem az RMS céljait, funkcióit és technológiai hátterét. Szakmai szempontból külön hangsúlyt kapnak a projektmenedzsment folyamatok, különösen a **tervezés**, az **ütemezés** és a **kockázatkezelés**, valamint annak bemutatása, hogy ezek miként járultak hozzá a projekt sikeres megvalósításához.

A dolgozat többek között a következő kérdésekre is választ keres:

- Hogyan tervezhető és menedzselhető hatékonyan egy egyedi vállalati szoftverfejlesztési projekt?
- Milyen módszerek és eszközök biztosítják az erőforrások optimális felhasználását és a kockázatok minimalizálását?
- Milyen szakmai tanulságok vonhatók le az önálló projektmenedzsment gyakorlatából, amelyek más projekteken is alkalmazhatók?

A következő fejezetben bemutatom a projekt konkrét célkitűzéseit és főbb jellemzőit, kiemelve, hogy a fejlesztett rendszer milyen problémákat old meg és milyen értéket teremt a vállalat számára.

2. fejezet

Projekt áttekintése és részletes bemutatása

2.1. A vállalat és az üzleti igények

A projekt célja a **TéDé Rendezvények** vállalat bérlet és projektkezelési folyamatainak teljes digitalizálása volt. A korábbi papíralapú és Excel-alapú folyamatok nem feleltek meg a vállalat növekvő igényeinek, és egy modern, integrált rendszer kialakítása vált szükségessé. Az online elérhető üzenetküldő és felhő alapú megoldások nem kínáltak megfelelő rugalmasságot és testreszabhatóságot, ezért egy egyedi fejlesztésű rendszer mellett született döntés.

A vállalat számára kiemelten fontos volt egy egységes, modern és felhasználóbarát rendszer, amely:

- automatizálja a bérleti folyamatokat,
- nyomon követi a készleteket,
- biztosítja az adminisztráció teljes körű kezelését,
- egyszerűsíti a munkafolyamatokat,
- csökkenti a hibalehetőségeket,
- javítja a kommunikációt a csapaton belül,
- egységesíti a dokumentációt,
- minden információ egy helyen elérhető,
- lehetővé teszi a gyors árajánlatkészítést.

2.2. A projekt célkitűzései

További célok voltak:

- modern, felhasználóbarát felület
- egyszerűen telepíthető és skálázható **Docker** segítségével
- könnyen karbantartható és bővíthető architektúra
- biztonságos hozzáférés-kezelés és jogosultságok
- részletes riportálási és statisztikai funkciók

2.3. A projekt összetettsége és kihívásai

A projekt felettébb összetettnek tekinthető, mivel:

- többféle felhasználói szerepkört kellett kezelnie (admin, munkatárs, menedzser),
- integrálni kellett különböző adatforrásokat és a bérlési folyamatokat,
- biztonsági és hozzáférés-kezelési követelményeknek kellett megfelelnie,
- a fejlesztés során több technológiát kellett összehangolni a rugalmasság és megbízhatóság érdekében.

2.4. Technológiai háttér

A rendszer fejlesztése a következő technológiákra épült:

- **Backend:** PHP a **Twig** sablonmotorral,
- **Frontend:** modern responsive felhasználói felület Twig sablonokkal,
- **Konténerizálás és telepítés:** Docker és Docker Compose, előre elkészített konténerekben,
- **Konfiguráció:** testreszabható `.env` fájlok segítségével.
- **Reverse proxy:** Nginx a kérések kezelésére és a statikus fájlok kiszolgálására,
- **Adatbázis:** MySQL a megbízható adatkezelés érdekében,

- **Verziókezelés:** Git a kód nyomon követésére és együttműködésre.

Ez a technológiai kombináció biztosítja a rendszer gyors telepítését, stabil működését és könnyű bővíthetőségét.

3. fejezet

Projekt életciklusa

A projektmenedzsment egyik legfontosabb alapelve, hogy minden projektnek megvan a maga életciklusa, amely jól elkülöníthető fázisokra bontható [2, 1]. Ezek a fázisok nemcsak logikai, hanem szervezeti és irányítási szempontból is meghatározóak, hiszen lehetővé teszik a projekt strukturált tervezését, nyomon követését és értékelését [2].

A hazai szakirodalom egyaránt öt alapvető fázist különít el [1, 2]:

1. **Projektindítás (Initiation)** a projekt céljainak, indokoltságának és alapvető paramétereinek meghatározása.
2. **Tervezés (Planning)** az ütemezés, erőforrások, kockázatok és feladatok részletes kidolgozása.
3. **Megvalósítás (Execution)** a tényleges fejlesztési és kivitelezési folyamatok végrehajtása.
4. **Ellenőrzés és irányítás (Monitoring & Controlling)** a projekt előrehaladásának, költségeinek és minőségének nyomon követése [2].
5. **Lezárás (Closure)** a projekt hivatalos befejezése, átadás és értékelés [1].

A projektciklus fázisai egymásra épülnek, ugyanakkor gyakran átfedésben is lehetnek: az ellenőrzési és irányítási folyamat például a megvalósítás teljes ideje alatt folyamatosan zajlik [2]. A modern megközelítések különösen az iteratív és ismétlődő fejlesztési modellek nem mindig lineáris struktúrát követnek, hanem sprint vagy iteráció formájában valósítják meg a fejlesztést [1].

Ennek ellenére a klasszikus projektciklus továbbra is nélkülözhetetlen a stratégiai és vállalati projektekben, mivel jól követhető keretet biztosít az egész folyamat számára [2].

A **TeDeRMS** fejlesztésénél egy klasszikus, ötfázisú modell szolgált alapul, melyet a projekt egyedi körülményeihez önálló fejlesztés, korlátozott erőforrások, vállalati integráció igazítottam. Az alábbiakban részletesen bemutatom, hogyan valósult meg a projekt életciklusa a gyakorlatban.

3.1. Projektindítás (Initiation)

A projektindítás során a következő lépések történtek:

- **Igényfelmérés:** a vállalat bérléskezelési folyamatait elemeztem, hogy pontosan fel-térképezsem a fejlesztendő rendszer funkcióit és a problémás területeket.
- **Célok meghatározása:** világos, mérhető célkitűzéseket rögzítettem, például az ad-minisztrációs idő csökkentését, a hibák minimalizálását és az automatizált riportok bevezetését.
- **Erőforrás-tervezés előkészítése:** meghatároztam a projekthez szükséges techno-lógiai és időbeli erőforrásokat. Mivel a fejlesztést önállóan végeztem, kiemelten fontos volt a prioritások és a munkaidő hatékony beosztása.

Ez a fázis biztosította, hogy a projekt kiindulópontja egyértelmű legyen, és a fej-lesztés a vállalat igényeinek megfelelően induljon el. Amennyiben a projektindítás nem lett volna alapos, a későbbi fázisokban jelentős problémák merülhettek volna fel, például félreértett követelmények esetében a fejlesztés nem a kívánt irányba haladt volna ezzel erőforrásokat és időt pazarolva.

3.2. Tervezés (Planning)

A tervezési fázisban a projekt sikerének záloga a részletes ütemezés és a kockázatok előrejelzése volt:

- **Ütemterv készítése:** a projekt főbb mérföldköveit és feladatait időrendi sorrendbe állítottam, így követhetővé vált a fejlesztés előrehaladása.
- **Erőforrás-tervezés:** részletesen meghatároztam az idő- és technológiai erőforrásokat, valamint a napi munkabeosztást, hogy az önálló fejlesztés gördülékeny legyen.
- **Kockázatelemzés:** azonosítottam a legfontosabb kockázatokat (pl. technikai hibák, adatvesztés, hibás üzleti logika), és kidolgoztam a megelőző és elhárító intézkedéseket.

A tervezés során alkalmazott módszerek: feladatlista, ütemezett mérföldkövek, koc-kázatmátrix és rendszeres önellenőrzés.

3.3. Megvalósítás (Execution)

A megvalósítás során a projekt tényleges fejlesztési munkája zajlott:

- **Fejlesztési folyamatok:** moduláris felépítésben, backend és frontend párhuzamos fejlesztése, verziókövetés GitHub-on.
- **Tesztelés:** folyamatos unit és funkcionális tesztek, hogy a rendszer stabil és hibamentes legyen.
- **Dokumentáció:** a kód és a rendszer működésének részletes dokumentálása, hogy a későbbi karbantartás és bővítés egyszerű legyen.

Ez a fázis biztosította, hogy a rendszer minden funkciója a terv szerint valósuljon meg, és a vállalat igényei teljesüljenek.

3.4. Ellenőrzés és irányítás (Monitoring és Controlling)

A projekt előrehaladásának nyomon követése kritikus volt a siker szempontjából:

- **Haladás nyomon követése:** a mérföldkövek teljesülésének ellenőrzése, eltérések azonosítása és korrekciója.
- **Kockázatok kezelése:** a kockázatmátrix folyamatos frissítése, a problémák gyors azonosítása és megoldása.
- **Minőségellenőrzés:** a rendszer funkcionalitásának és stabilitásának folyamatos ellenőrzése a hibák minimalizálása érdekében.

Ez a fázis biztosította, hogy a projekt ne csússzon ki a tervezett keretek közül, és minden mérföldkő a kívánt minőségben valósuljon meg.

3.5. Projekt lezárás (Closure)

A projekt lezárásakor a következő tevékenységek történtek:

- **Rendszer átadása:** a TeDeRMS teljes körű telepítése a vállalat környezetében.
- **Dokumentáció:** részletes használati útmutatók és adminisztrátori kézikönyvek készítése.
- **Oktatás:** a kulcsfelhasználók képzése a rendszer hatékony használatára.
- **Tanulságok összegzése:** a projekt során szerzett tapasztalatok dokumentálása, javaslatok a jövőbeli bővítésekhez.

Ez a fázis biztosította, hogy a rendszer hosszú távon stabilan és hatékonyan működjön, miközben a projektmenedzsment folyamatok tanulságai később is felhasználhatók.

4. fejezet

Kockázatkezelés és problémamegoldás

A **TeDeRMS** projekt során a kockázatkezelés kulcsfontosságú szerepet játszott, mivel a projekt önálló fejlesztésként valósult meg, így minden döntés és probléma gyors és hatékony kezelést igényelt. A kockázatkezelés célja a potenciális problémák előrejelzése, azok hatásának minimalizálása, valamint a projekt sikerének biztosítása volt.

4.1. Felmerült problémák

A fejlesztés során több jelentősebb kihívás merült fel:

- **Technológiai problémák:** a rendszer stabilitása és a Docker-konténerek konfigurációja kezdetben nem volt optimális, ami futtatási hibákat okozott.
- **Funkcionális kihívások:** a bérléskezelési folyamatok pontos leképezése a szoftverben nehézkes volt, különösen a különböző státuszok és jogosultságok kezelése.
- **Időmenedzsment:** mivel a projektet önállóan végeztem, a napi munka és a projekt előrehaladása közötti egyensúly fenntartása komoly kihívást jelentett.
- **Tesztelés és hibajavítás:** a rendszer moduláris felépítése miatt a hibák lokalizálása és javítása több iterációt igényelt.

4.2. Kockázatok azonosítása és priorizálása

A projekt elején egy **kockázاتمátrixot** készítettem, amely segített azonosítani a legkritikusabb problémákat:

- **Magas prioritású kockázatok:** kritikus hibák a backend működésében, adatvesztés, biztonsági hiányosságok.

- **Közepes prioritású kockázatok:** kisebb felhasználói felületbeli problémák, riportálási hibák.
- **Alacsony prioritású kockázatok:** kisebb esztétikai hibák vagy későbbi bővítési igények.

A priorizálás lehetővé tette, hogy a projekt során először a legkritikusabb problémákra koncentráljak, így a rendszer stabilitása és funkcionalitása biztosított volt.

4.3. Megoldási stratégiák

A felmerült problémákra a következő stratégiákat alkalmaztam:

- **Technológiai problémák kezelése:** a Docker-konfiguráció és a PHP/Twig környezet iteratív finomhangolása, rendszeres teszteléssel.
- **Funkcionális kihívások kezelése:** a bérléskezelési modulok folyamatos tesztelése és a logika iteratív javítása a hibák kiküszöbölésére.
- **Időmenedzsment:** napi és heti ütemtervek készítése, a feladatok priorizálása a projekt előrehaladása érdekében.
- **Tesztelés és hibajavítás:** moduláris tesztelési stratégia alkalmazása, ahol minden modul külön ellenőrzésen esett át, majd a teljes rendszer integrációját is teszteltem.

4.4. Tanulságok a kockázatkezelésből

A projekt során szerzett tapasztalatok alapján a kockázatkezelés kulcsfontosságú az önálló fejlesztésekben:

- A problémák előrejelzése és a priorizálás jelentősen növeli a projekt sikerességét.
- A rendszeres tesztelés és dokumentáció minimalizálja a hibák előfordulását és gyorsítja a hibajavítást.
- Az időmenedzsment és a feladatpriorizálás elengedhetetlen a projekt gördülékeny előrehaladásához.

Összességében a tudatos kockázatkezelés és a gyors problémamegoldás jelentősen hozzájárult a **TeDeRMS** sikeres elkészítéséhez és stabil működéséhez.

5. fejezet

Tanulságok és szakmai összegzés

A **TeDeRMS** projekt nem csupán egy vállalati bérleskezelő rendszer létrehozását jelentette, hanem egy teljes önálló projektmenedzsment gyakorlatot is. A fejlesztés során szerzett tapasztalatok világosan megmutatták, hogy a siker kulcsa a tudatos tervezés, a kockázatok proaktív kezelése és a moduláris, skálázható fejlesztési megközelítés.

5.1. Projektmenedzsment tapasztalatok

Tervezés, ütemezés és fókusz: A részletes ütemterv és mérföldkövek alkalmazása biztosította, hogy minden kritikus feladat időben elkészüljön. Az önálló munkavégzés során a prioritások helyes meghatározása kulcsfontosságú volt: a fókusz a rendszer stabil és hibamentes működésén maradt.

Proaktív kockázatkezelés: A kockázattárix és a folyamatos kockázatfigyelés lehetővé tette, hogy a problémákat még a bekövetkezésük előtt azonosítsam és minimalizáljam. Ez a stratégia elengedhetetlen volt a projekt gördülékeny előrehaladásához.

Önálló erőforrás-menedzsment: Az önálló projekt során az idő és a technológiai erőforrások optimalizálása kritikus volt. A napi és heti ütemezés, valamint a feladatprioritások betartása biztosította a folyamatos előrehaladást.

5.2. Fejlesztési és technológiai tanulságok

Moduláris, skálázható architektúra: A rendszer moduláris felépítése lehetővé tette a könnyű karbantartást, gyors hibajavítást és a jövőbeni bővítéseket. Ez a rugalmasság a vállalat hosszú távú igényeit is kiszolgálja.

Folyamatos tesztelés és dokumentáció: A moduláris tesztelés és a részletes dokumentáció biztosította, hogy a rendszer stabilan működjön, a hibák gyorsan javíthatók legyenek, és a felhasználók könnyen eligazodjanak a rendszerben.

Technológiai integráció: A PHP/Twig backend és a Docker alapú konténerizálás kombinációja biztosította a rendszer könnyű telepíthetőségét, stabilitását és rugalmasságát, így a vállalat gyorsan és biztonságosan tudta használni az új rendszert.

5.3. Összegzés: a gyakorlati tanulságok

A **TeDeRMS** projekt világosan bizonyította, hogy egy önállóan megvalósított szoftverfejlesztési projekt is lehet strukturált és szakmailag magas színvonalú, ha:

- tudatosan tervezünk,
- proaktívan kezeljük a kockázatokat,
- modulárisan és skálázhatóan fejlesztünk,
- valamint folyamatosan dokumentálunk és tesztelünk.

A projekt során szerzett tapasztalatok értékesek nemcsak a jelenlegi fejlesztéshez, hanem a jövőbeli projektekhez is. Különösen a gyakorlati projektmenedzsment, önálló fejlesztés és kockázatkezelés területén adnak mérhető és átvihető szakmai tudást.

Ábrák jegyzéke

Irodalomjegyzék

- [1] Szalay Imre: *Projektmenedzsment alapok: módszertan és gyakorlat*. Budapest, 2018, Budapesti Műszaki és Gazdaságtudományi Egyetem. ISBN 9789633133377.
- [2] Hajdu Miklós: *Projektmenedzsment*. Budapest, 2014, Akadémiai Kiadó. ISBN 9789634540181.