

**BUDAPESTI GAZDASÁGI EGYETEM  
GAZDÁLKODÁSI KAR ZALAEGERSZEG**

# **Szabadságnyilvántartó rendszer tervezése Agilis módszerrel és UX design felhasználásával**

Belső konzulens: Dr. Gubán Miklós

Külső konzulens: Ferecskó-Kereszturi Beáta

Kocsis Gábor

Nappali

Gazdaságinformatikus

Logisztikai informatikus

**2020**

**NYILATKOZAT**

a szakdolgozat digitális formátumának benyújtásáról

Hallgató neve:	<b>Kocsis Gábor</b>		
Szak:	<b>Gazdaságinformatikus</b>		
Szakirány (ha van):	<b>Logisztikai informatikus</b>		
Neptun kód:	<b>UOOJU1</b>	Benyújtás éve:	<b>2019.</b>
Szakdolgozat/záró-dolgozat címe:	<b>Szabadságnylvántartó rendszer tervezése Agilis módszerrel és UX design felhasználásával</b>		
Belső (operatív) konzulens neve:	<b>Dr. Gubán Miklós</b>		
Külső (szakmai) konzulens neve:	<b>Ferecskó-Kereszturi Beáta</b>		
Legalább 5 kulcsszó a dolgozat tartalmára vonatkozóan:	<b>agilis, Scrum, UX design, usability, tervezés</b>		

Kérjük a szerzői döntésnek megfelelő opciót aláhúzni:

Hozzájárulok / **nem járulok hozzá**, hogy szakdolgozatomat az egyetem az interneten a nyilvánosság számára repozitóriumában közzétegye.

A hozzájárulás szerzői feltételei:

- mások számára a közzététel semmilyen formában nem engedélyezett,
- a dolgozat magáncélra letölthető, a forrás és nevem megjelölésével szabadon idézhető, de az idézést meghaladó felhasználás (átvétel) tilos,
- hozzájárulásom időtartamra nem korlátozott és bármikor visszavonható.

Hozzájárulás hiányában a dolgozat csak az egyetem könyvtáraiban az arra kijelölt számítógépeken, képernyős megtekintéssel kutatható. Egyéb hozzáférés, többszörözés nem engedélyezett.

Felelősségem tudatában kijelentem, hogy szakdolgozatom digitális adatállománya mindenben eleget tesz a vonatkozó hatályos intézményi előírásoknak, tartalma megegyezik nyomtatott formában benyújtott szakdolgozatommal.

Dátum: 2019. december 16.

  
.....  
hallgató aláírása

**A digitális szakdolgozat könyvtári benyújtását és átvételét igazolom.**

Dátum: 2019. DEK. 16

  
.....  
könyvtári munkatárs

## Tartalomjegyzék

1.	Bevezetés .....	2
2.	Előzmények, jogszabályi háttér .....	4
2.1.	Előzmények, követelmények .....	4
2.2.	Jogszabályi háttér .....	6
3.	Az Agilis szoftverfejlesztésről röviden.....	9
3.1.	Scrum, mint agilis keretrendszer.....	13
4.	A tervezés folyamata, lépései .....	20
4.1.	A felhasználók megismerése.....	21
4.2.	A felhasználók útja – user journey.....	25
5.	Prototípus tervezés és megvalósítás.....	28
5.1.	A prototípus megvalósítása.....	28
5.2.	A prototípus bemutatása.....	33
6.	A prototípus tesztelése, eredmények értékelése .....	40
6.1.	A usability teszt.....	40
6.2.	Tesztelési észrevételek, javaslatok.....	46
7.	Összefoglalás .....	49
8.	Irodalomjegyzék .....	52
9.	Ábrajegyzék .....	54

## 1. Bevezetés

Szakedolgozatom témájának kiválasztásakor fontos szempont volt, hogy aktuális, esetlegesen a jövőbe mutató téma legyen. Ezenfelül arra törekedtem, hogy kapcsolódjon felsőoktatási tanulmányaimhoz, illetve a duális gyakorlati helyemül szolgáló Zalaszám Informatika Kft. (későbbiekben Zalaszám) profiljához illeszkedjen, mely egy több, mint 40 éves tapasztalattal rendelkező zalaegerszegi székhelyű informatikai vállalkozás. A Zalaszám célja, hogy az informatikai termék-előállítás és szolgáltatások piacán és az információbiztonság területén minden partnere számára folyamatosan kiemelkedő és egyenletes minőséget, magas megbízhatóságot biztosítson. A cég mottója: "A mai szolgáltatásokkal a holnapit kell megalapozni."<sup>1</sup>

Ezek tudatában is esett a választás egy teljesen új rendszer tervezésére a papírra rajzolt „skicctől” a kattintható prototípusig, melyet folyamatos revízió és átdolgozás követhet az agilis szoftverfejlesztés és UX Design<sup>2</sup> jegyében. Hiszen a folyamatosan fejlődő világunk a 21. század ezen szakaszára odáig jutott, hogy ez a két terület kéz a kézben jár, helyenként össze is mosódik.

Egyetemi éveim alatt több gazdasági és informatikai területtel találkoztam, ismerkedtem meg. Témámhoz főként az Információs rendszer fejlesztés és a Projektmenedzsment tárgy kapcsolódik, de fellelhető benne az egyetemi évek alatt megismert és autodidakta módon bővített HTML alapú programozás, illetve egy kevés humánmenedzsment is. Továbbá segítségül voltak a Zalaszámnál tartott különböző oktatások az Agilis fejlesztésről – azon belül is a Scrumról – illetve a UX design képzések.

A forrásaim nagy részét az elektronikusan fellelhető külföldi és magyar szakirodalmak, jogszabályok, cikkek teszik ki. De Pásztor Dávid: UX Design és Mike Cohn: Agile Estimating and Planning című könyve is meghatározó volt, mind a felhasználói felület tervezésekor, mind a Scrum módszertan elsajátítása során.

Véleményem szerint az Agilis szoftverfejlesztés és a UX Design gyerekcipőben jár Magyarországon, mi sem bizonyítja ezt jobban, minthogy kevésnek mondható a fellelhető magyar nyelvű szakirodalom. Szerencsére az elmúlt években az irodalom jelentősen növekedett, Pásztor Dávid könyve a UX Designról áttörésnek számított a területen. Az,

---

<sup>1</sup> Zalaszám Informatika Kft. – Rólunk

<sup>2</sup> User Experience Design vagyis felhasználói élmény tervezés.

hogy angol nyelvű forrásokat kellett használnom szakdolgozatom elkészítése során nem nehezítette meg a feladatot, sőt hálás dolog bővíteni a nyelvtudást. Az úgynevezett „kellestem a hasznossal” mondás teljes mértékben igaz, hiszen úgy gondolom nyelvtudás 2019-ben – nem csak – az IT területén kiemelten fontos.

A dolgozat során arra a kérdésre keresem a választ, hogy mire van szüksége a Zalaszámos dolgozóknak egy szabadságnylvántartó rendszerben a saját és vezetői elképzeléseken kívül, figyelve arra is, hogy ne legyen specifikus, így a későbbiek során is értékesíthető legyen a vállalati szférában.

Ezenfelül megfogalmaztam egy hipotézist, mely szerint az új kollégák nem feltétlenül ismerik a szabadságolás menetét a Zalaszámnál, melyre véleményem szerint a későbbi interjúk során kapok választ.

Diplomamunkám elkészítése során azért is esett a Scrum módszertanra a választás, mivel úgy gondolom, hogy hiába adottak a vezetői elvárások a rendszerrel kapcsolatban, mindenképp meg kell felelni az alkalmazottak igényeinek is, amelyek folyamatosan időről-időre változhatnak. Ezért nem volt célszerű a vízésés modellt alkalmazni, hiszen még a saját elképzeléseim is változtak napról napra.

Kiemelendő a duális gyakorlati helyemül szolgáló Zalaszám Informatika Kft. támogatása is. Nagy segítség volt a konzulensemtől, mentoromtól, kollégáktól, illetve a fejlesztő csapat tagjaitól kapott három és fél éves önzetlen segédkezés.

## 2. Előzmények, jogszabályi háttér

### 2.1. Előzmények, követelmények

Napjainkban a technológia folyamatosan fejlődik, mindig keresi az újításokat. Elvárás, hogy meg kell felelni a modern kor vívmányainak. Ezért is merült fel a Zalaszámnál először ötletként, később igényként egy szabadságnylvántartó rendszer webes felülete és ehhez kapcsolódóan mobiltelefonos applikáció létrehozása. A fő cél a jelenleg Excelben több kolléga által párhuzamosan – különböző céllal – vezetett nyilvántartások kiváltása. Az informatikai rendszerben támogatni kell a jelenlegi folyamatot, helyenként automatizálva, kiküszöbölve a felhasználói beavatkozást.

Egy új rendszer esetén mindig meg kell felelni az elvárásoknak, ebben az esetben sincs ez másképp. Ezeknek az elvárásoknak egy része jogszabályi eredetű. Az alkalmazásnak az alapvető tevékenységek biztosítása mellett (mint például szabadság igénylés, jóváhagyás a munkahelyi vezető által, engedélyezés az adott igazgató, ügyvezető igazgató által) több más felhasználói igényt kielégítő funkciót is zökkenőmentesen garantálnia kell:

- Az alkalmazásban a már meglévő adatok alapján a munkavállalónak az adott évben kiadható szabadságainak megjelenítése, az aktuális állapot (mennyi kivehető/kiadható napja van) lekérdezési lehetősége a dolgozó és a vezetők által.
- A munkanap áthelyezéseket automatikusan vegye figyelembe a rendszer a szabadságok kezelésénél, lehetőség legyen adott évre előre felrögzíteni a rendszerben ezeket a napokat, ünnepnapokkal együtt és akár megjeleníteni külön színnel, hogy a felhasználó számára szembetűnő legyen.
- A munkavállaló kivenni tervezett szabadságnapjainak rögzítése úgy, hogy figyelembe vegye az előző pontban leírtakat, ezáltal könnyítve a tervezést.
- Folyamatos integráció a vállalatirányítási rendszerrel, a Zalasám esetében az OrganE-vel, ezzel is elősegítve a szabadságok figyelembevételét a havi elszámoláshoz, bérszámfejtéshez.
- A rendszerben kezelni kell a szülési- és betegszabadság kérdését is.
- A felhasználók jogosultságokkal szabályozva meg tudják nézni, hogy adott napon mely kolléga van szabadságon, melyik munkaterületet, projektet érinti a távollét

és meddig tart. Ez a kapacitás becslések és más alkalmazottak szabadságolása miatt elengedhetetlen.

- Szintén jogosultság kezeléssel kell megvalósítani, hogy a munkavállaló a saját munkaügyi adatait érhesse el, azokat és a családtagjai adatait szerkeszteni tudja, esetleg hozzáadni a meglévőkhöz.
- A mobiltelefonos alkalmazásnak is ugyanazt az adatbázist kell használnia, mint a webesnek, ezáltal lehet biztosítani az adatok konzisztens kezelését.
- Kötelező szabadság elrendelésének lehetősége, mivel általában a munkanap áthelyezésekkor – ügyvezetői utasítás alapján – szabadságot kell igénybe venni.
- A munkaviszony megszűnésekor a megmaradt szabadságnapok elszámolása, melyet a törvény szabályoz.

## 2.2. Jogszabályi háttér

Mint általában minden szoftvernek, rendszernek meg kell felelnie a jogszabályoknak, így a szabadságnylvántartónak is, főként a Munka Törvénykönyvének, illetve a hazai és Európai Uniós adatvédelmi törvénynek (GDPR).

A szabadságok kiadását, nyilvántartását a munka törvénykönyvéről szóló 2012. évi I. törvény (továbbiakban Munka Törvénykönyve) szabályozza. A jogszabály értelmében a munkavállalónak minden naptári évben szabadság jár, amely alap- és pótszabadságból áll. Az alapszabadság mennyisége 20 nap, ha a dolgozó év közben kezdi meg a munkáját, akkor ezt időarányosan kell meghatározni.

A pótszabadságoknak több fajtája van, amelyeket a Munka Törvénykönyve XI. fejezetének 56. bekezdése taglal.

Életkor szerint az alábbi pótszabadságok járnak, ez először akkor jelenik meg, amikor a munkavállaló betöltötte a szükséges életévet. [1]

Életkor	Pótszabadság
a 25. életévtől	1 munkanap
a 28. életévtől	2 munkanap
a 31. életévtől	3 munkanap
a 33. életévtől	4 munkanap
a 35. életévtől	5 munkanap
a 37. életévtől	6 munkanap
a 39. életévtől	7 munkanap
a 41. életévtől	8 munkanap
a 43. életévtől	9 munkanap
a 45. életévtől	10 munkanap

1. ábra: Életkor alapján járó pótszabadság

Saját szerkesztés a Munka Törvénykönyve 117. § alapján

URL: <https://net.jogtar.hu/jogszabaly?docid=A1200001.TV> Letöltés időpontja: 2019.09.25.

A munkavállaló amennyiben 16 évnél fiatalabb gyermeke van szintén pótszabadságra jogosult. Egy gyermek esetén 2, kettő esetén 4, három vagy annál több esetén 7 munkanap pótszabadság jár. Amennyiben az az eset áll fent, hogy a gyermek fogyatékos, gyermekenként 2 nappal több jár. [1]



*„Az apának gyermeke születése esetén, legkésőbb a születést követő második hónap végéig öt, ikergyermek születése esetén hét munkanap pótszabadság jár, amelyet kérésének megfelelő időpontban kell kiadni.”* 2012. évi I. törvény XI. fejezet 59. rész 118.§ (4).

Vannak olyan esetek is, amelyek közvetlenül nem érintik a Zalaszámot, de a későbbi értékesíthetőség szempontjából fontos, hogy a rendszer kezelni tudja ezeket a pótszabadságra jogosító kitételeket. A 18 év alatti munkavállalóknak öt nap szabadság jár, utoljára akkor, amikor betöltik a 18. életévüket. Ezenfelül a föld alatt nap, mint nap dolgozók, vagy az állandó ionizáló sugárzásnak<sup>3</sup> kitett munkahelyen dolgozók évente 5 nappal többet tölthetnek a munkától távol. Továbbá, ha a rehabilitációs szakértői szerv megállapít egy legalább 50 százalékos mértékű egészségkárosodást szintén 5 nap pótszabadságra jogosult a munkavállaló.[1]

A Munka Törvénykönyve kitér a betegszabadság kérdésére is. Ha a munkavállaló betegség miatt keresőképtelenné válik, akkor évente 15 nap szabadságra jogosult. Ugyanaz a szabály érvényes itt is, mint az alapszabadság esetében, ha év közben kezdődik a munkaviszony, akkor időarányosan kell biztosítani.[1]

Másik fontos a vállalatok nagy részét érintő helyzetben meghatározásra került a Munka Törvénykönyve 62. részében a szülési és fizetés nélküli szabadság. A törvény úgy határoz, hogy az édesanya 24 hét szabadságra jogosult, amiből kettőt köteles kivenni, továbbá, ha egyben szeretné igénybe venni, akkor 4 hétnek a szülés várható időpontja elé kell esnie. A fizetés nélküli szabadság hasonlóan az alapszabadsághoz minden munkavállalónak jár, viszont nincs limitálva a mennyisége napokra lebontva, de legfeljebb 2 év lehet és 15 nappal az igénybevétel előtt jelezni kell. Ezt több okból is lehet kérni a munkáltatótól.[1]

A törvény a szabadságok kiadásáról is rendelkezik, főszabályként az esedékesség évében kell kiadni, de ha év végén a munkavállaló megkezd, akkor a következő évbe „átnyúlhat” 5 munkanap erejéig, így az az esedékesség évében kivett szabadságnak minősül.[1]

---

<sup>3</sup> Ionizáló sugárzás: Ionizáló sugárzás az olyan sugárzás, amely héjelektron-reakciók révén az elnyelő anyag atomjait gerjesztik, vagy ionpárokká változtatják.”

Forrás: BME Nukleáris Technikai intézet

Letöltés időpontja: 2019.10.21.

Hozzáférés (URL): [http://oftankonyv.reak.bme.hu/tiki-index.php?page=Az+ionizáló+sugárzás+biológiai+hatása\\_ptg](http://oftankonyv.reak.bme.hu/tiki-index.php?page=Az+ionizáló+sugárzás+biológiai+hatása_ptg)

A munkáltatónak lehetőséget kell biztosítania a munkavállaló számára, hogy egy naptári évben legalább 14 egybefüggő napra mentesüljön a munkavégzési kötelezettsége alól. Természetesen ez akkor érvényes, ha nincs ettől eltérő megegyezés a munkaszerződésben. [1]

A Munka Törvénykönyve szerint a munkaviszony megszűnéskor, ha a munkáltató az arányos szabadságot nem adta ki, azt meg kell váltani. Ha a munkavállaló az arányosnál több napot vett ki, vissza kell fizetnie a távolléti díj összegét a munkáltató számára. [1]

Ennek tükrében tehát a programnak biztosítania kell a funkciót: az arányosan járó, de ki nem adott szabadságnapokra, valamint az arányosnál több kiadott napra járó távolléti díj összegét ki kell számolnia.

### 3. Az Agilis szoftverfejlesztésről röviden

*„Egy jó terv azonnal végrehajtva jobb,  
mint egy tökéletes terv jövő héten végrehajtva.”*

George S. Patton

Az egyetemi és duális gyakorlati éveim alatt sokféle ismeretanyaggal ismerkedtem meg, ezek közül az Agilis szoftverfejlesztés és projektmenedzsment fogott meg legjobban, azon belül is a Scrum módszertan.

Az agilis szoftverfejlesztés az iteratív fejlesztési módszerek csoportjába tartozik, amelyet 2001-ben az Agile Manifesto (magyarul Agilis Kiáltvány) nevű publikációban adtak ki. A 17 szoftverfejlesztő szerző által kiadott dokumentum elsősorban egyfajta értékrendszer, követendő útmutatás. A kiáltvány röviden és tömören megfogalmazza a legfontosabb értékeket:

- Az egyéniségek és interakciók fontosabbak a folyamatoknál és az eszközöknél
- A teljes körű dokumentáció helyett működő szoftver
- Együttműködés a megrendelővel, ügyféllel szemben a szerződéses tárgyalásokkal
- Folyamatosan alkalmazkodni a változásokhoz, azzal szemben, hogy az eredeti tervet követjük [3][5][10]

A folyamatos alkalmazkodást és gyors cselekvőképességet már Patton tábornok is felismerte, ahogy a kezdő idézetben is fogalmazott, többek között ennek is köszönhető sikerességét a II. Világháború során. Az agilis értékrendszer lényege, hogy gyorsak, változásokra készek legyünk, hiszen bármelyik pillanatban változhat az ügyfél igénye. [6]

A 2001-ben kiadott kiáltványban Kent Beck 12 alapszabálynak nevezhető támpontot fogalmazott meg fejlesztőtársaival.

- A legfontosabb az ügyfél kielégítése használható szoftver gyors és folyamatos átadásával.
- A követelmények kései megváltozása sem okozhat problémát, hiszen az agilis fejlesztők felkészültek a változásra.
- A működő szoftver / prototípus átadása rendszeresen, lehetőség szerint egy-két hetente.
- Napi szintű kapcsolattartás a megrendelő és a fejlesztők között.

- Motivált emberek köré kell építeni a projektet, meg kell adni számukra a szükséges eszközöket és támogatást, ezenfelül bizalmat a munkájuk iránt.
- A leghatékonyabb módszer az információ cseréjére, áramlására a fejlesztő csapaton belül a szemtől-szembeni megbeszélés.
- A működő szoftver az előrehaladás legfőbb mértékegysége.
- Az agilis folyamatok hozzájárulnak a fenntartható fejlesztéshez.
- Folyamatos érdeklődés a technikai kitűnőségnek, illetve egy jó dizájnnal kihangsúlyozhatjuk az agilitást.
- Az egyszerűség elengedhetetlen.
- A legjobb tervek, követelmények és a dizájn önszerveződő csapatoktól származik.
- A csapatok rendszeres időközönként reagálnak a változásokra, hogy még hatékonyabbak legyenek. Ezt követően összehangolják a változást a csapat működésével. [5][9]

Egy új termék fejlesztésének becslése és tervezése ijesztő feladat, főleg akkor, ha a tévhiteinkkel, félreértelmezésekkel nehezítjük. Az agilis szemlélet során az értékrendbeli hangsúlyok nagymértékben megváltoztak a korábban megszokott vízésés modellhez képest. Sok agilis fejlesztési módszer létezik, melyek elősegítik a fejlesztést, ösztönzik a csapatmunkát, az egymás és ügyfélközti együttműködést a projekt életciklusán keresztül. Ezen szemlélet törekszik egy feladatot több kisebb részre bontani, ezáltal egy-egy fejlesztési ciklus 2-4 hétig tart. Ezt a ciklust, melyben hétről-hétre valami újat adnak hozzá a projekthez „idődoboznak” nevezik. Lehetőség nyílik arra, hogy hamar visszajelzést kapjunk a termékről, hiszen iterációról iterációra bemutatásra kerülhet a jelenlegi és a leendő ügyfelek számára is. Ennek köszönhetően nincs olyan jellegű tervezés és dokumentáció, mint például a vízésés modell esetében, mindig csak adott ciklusra vonatkozóan. Természetesen nem jelenti azt, hogy nem kell dokumentálni, csak a hangsúly nem azon van. A kiindulási alapja ennek az elvnek az, hogy a projekt indulásakor nem lehet előre meghatározni egy szoftver minden részletre kiterjedő rendszertervét, hiszen változhatnak a technológiai körülmények, változhat az ügyféligény vagy akár a csapat összetétele is. [2][5][10]

Ahogy már említettem korábban, az agilis lényegi eleme, hogy önszervezőek a fejlesztő csoportok. Az agilis csapatok egymásba vetett bizalma és egyéni képességeik képesek

előre vinni az adott projektet, hiszen tudják, hogy egy jól együttműködő csapat könnyedén túlszárnyal egy hibásan vagy elavult módszer alapján dolgozó közösséget. Nagy általánosságban véve a fejlesztő team nem egy konkrét feladatra specializálódott, hanem összetétele többféle területről származik. A legtöbb esetben tartalmaz backend, frontend vagy akár full-stack fejlesztőt, rendszerszervezőt, tesztelőt (akár átfedés is lehet), illetve egy személyt, aki az egész csapatot összefogja, a Product Owner (Projektgazda). A frontend fejlesztő a felhasználói felülettel foglalkozik, azzal, amivel a felhasználó először találkozik. A backend fejlesztő ennek a háttéréért, adatbázis struktúráért és szerver oldalért felelős. A full-stack fejlesztő, aki mindezeket együttesen képes kezelni. Ideális esetben a team 5-9 főből áll és célszerű, ha egy légtérben, irodában tartózkodnak. Ez természetesen egyenes arányosságban áll azzal, hogy az egymás közti kommunikáció felgyorsul, az információ gyorsabban áramlik, mindazonáltal elősegíti a csapaton belüli együttműködést is. [3][5][10][17]

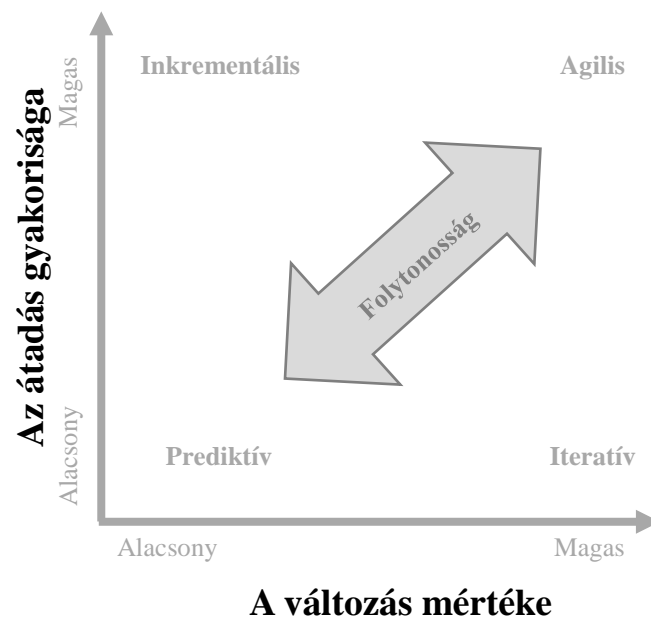
Az agilis csapatok általában háromszintű tervezést alkalmaznak: projekt életciklus tervezés, iteráció tervezés, napi tervezés. Az első esetben a tervezés az átadásig tervez, projekttől függően változó időtartammal, amely lehet 3-6-9-12 hónap is. Az iterációra vonatkozó elképzelések a korábban már említett 2-4 hetes ciklust ölelik fel. A napi tervezés a csapattagok adott napra vállalt kötelezettségei, amelyeket a „daily stand-up” során mondanak egymásnak. [10]

Az időtervezés is relatív, hiszen van az ideális idő, ami alatt el kellene készülnie a projektnek, illetve az eltelt, valós idő, ami alatt ténylegesen elkészült a program. Mike Cohn egy remek példán keresztül mutatja be a kettő közti különbséget. Egy szimpla kérdés: Hivatalosan milyen hosszú egy amerikai futballmeccs? Aki már hallott róla, az egyből azt mondja, hogy 60 perc. Ideális esetben ennyi is lenne, de ez sosincs így, hiszen általában 3-4 órán keresztül tart a mérkőzés mire az óra „eléri” a 60 percet. A különbség oka egyszerű, a közbejövő tényezők, játékmegszakítások. Ugyanez igaz a projektekre is, hiszen bármikor közbejöhet valami, ami miatt nő a szükséges idő. [10]

Az „idődoboz” vagy iteráció hasonló egy életciklushoz. Ugyanúgy tartalmazza a tervezést, követelményspecifikációt, a konkrét programozást és a tesztelést. Tehát lényegében tartalmazza a vízésés modell során alkalmazott tevékenységeket, mégis más és hatékonyabb fejlesztési módszertan.

A hatékonyság főként akkor mutatkozik meg igazán, ha egy teljesen új szoftvert, terméket állítunk elő. Ebben az esetben nincs semmiféle „kapaszkodó” csak a csapat addig meglévő tudása és tapasztalata. Jellemző az ilyen projektekre az is, hogy nem feltétlenül látjuk az elkészítendő termék körvonalait. Nincs a birtokunkban az az információ, amellyel megfelelő specifikációt tudnánk a szoftverhez készíteni. Továbbá nem biztos, hogy a team azonnal tud megoldást találni a problémára, mert elég gyakran az ügyfél sem tudja teljes mértékben, mit szeretne. [3]

A projektek általában kétféle új ismeretanyagot adnak. Az egyiket a termékről, a másikat a projektről. Mindkettő hasznos, hiszen az agilis módszerek az adaptálható technikákat helyezik előtérbe, mellyel könnyen lehet reagálni a változásokra. [3]



2. ábra: A projekt életciklus és folytonosság kapcsolata  
Saját szerkesztés az Agile Practice Guide (p. 19.) alapján [13]

Az ábrából is látszik, hogy ha a csapat agilis szemléletmódot használ gyakrabban és fokozatosan adja át az elkészült terméket, ezáltal korábban kap visszajelzést az ügyfél véleményéről és elképzeléseiről. Ez üzleti szempontból is fontos lehet, hiszen a meglévő és új információk alapján könnyebben tud tervezni, vagy akár egy befektetési megtérülést (ROI – Return of Investment) számolni.

Ahhoz, hogy gyakran átadható legyen a termék, szükség van az ügyfél bevonására a fejlesztésbe, sőt a tervezési fázisba is. Mindenképp kell egy személy, aki képviseli az ügyféligényeket. Legfontosabb feladata, hogy a tervezés és fejlesztés során felmerülő

kérdésekre a lehető leghamarabb választ adjon, illetve az új igényeket jelezze a team felé, hogy minél gyorsabban reagálni tudjanak rá és az alapján tervezzenek. [5]

### **3.1. Scrum, mint agilis keretrendszer**

A Scrum egy agilis szoftverfejlesztési keretrendszer, de előszeretettel használják projektmenedzselésre és humánmenedzsment területen is. Egy keretrendszer melyen belül komplex problémákat oldanak meg adaptív módon, miközben produktívan és kreatívan „szállítják” az adott terméket a lehető legmagasabb szinten. Ez egy „könnyűsúlyú”, könnyen érthető iránymutatás. Ken Schwaber és Jeff Sutherland szerint meglehetősen nehéz „mesteri szinten” alkalmazni. [8]

Már a 90-es években felmerült Ken Schwaberben egy scrumszerű módszer, de akkor még nem volt kiforrott. Majd később Jeff Sutherlanddal karöltve elkezdtek kidolgozni a keretrendszer alapelveit saját tapasztalataik alapján. Sokan azonosítják a rögbiben használt Scrumra, hiszen sok a közös a kettőben. A legfontosabb mindkettőben a csapatmunka és egymás támogatása. A keretrendszer „ereje” is ebben rejlik, a kisméretű csapatok rugalmasságában és alkalmazkodóképességében. A komplex technológiai, piaci és környezeti hatások ugrásszerű növekedésével és változásával a Scrum könnyedén megbirkózik. [8]

A Scrum az agilis módszertan elemeit használja, maga a keretrendszer csapatokból ahhoz rendelt szerepekből, eseményekből és szabályokból áll. Ezen kívül a folyamatok ellenőrzését tapasztalatok alapján végzik, vagyis empirikus módszerrel. Az empirizmus szerint a tudást kétféleképpen szerezhetjük meg: tapasztalatokból, illetve az elsajátított ismereteken alapuló döntéseinkből. [8]

A tapasztalat alapú folyamatrevízió megvalósítása három összetevőn alapul:

- áttekinthetőség
  - Az eredményért felelős személyek számára láthatóvá kell tenni a folyamat lényeges nézőpontjait. Ez az elv megköveteli, hogy egy közös szabvány szerint dolgozzanak, amely minden résztvevő számára ugyanazzal a jelentéssel bír. A Zalaszám esetében ez az ISO-t jelenti.

- revízió
  - A résztvevőknek gyakran ellenőrizni kell a haladást, annak érdekében, hogy a hibákat ki lehessen küszöbölni. Mindazonáltal az ellenőrzés gyakorisága nem mehet a munka rovására.
- adaptáció
  - Amennyiben megállapításra kerül, hogy a projekt/fejlesztés nem megfelelő ütemben halad, módosítani kell a folyamatot annak érdekében, hogy a hibák és az akadályozó tényezők kiküszöbölésre kerüljenek.[8]

Ahogy az Agilis kiáltványban is meghatározásra kerültek szerepkörök, a Scrumban is megjelennek, csak lényegesebben lebontva minden – adott projektet érintő – területre koncentrálnak.

A Scrum önszerveződő csapata három fő „elemből” áll: a Product Owner (Terméktulajdonos), Development Team (Fejlesztő csapat) és a Scrum Master („Scrum Mester”). Az önszerveződésnek köszönhetően saját maguk döntenek el, hogyan tudják a legjobban elvégezni a feladatokat, azzal szemben, hogy valamelyik vezető beosztású irányítónak ők. Ezek a csapatok az agilis módszertannak megfelelően rendszeresen és fokozatosan „szállítják” az elkészült terméket az ügyfél részére. [8]

A Terméktulajdonos felelős azért, hogy a termék értéke a lehető legmagasabb legyen, a Fejlesztőcsapat munkája a tudásukhoz mérten a maximális legyen. Az, hogy egy adott csapatban ki a Terméktulajdonos sok mindentől függ. Projektenként és szervezetenként is eltérő lehet, de az egyének meglévő tudása és tapasztalata is befolyásoló tényező. A Terméktulajdonos határozza meg a Product Backlog-ot (Termék teendőlista), illetve az abban foglalt feladatok egyértelmű leírását. Ezen felül feladata annak biztosítása, hogy a Backlog minden csapattag számára könnyen hozzáférhető, egyértelmű és egyszerű legyen, hiszen ez alapján zajlik a tervezés és fejlesztés. Ezeket a feladatokat természetesen végezheti a fejlesztőcsapattal karöltve, vagy akár nekik is kioszthatja, de a felelősség minden esetben az övé lesz. [8]

A Fejlesztőcsapat az Agilis módszertannak megfelelően 5-9 szakemberből áll. Ez azért is fontos, mivel ha ennél kisebb létszámú a csapat nem feltétlenül tud nagy mennyiségű munkát végezni. Ezenfelül akár kompetencia korlátokba is ütközhetnek, aminek eredményeképp nem sikerül az iteráció végére átadható verziót készíteni a termékből. Amennyiben nagyobb méretű a csapat, megnehezedhet a kommunikáció, esetleg külön



modulon dolgozó tagok zavarhatják egymást a mindennapos kommunikáció során. Természetesen ezek nincsenek köbe vésvé, nem igaz minden Scrumot alkalmazó vállalatra, a megfigyelések vezettek ezekre a megállapításokra. A tagok ezért is azon dolgoznak, hogy minden egyes fejlesztési, tervezési ciklus (Sprint) végére átadható legyen a termék (szoftver) adott időszakra vállalt fejlesztése, hiszen az eredményesség mértéke az átadott termékmennyiség nagysága adott ciklusra vetítve. [8]

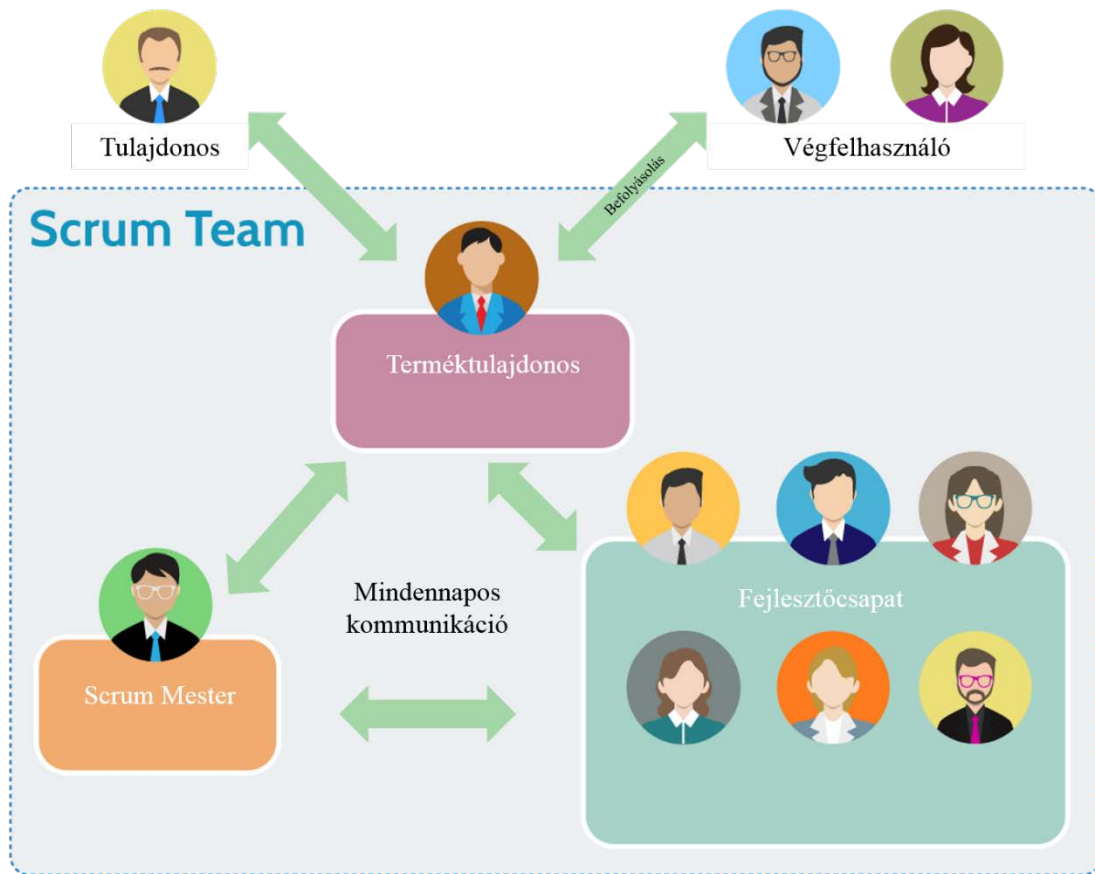
A Fejlesztőcsapatban nincs hierarchia, csak munkakör alapú különbségek lehetnek. Természetesen egy szoftverfejlesztés esetében elkülönülhet a szervező, tesztelő, fejlesztő feladatkör, de ugyanaz a felelőssége mindenkinek, hiszen a felelősség az egész csapatra hárul. Általában a csapatok úgy állnak össze, hogy minden olyan kompetenciával rendelkeznek, amely szükséges és elégséges feltétele a termék lehető legmagasabb szintű előállításának. Magától értetődő, hogy egy fejlesztési projekt kezdetekor egy szervező/tesztelő beállítottóságú csapattag nem fog programot írni, de amíg nincs mit tesztelni, addig sem tétlenkedik, hanem proaktívan készül a következő iterációra, vagy a jelenlegihez készít tesztelési tervet. [8]

A Scrum Mester felelős azért, hogy maga a módszertan és keretrendszer működjön. Segít a csapattagoknak megérteni mi is az a Scrum, mind elméletben, mind gyakorlatban egyaránt. Segítséget, támogatást nyújt:

- azoknak, akik nem tartoznak közvetlenül a csapatba (pl. a cég más alkalmazottjai) megérteni a Scrum lényegét, miért hasznos, hogy fog ebből a cég profitálni
- a Scrum eseményeinek megszervezésében és lebonyolításában
- a Teamnek a lehető legmagasabb színvonalú termék létrehozásában
- a módszerek után kutatásban, melyekkel még hatékonyabban kezelhető a Backlog
- mindenki számára elsajátítani az agilitást.

Összefoglalóan ő a projektmenedzser szerepét betöltő személy.

A tulajdonos vagy menedzser feladata a fejlesztéshez szükséges megfelelő környezet kialakítása és biztosítása annak érdekében, hogy minden a lehető legjobban működjön a projekten belül.



3. ábra: A Scrum Csapat

Saját szerkesztés a Visual Paradigm What is Scrum Team ábrája alapján.

Letöltés időpontja: 2019. október 21.

URL: <https://www.visual-paradigm.com/scrum/what-is-scrum-team/>

A Scrum két szerepkörre választja a módszertant. A szerepkörökhöz tartozik egy tréfás megközelítés is, amelynek alapja egy vicc:

A csirke és a disznó sétálnak az utcán. Egyszer csak a csirke megszólal:

- „Te, nyissunk egy éttermet!”
- Erre a disznó: „Jó ötlet, mi legyen a neve?”

A csirke gondolkozik, majd rávágja:

- „Legyen Sonkástojás!”

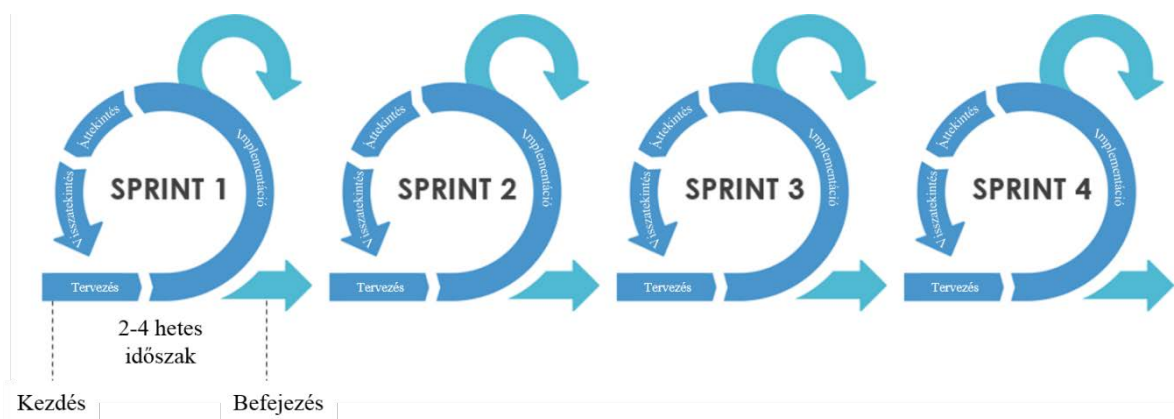
A disznó erre:

- „Nem tetszik, mert én biztosan mindent beleadnék, te meg éppen csak hogy részt vennél benne.” [5]

A vicc Scrumban történő interpretációja az, hogy a disznók azok, akik mindent megtesznek azért, hogy a projekt sikeres legyen. A csirkék pedig azok, akik szintén érdekeltek az adott projektben, de ők annak a haszonélvezői és a felelősség sem feltétlenül

az övük, ha nem sikeres a projekt. Jelen esetben a „disznók” a Fejlesztői Csapat tagjai, a csirkék pedig a menedzsment és az üzleti szereplők. [5]

Korábban már említettem az agilis szoftverfejlesztés kapcsán az iterációkra bontott fejlesztési ciklust. Ezt a Scrumban Sprintnek hívják. Ken Schwaberek szerint ez a keretrendszer „szíve”, hiszen ez alatt a 2-4 hetes ciklus alatt zajlik a fejlesztés és tervezés, egy új inkrementum létrehozása. A Sprints egymást követik, mindegyik Sprint Planning-gel (Sprint tervezés) kezdődik és Sprint Retrospective-vel zárul. A Sprinteket addig kell ismételni, amíg a Product Backlogból el nem fogynak a feladatok, teendők. Ezenfelül a Sprint része az elvégzett munka és az eredményül kapott szoftver is. [5][8]



4. ábra: Sprints a Scrumban

Saját szerkesztés Visual Paradigm Sprints in Scrum ábrája alapján Letöltés dátuma: 2019. október 25.

URL: <https://www.visual-paradigm.com/scrum/why-fixed-length-of-sprints-in-scrum/>

A szakirodalom a korábban említett két történésen kívül három másik eseményt tárgyal:

- Backlog Grooming
- Sprint Review
- Daily Meeting

A nemrég leírt Sprint Planning keretein belül beszél meg a csapat, hogy az ügyféligényeket sorrendbe rendezve mit tudnak vállalni a következő ciklusra. Figyelembe veszik, hogy menetközben mennyire és miben változott az ügyféligény és aszerint „húzzák be” a Sprint Backlogba a feladatot. Ha már megvolt az első sprint egy projektnél, a következő tervezés során azt is meg kell beszélni, hogy sikerült-e a vállalt feladatokkal végezni vagy túlvállalták magukat a Fejlesztőcsapat tagjai. Értelemszerűen, ha „alulvállalta” magát a csapat, akkor érdemes a következő Sprintre a becslések alapján több feladatot vállalni. Ez a felülvizsgálat a Sprint Planning keretein belül is lehet vagy egy külön eseményben a Backlog Grooming (teendőlista finomítás) során. Gyakorlatban

ez a kettő összemosódhat, ám ilyenkor hosszabb idejű megbeszélésre kell számítani. [5][8]

A Sprint Review Meeting (Sprint áttekintés) hasonlít a Sprint Planningre, sőt gyakorlatban meg is egyezhet, mivel ezen esemény során áttekinti a csapat az előző Sprint feladatait. Annyi a lényeges különbség a Planninghez képest, hogy a Review-n részt vehet a megrendelő is, ezáltal visszacsatolást kaphat a team azonnal, közvetlenül az ügyféltől. [5][8]

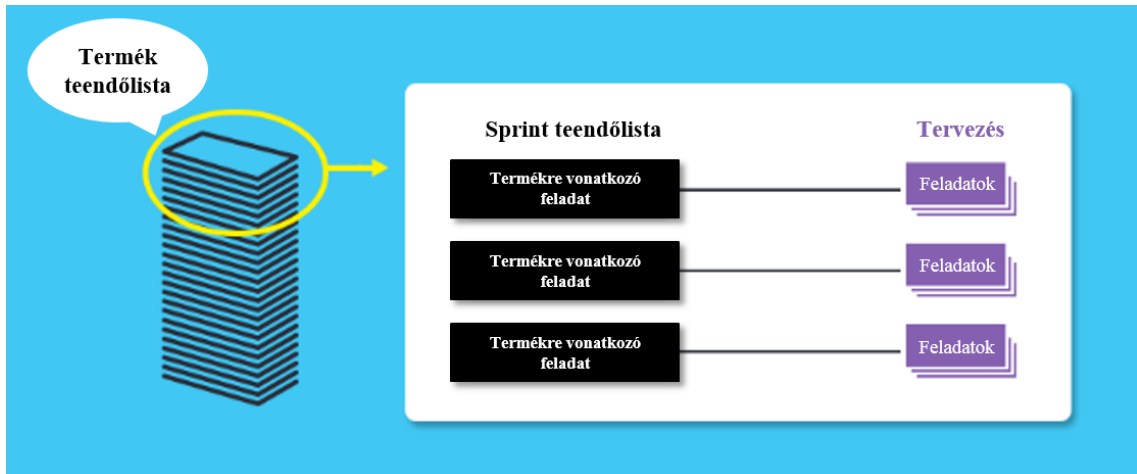
A Retrospective (visszatekintés) egy mélyebb szintű áttekintés, itt nem konkrétan a feladatokat nézi át a csapat, hanem azt, hogy az előző Sprint során milyen problémák hátráltatták a fejlesztőcsapatot a munkában, illetve mik voltak a Sprint során a jól működő elemek, amiket a későbbiek során is meg kell tartani. Természetesen a probléma előjöhethet a napi szintű megbeszélésen is, de ott nincs idő annak orvoslására. A Retrospective lehetőséget nyújt a csapattagoknak arra, hogy előadják észrevételeiket, melyekkel elháríthatják az esetleges akadályokat, optimalizálhatják a fejlesztést. Itt csapat szinten történik a „továbbfejlesztés” és az optimalizálás. [5][8]

A Daily Meeting vagy Daily Stand up napi rendszerességgel megtartott megbeszélés, ugyanabban az időpontban, ugyanazon helyen, ezáltal is növelve az egymásközi kommunikációt és bizalmat. Maximum 15 percig tarthat, ahol minden csapattag egyesével elmondja mit csinált az előző nap, illetve mit fog csinálni azon a napon. Ezenfelül beszámolnak arról is, hogy ütköztek-e akadályokba a feladat elvégzése során. Ahogy a nevéből is adódik, hogy a megbeszélés állva zajlik, nem szabad leülni, ezáltal is elősegítve, hogy ne lépjenek túl a 15 perces időkorlátot. [5][8]

Az eseményeken kívül még részletezésre kerültek az úgynevezett Scrum munkaanyagok is, amely a korábban már említett Product és Sprint Backlog. Mindkettő egy lista az elvégzendő feladatokról, sorba rendezve prioritás szerint, annyi különbséggel, hogy a Product Backlog a teljes fejlesztésre vonatkozik, a Sprint csak az adott iterációra. A termék teendőlista nem gondolható teljesnek és véglegesnek az első meghatározást követően, hiszen csak körülbelül ismerjük a követelményeket, melyek természetesen állandóan változnak. Ahogy Hérakleitosz is fogalmaz „Semmi sem állandó, csak a változás maga.” [8]

A Product Backlog permanensen és dinamikusan változik azért, hogy meg lehessen határozni mi kell ahhoz, hogy a szoftver vagy termék hasznos és versenyképes legyen a piacon. Egy-egy Sprint lezárása után – amikor átadásra kerül a termék – a felhasználó használni kezdi azt, egyből új feladatokat generál, hiszen az észrevételeket át kell vezetni

a rendszerben. A feladatlista felülvizsgálata és finomítása – a szintén korábban említett – Backlog Grooming, melynek folyamatait és tevékenységét a Scrum csapat dönti el, de fontos kihangsúlyozni, hogy ez a felülvizsgálat nem köthet le a team kapacitásából 10 százaléknál többet. Természetesen a feladatok között kialakítandó hierarchia során az a mérvadó, hogy mely tétel van jobban részletezve, mely hangsúlyosabb. [8]



5. ábra: A Product Backlog és a Sprint Backlog kapcsolata

*Forrás: Saját szerkesztés a QuickScrum ábrája alapján Letöltés dátuma: 2019. október 26.*

*URL: <https://www.quickscrum.com/Article/ArticleDetails/4047/4/Product-Backlog-Items>*

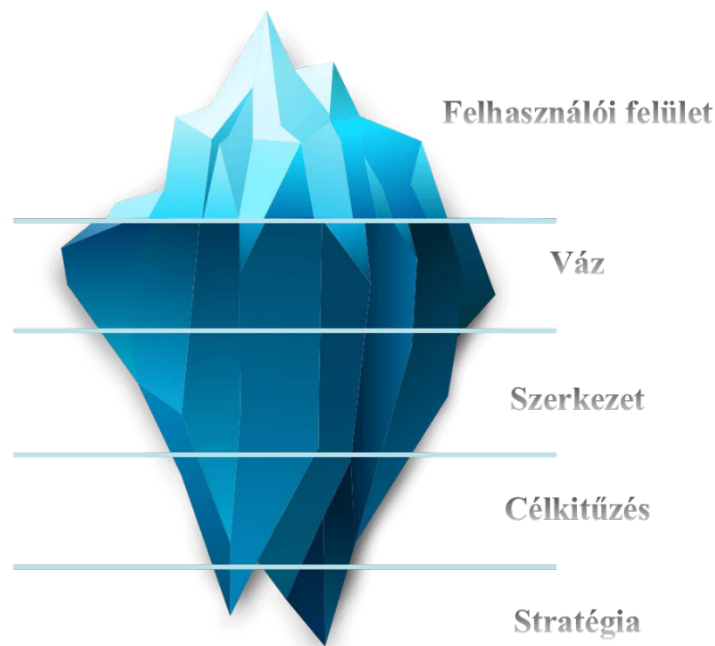
A Sprint Backlog azon elemek halmaza, melyek a Product Backlogból az adott iterációra kiválasztásra kerültek, ezt szemlélteti az ábra is. A Sprint teendőlista érzékelhetővé teszi azt a munkát, melyet a team elengedhetetlennek tart elvégezni a ciklus során. A munka előrehaladtával a ráfordítás becslés is változik, hiszen lesznek olyan feladatok, amelyeknek már korábban „kitaposták” az ösvényt, így ami azelőtt nehéznek bizonyult később könnyen megvalósítható feladat lesz. A munkaanyagoknak minden esetben átláthatónak kell lennie, hogy mindenki számára könnyen érthető legyen, hiszen így optimalizálható a fejlesztés és növelhető sprintről sprintre az átadott termékmennyiség. [8]

A különböző Scrumon belüli tevékenységeken és munkaanyagokon kívül fontos meghatározni, hogy a csapat számára mi jelenti azt a fogalmat, hogy „kész” (Definition of Done, DoD). Egy projekt elején nehéz erre érdemi választ adni. Azért is fontos, mivel a Sprint Planningen a kész feladatok alapján tervezzük a következő iterációt. Természetesen, ahogy a csapat halad előre az időben tapasztaltabb lesz és a szerzett tapasztalatok alapján folyamatosan fejlődik a „kész” fogalma is. [8]

## 4. A tervezés folyamata, lépései

A tervezés során funkcionalitás oldalról közelítettem meg a rendszert, melynek során felhasználtam a cégnél elsajátított UX Design képzés elemeit, illetve ötvöztem az Agilis módszertannal. A fúzió egyszerű volt, hiszen a UX Design is az agilitásra épül.

De mi is az a User Experience Design vagyis felhasználói élmény tervezés? Amikor 10-ből 9 ember először meghallja egyből egy jól kinéző, divatos felületre gondol, pedig a valóság teljesen más.



6. ábra: A UX Design "jéghegye"

Saját szerkesztés a UX Planet ábrája alapján

Kép forrása: [www.pngfly.com](http://www.pngfly.com) Letöltés dátuma: 2019. október 28.

Ahogy az ábrából is látszik a felhasználó csak a jéghegy csúcsát látja, mögötte viszont rengeteg teendő van. A legfontosabb, hogy helyezzük középpontba a felhasználót, hiszen neki készítjük a szoftvert. Meg kell tudni, hogy milyen problémái vannak, így az alkalmazással tudunk segíteni nekik megoldani azt. [11]

## 4.1. A felhasználók megismerése

Az UX ezt a fajta megismerést a personák (személyiség vagy karakter) feltérképezésének nevezi. Ennek során interjúkat készítettem különböző beosztású kollégákkal, annak érdekében, hogy mire van szükségük egy szabadság nyilvántartó szoftver kapcsán.

Ezeket az interjúkat Pásztor Dávid UX Design könyve, illetve az oktatás során kapott instrukciók alapján készítettem. Az egyik legfontosabb szabály, hogy nyílt kérdésekkel érdeklődjünk, hiszen hagyni kell az interjúalanyt kibontakozni, ezáltal több információt megtudunk a vágyairól, elvárásairól. Fontos, hogy megtaláljuk a jelenlegi folyamat során a „fájdalompontokat”, melyeket a rendszerrel szeretnénk kiküszöbölni. [11]

A különböző beosztású embereket különböző szemszögek szerint kérdeztem, de a lényeg minden esetben az volt, hogy mit szeretnének látni. Első körben pár személyesebb jellegű kérdést tettem fel, mint például „Mesélj magadról, mi jellemző rád?”, hiszen fontos, hogy megismerjük a felhasználó személyiségét, ne csak a rendszerrel szemben támasztott elképzeléseit. A továbbiakban a korábban említett nyílt kérdéseket alkalmaztam, illetve munkavállaló-felettes kettősre bontottam, hogy mindkét aspektusból megfelelő képet kapjak. Az interjú során az alábbi kérdéseket alkalmaztam:

- Munkavállaló esetén:
  - Hogy látod a folyamatot a szabadságigénylés során?
  - Mi viszi el a legtöbb időd?
  - Mi van abban az esetben, ha sürgősen szabadságra kell menned és otthonról kellene azt igényelned?
  - Legutóbb, amikor több projektben dolgoztál hogyan történt a szabadság igénylés? Okozott-e ez problémát?
  - Mennyire tudod előre tervezni a szabadságaidat? Látsz ebben problémás területet?
  - A szabadságigénylési folyamaton miben változtatnál?
- Felettes:
  - Hogyan zajlik a szabadságigénylés folyamata?
  - Mit tettél, amikor nem vagy csak később értesültél egy szabadságolásról?
  - Hogyan tudod nyomon követni a szabadság igényeket, azok kiadását?
  - Ha tudnád, hol és hogyan egyszerűsíténéd a folyamatot?

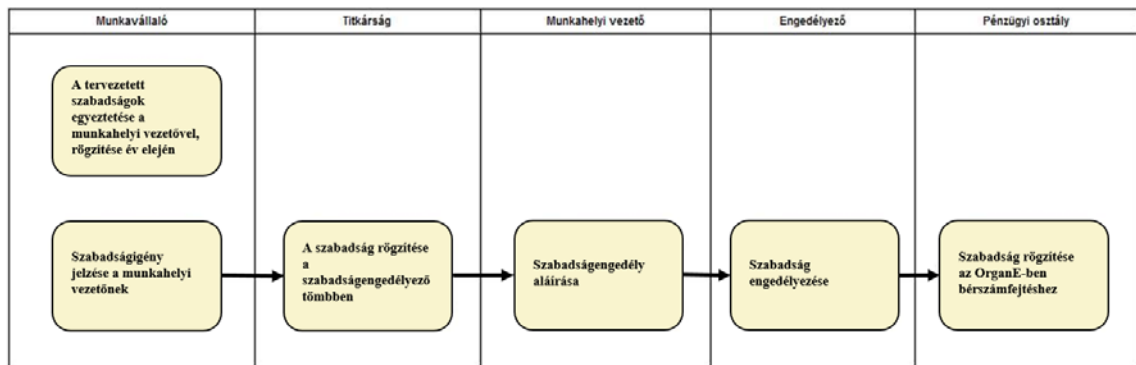
Első körben a vezető beosztású kollégákat kérdeztem, akik érintettek a szabadságok jóváhagyásában. Osztályvezető esetében felmerült a korábban említett elvárás a leendő rendszer kapcsán: az Excel táblázatok kiváltása. Elmondása szerint több táblázatból dolgozik, ami nehezíti az átláthatóságot, de ez alapján követi nyomon a szabadságokat. Nehezíti az engedélyezést, hogy minden projektvezetővel egyeztetni kell, nehéz összehangolni a projekt tervvel, illetve nehéz a kapacitásbecslés is egyeztetés hiányában. A Zalaszám esetében többféle igénynek kell megfelelni, mivel vezető kollégából több van, így az igények is változnak. Nincs ez másképp az ügyfélszolgálat esetében sem. A HelpDesk-es munkatársat kérdezve kiderült, hogy kifejezetten speciális helyzetben vannak, mivel mindig biztosítani kell a megfelelő létszámot. Akár 1-1 nap szabadság is problémát okozhat, ezért össze kell hangolni az ügyelettel. Növeli a problémát az is, hogy saját Excel táblájuk és szabadság igénylő tömbjük van a feladatra. A HelpDesk számára a tervezés is nehezebb, mivel nem látják előre a feladatokat. A legtöbb időt az egyeztetés veszi el és általában ütközések esetén érdekellentétek is keletkeznek. Ilyen helyzetben a vezetőnek „kellemetlen igazságot tennie”. Javaslatként elhangzott, hogy jó lenne egy megjegyzés mező, amiben fel lehet tüntetni az indokot, miért az egyik kolléga kapta meg a szabadságot a másik helyett.

Az interjúzást egy másik szemszögből a pénzügy és bérszámfejtés oldaláról folytattam. Ennek során kiderült, hogy a pénzügyi osztály is egy külön Excel táblázatot használ, amely így nehezebbé teszi az adatok konzisztens kezelését. A szabadságigénylési folyamatot is optimalizálni kell, hiszen papír alapú. A munkavállaló szóban egyeztet a feletteseivel, majd kiírja a titkárságon az erre kialakított tömbben.

Korábban már említettem, hogy az egyik fő cél ennek a kiváltása, hiszen a cégnél alkalmazott ISO nem szabályozza, hogy papír alapon vagy rendszerben történjen az igénylés. Ezenfelül az Excel táblázatban minden munkavállaló külön munkalapon szerepel, ahol minden szabadsággal kapcsolatos információ megtalálható. Az itt található adatokat a Pénzügy rögzíti az OrganE rendszer Munkanapló moduljába. Ezt követően lesz elszámolható a szabadság.



A kapott információk és a saját tapasztalatok alapján az alábbi folyamatábra írható le.



7. ábra: A szabadságigénylés jelenlegi folyamata

*Forrás: Saját szerkesztés az interjúk és tapasztalatok alapján*

Szintén elhangzott javaslatként a korábban említett papír alapú engedélyezési folyamat elhagyása, helyettesítése a webes alkalmazásban különböző jogosultsági funkciókkal. Célszerű lenne az adatok multiplikálásának elkerülése is, hiszen különböző kollégák ugyanazt az adatot többször is rögzítik. Új igényként merült fel az interjú során, hogy jó lenne kapcsolatot teremteni a cég beléptető rendszerével is. Szintén új elvárás, hogy kényelmi funkcióként a jóváhagyott szabadságokat utólag, később le lehessen menteni PDF formátumban vagy egyből kinyomtatni azt. Ez azért fontos, mert kiderült – bár az ISO nem írja elő – de az adóhatóság ellenőrizheti a szabadságok nyilvántartását, hiszen jogszabályi kötelezettsége a munkáltatóknak ezen adatok nyilvántartása.

Mint minden folyamat esetében, mindig vannak kiküszöbölendő problémák, amik a papír alapú igénylés esetén fennállnak, mint például a következő évre átvihető szabadságok kezelése. A kialakítandó webes alkalmazásnak tudnia kell kezelni a következő évre átvihető szabadságokat, valamint azt, amikor az előző évben megkezdett, de megszakítás nélküli szabadságon van a munkavállaló januárban, az még előző évi szabadságnak számít.

Az interjúzást a titkárságon folytattam, mivel az ott dolgozók fontos szereplői a jelenlegi folyamatnak. Legfőbb probléma az volt, hogy a kollégák nem feltétlenül ismerik a folyamatot, nem íratják alá a felettséssel az igényüket. Ez már tekinthető kiküszöbölt problémának, hiszen az alkalmazás egyik lényegi eleme, hogy jogosultság- és folyamatkezeléssel gördülékennyé tegye a szabadságigénylés és jóváhagyás procedúráját. Kiderült továbbá az is, hogy ők egy Google Naptárban rögzítik a szabadságokat, viszont az adatokat csak a vezetők látják.

A tervezés sokak számára nehézséget okoz, hiszen nehéz előre megmondani mikor lenne jó. Sokszor adott kolléga gyermekének iskolaszünete határozza meg a szabadságot. Nem tudná megmondani, hogy ki, mikor, milyen célból figyeli a tervezésre kialakított Excel táblát, de általában a 3-4 napot meghaladó igényeket rögzítik benne előre, ez viszont nem jelenti azt, hogy ki is veszik szabadságként. Nagy általánosságban véve, ha már az adminisztratív oldal informatikai rendszerrel lenne támogatva, kiváltva, már gyorsítana a folyamaton.

Minden nézőpontot figyelembe véve végül egy rendszerszervező kollégát interjúvoltam meg. Hasonló problémákat sorolt fel, mint korábban az osztályvezető. Úgy tűnt tisztában van az igénylési folyamat minden lépésével, hiszen szinte minden lépésére optimalizálási megoldást javasolt. Főként a hosszú egyeztetési folyamatot szeretné kiküszöbölni, főleg akkor, amikor váratlanul azonnal szabadságra kell menni. Adódott már konfliktusa vezetőjével, mert egy előre jelzett szabadságról elfeledkezett és számolt vele a kapacitásbecslés során arra az időszakra. Ez könnyedén elkerülhető, ha rendszerben történik már maga az igény jelzése is, amiről a felettes értesítést kap.

## 4.2. A felhasználók útja – user journey

Még mielőtt prototípus készítésbe fognék, először fontos meghatározni, hogy milyen folyamatok vezetnek a célhoz, vagyis mi szükséges ahhoz, hogy gördülékenyen működjön rendszerszinten a szabadságok igénylésének folyamata. Mindazonáltal, mivel több persona kialakult az interjúk során, így a felhasználó útját csak egy esetben írom le. Értelemszerűen ahány personánk van, annyi user journey<sup>4</sup>-t lehet felírni.

Természetesen itt még nincs szó képernyő tervezésről, hanem a jelenlegi folyamatot figyelembe véve fel kell mérni, hogy mit és milyen sorrendben tesznek a felhasználók. Ilyenkor a technológiai háttér másodlagos szempont. Az interjúk alatt kiderültek, hogy mik az alkalmazottak motivációi a rendszer kapcsán és mik a problémáik a jelenlegi folyamat során. [11]

A cél a felhasználó útjánál (user journey) az, hogy a szabadságokat átláthatóan optimálisan kezeljük, beleértve az igénylési és jóváhagyási folyamatot. Egy egyszerű lineáris procedúrát tervezünk, amely ideális és eljuttatja a felhasználót a célig. Ahhoz, hogy ezt meghatározzuk, szükség van a kontextusra, folyamatra és eszközre, ami jelen esetben adott. Ezekon kívül fontos, hogy a felhasználó mit érez egy-egy lépéskor, mit gondol ez alatt, van-e ötlete a megoldás optimalizálásához. [4][11]

A user journey célja jelen esetben a jelenlegi állapot bemutatása, ezért a problémákat helyeztem előtérbe, hiszen főként azokat mondták el a kollégák az interjúk során.

Az érzelmek kifejezésére használható hangulatjel (emoji) is, mivel ezek vizuálisan kifejezik az emóciót. Továbbá az érzelmek kifejezik az adott történésre kiváltott reakciót is. [11]

---

<sup>4</sup> A user journey (felhasználó útja) olyan események vagy élmények sorozata, amelyekkel a felhasználó egy termék vagy szolgáltatás használata közben találkozhat. [11]

A cél alapján az alábbi út állítható össze a personára:

<b>Érez</b>	<b>Gondol</b>	<b>Csinál</b>	<b>Ötlet</b>
Bizonytalanság	Hogyan fogom betervezni a feladatokat?	Szabadságok ütemezése, tervezése	Átlátható naptár felület
Tanácsstalan	Elmehetek-e szabadságra? Kit kell értesíteni a szabadságigényről?	Szabadságigénylés benyújtása	Értesítések küldése elektronikus úton is
Tájékozatlanság	Van-e kiadható szabadságom? Nyilvántartás alapján engedélyezhető-e?	Szabadságadatok ellenőrzése	Munkavállalóra vetített szabadság kimutatás (éves, kiadható, felhasznált stb.)
Kíváncsiság	Milyen feladatokban van rám szükség?	Ütemezett feladatok, tevékenységek függvényében elbírálás	Munkavállalóra vetített feladatok kimutatása, visszajelzés kérése
Bizonyosság	Teljes körű információkkal rendelkezem-e?	Döntés a szabadság engedélyezéséről	Döntéstámogató felület
Aggodalom	Kiket kell értesíteni? Mindenki értesült-e róla?	Visszajelzés munkavállalónak, érintetteknek	E-mail, SMS, értesítés küldés
Unottság	Hol kell átvezetnem? Milyen dokumentumokra van szükség?	Adminisztráció	Megfelelő dokumentáció komplett előállítás

8. ábra: A felhasználó útja

*Forrás: Saját szerkesztés az interjúk és tapasztalatok alapján*

Ezek alapján az információk alapján már látszik a tervezendő képernyő, prototípus, hiszen már tudott, hogy melyik funkció vagy lépés a folyamatban milyen érzelmeket vált ki a felhasználóból.

UX szempontból nincs alapvetően szabályozva, hogy egy folyamatot milyen lépésekre bontunk fel, ezt maga a projekt dönti el, ebben az esetben a szabadság igénylésének, jóváhagyásának, nyilvántartásának folyamatai a döntőek. A tervezés során arra törekedtem, hogy az adott lépés áttekinthető és könnyen teljesíthető legyen, így az alkalmazottak nem vesznek el lelkesedésüket a használat során. [11]

Fontos arra is figyelmet fordítani, hogy a leendő alkalmazás használatát ne kényszerként élje meg a felhasználó (például: „Jaj ki kell írnom a szabadságomat, de megint azzal az alkalmazással?!”), hanem legyen ez a folyamat egy kellemes szokás, ami kialakul az emberek tudatában.

## 5. Prototípus tervezés és megvalósítás

### 5.1. A prototípus megvalósítása

A leendő felhasználók megismerését követően és minden igényt figyelembe véve nekiállhattam a prototípus tervezésének. A képernyők tervezésének egyik alapja, hogy először papír alapon felvázoljuk körülbelül a kinézetet, aztán azt folyamatosan javítgatva eljutunk a drótvázon át a „kattintható” prototípusig.

A képernyő tervezésnek 6 alapszabálya van, ami kifejezetten a webes felületre koncentrál, a hetedik a mobiltelefonos alkalmazásokra vonatkozik. Mielőtt a szabályoknak megfelelőt terveznénk, fontos átgondolni, hogy a skicc vagy drótváz megfelel-e a 6 tulajdonságnak, amivel minden felületnek rendelkeznie kell, hiszen ha már ezeknél az állapotoknál is fennáll a hat tulajdonság megléte, akkor a prototípus is birtokolni fogja. Agilitási szempontok miatt is fontos, hogy az esetleges hiány hamar kiderüljön.



9. ábra: A UX "méhsejt"

Forrás: Saját szerkesztés Katarina Karagianni „UX Honeycomb” ábrája alapján

(URL): <https://uxdesign.cc/optimizing-the-ux-honeycomb-1d10cfb38097>

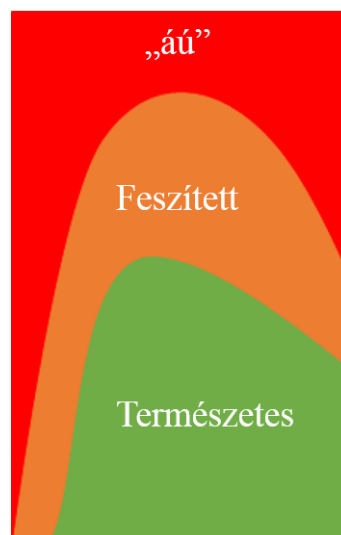
Ez egy elterjedt és népszerű reprezentációja, hogy milyen összetevőkre kell figyelni a tervezés során. A „méhsejt” jól mutatja a felhasználói élmény 7 aspektusát. Karagianni

szerint 3 fő területre lehet osztani az ábrát: gondolkodás, érzelem és használat. Az átmenet a hitelesség esetén azért van, mivel, ha használunk egy alkalmazást gondolati és érzelmi alapon is eldönthetjük, hogy az adott oldal hiteles-e. [7]

A tervezés 6 alapszabálya a következő:

- A képernyőnek 3 kérdésre választ kell adnia
- Nélkülözhetetlen a vizuális hierarchia
- A design része a szöveg is
- A konvenciók használatának fontossága
- Figyelni kell a hasznos területek arányára
- Áttekinthetőség és egyszerűség

A hetedik szabály – mint már korábban említettem – a mobiltelefonra vonatkozik. Ha mobilra tervezünk, a kezünknek is tervezünk. Ez azt jelenti, hogy a későbbiekben a mobil applikáció vagy mobil kompatibilis webes alkalmazás (reszponzív<sup>5</sup> weboldal) tervezése során figyelni kell, hogy ha egy kézzel használunk egy telefont, ujjunkkal csak a telefon alját érijük el, a többi esetben már „megfeszül” a hüvelykujj. Továbbá az is fontos, hogy az ikonok, linkek minden ujjméret számára könnyen használhatók legyenek.



10. ábra: Egy átlagos jobb kezes felhasználó telefonhasználati "problémái"

*Forrás: Saját szerkesztés Pásztor Dávid UX Design könyve alapján*

---

<sup>5</sup> A reszponzív weboldal alkalmazkodik különböző képernyők méretéhez azért, hogy minden böngészést támogató eszközön (PC, laptop, mobil, tablet) ugyanúgy optimális felhasználói élményt nyújtson.

Az ábrán is jól látható, hogy egy átlagos felhasználó a jobb kezével használja a mobiltelefont ezeket az érzéseket válthatja ki belőle a képernyő különböző pontjainak elérése. Természetesen ez minden eszköz esetében más, ez csak egy általános reprezentáció, hogyan érdemes tervezni. [11]

A hat alapszabályra visszatérve – a webes felület esetében – a 3 kérdés szabály arra vonatkozik, hogy ha egy adott oldalra, képernyőre érkezünk, tudjuk-e hol vagyunk, mit lehet ott csinálni és hogyan lehet onnan továbblépni. A felhasználók nagy része, amikor megérkezik egy oldalra, körülnéz, hogy hol van. [11]

Ennek tükrében törekedtem arra a tervezés során, hogy minden képernyő egyértelmű legyen.

11. ábra: A bejelentkező felület a rendszerben

Forrás: Saját szerkesztés

Egyszerű példa a bejelentkezési felület, amint megnyitja a felhasználó az oldalt, azonnal látszik, hogy hol van, mit tud itt csinálni és hogyan tud továbblépni. Természetesen felmerülhet a kérdés, hogy miért nincs lehetőség regisztrációra. Ez azért lehetséges, mivel minden Zalaszámos, illetve későbbi felhasználó egy előre létrehozott felhasználónevet kap (pl.: Kovács József esetén kovacsj) és ehhez egy alapértelmezett jelszót, melyet az első bejelentkezés után módosítani lehet és biztonsági szempontok miatt kell is. Ezt a



funkciót a jövőben a Keycloak nyílt forráskódú Java nyelven íródott autentikáció támogató szoftver látja el.

A vizuális hierarchia a tervezés során az egyik legfontosabb tényező, hiszen a felhasználó, amikor megérkezik egy oldalra a szembetűnő dolgokat nézi meg először.



12. ábra: Vizuális hierarchia

Forrás: Saját szerkesztés

Ahogy az ábra is mutatja, szemünk becsapható. Az első sor esetében már az is közrejátszik, hogy egy a képernyő jobb oldalán lévő tartalmat kevésbé veszünk észre és figyelembe. Ez különböző tesztekkel is alátámasztották, miszerint a felületeken az esetek többségében először a bal oldalt nézzük meg, onnan haladunk lefelé, illetve jobbra. A jobb alsó sarokra már nem jut a figyelmünkéből. Ezt a tesztet Nielsen féle F tesztnek nevezik, hiszen ha felrajzolunk egy hőtérképet az egy F betűt formáz meg. [11]

Nem elhanyagolható tény az sem, hogy a szöveg a design része. Ez azt jelenti, mivel megvan, hogy a felületen a felhasználó mely részeket tekinti meg először, ezért a figyelmét meg kell fogni megfelelő méretű, színű és stílusú betűtípussal, szöveggel. Ahogy a mondás is tartja az egyszerű nagyszerű, a felhasználók nagy része nem szereti a túl díszes font készletet. Sokkal kifinomultabb és letisztult egy Arial vagy Times New Roman stílussal írt szöveg, mint például egy *Edwardian Script*. Az utóbbi megnehezíti a felhasználó számára az olvasást és egy idő után abbahagyja az alkalmazás használatát, bármennyire is szüksége lenne rá. Ezért a tervezés során egyszerű betűtípust és fekete, fehér és szürke színt használtam, természetesen a háttér függvényében, mivel az sem jó, ha egy adott szöveg elveszik a háttérben. [11]

Vannak olyan íratlan szabályok, megállapodások, amelyeket célszerű betartani minden weboldal tervezése során.

Ilyenek például:

- az alkalmazás vagy cég logóját a bal felső sarokban helyezzük el
- a menüsáv vagy a baloldalon vagy felül legyen
- a keresési funkciónak fent középen vagy jobb felső sarokban kell elhelyezkednie
- a lábléc tartalmaz fontos információkat a céggel kapcsolatban (pl.: kapcsolat, információk, impresszum)

Ezek azért fontosak, mert a felhasználó már hozzászokott ezekhez és egy új rendszer tervezése során nem célszerű „felrúgni” a konvenciókat, hiszen a felhasználói élmény maximalizálása a cél. [14]

Minden esetben figyelni kell a hasznos területek arányára, ebben az esetben ezt könnyű megoldani, hiszen az egész oldal a „miénk”, nem jelennek meg rajta reklámok vagy fél oldalt kitöltő animált képek. Ez egy másik szempontból is fontos, mivel egy oldal betöltési ideje az azon található objektumok, képek, videók nagyságától is függ. Minél gyorsabban tölt be egy adott oldal, annál elégedettebb lesz a felhasználó, ellenkező esetben akár el is hagyhatja a weblapot. A Google, az Amazon és a Walmart is végzett erre vonatkozóan kutatásokat, nagyjából mindegyik arra jutott, hogy átlagosan 1 felhasználó 2-3 másodpercet hajlandó várni egy oldal betöltésére. Amennyiben ennél többet kellett várni, nőtt a valószínűsége az oldal elhagyásának. [15]

A tervezés során fontos volt az egyszerűség és átláthatóság is, ami lényegében nem csak a UX Design szerinti utolsó „előírás”, hanem a 3 kérdés szabálynál is kiemelt szerepe van. Ha egy oldal egyszerű és átlátható, könnyedén meg tudjuk válaszolni a három alapkérdést. Ebben az esetben egyszerűbb a helyzet, mivel egy igénylő, nyilvántartó rendszer lesz, így elkerülhetők a felesleges animációk vagy GIF-ek.

## 5.2. A prototípus bemutatása

A korábbi tervezési tapasztalatok, az interjúk és a UX design alapján készítettem el az alábbi kezdőképernyővel rendelkező prototípust:

The screenshot shows the SzabiZZ application interface. On the left is a dark blue sidebar with navigation links: Kezdőlap, Szabadságok (with sub-links Szabadság jóváhagyás and Szabadság engedélyezés), Feladatok, Értesítések, Törzsadatok (with sub-link Alkalmazottak), and Profil beállítások (with sub-links Adatok and Értesítések). The main content area has a light grey header with the SzabiZZ logo, search, notification, and user profile icons (Engedélyező Endre). Below the header are two tables.

Név	Szabadság kezdete	Szabadság vége	Kivett napok száma	Kivehető napok száma	Műveletek
Kovács Antal	2019.10.25. (P)	2019.10.25. (P)	1	12	
Lantos Béla	2019.10.28. (H)	2019.10.31. (Cs)	4	11	
Tamás Tamás	2019.10.28. (H)	2019.10.28. (H)	1	7	
Horváth László	2019.10.28. (H)	2019.10.31. (Cs)	4	3	
Németh Kitti	2019.10.31. (Cs)	2019.10.31. (Cs)	1	9	

Igénylő	Szabadság kezdete	Szabadság vége	Kivett napok száma	Kivehető napok száma	Jóváhagyva?	Műveletek
Kovács Gábor	2019.10.25. (P)	2019.10.25. (P)	1	12	igen	
Kis Artúr	2019.10.28. (H)	2019.10.31. (Cs)	4	11	igen	

Copyright 2019 © Zalaszám Informatika Kft.

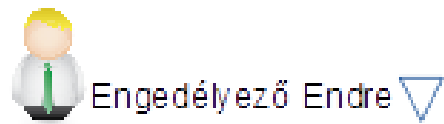
13. ábra: A prototípus kezdőképernyője

Forrás: Saját szerkesztés

A tervezési folyamat során próbáltam lemodellezni a jelenlegi folyamatot, mely alapján három szerepkör köré épült a prototípus. Ez a 3 jogosultsági szint az igénylő, jóváhagyó és engedélyező. Az ábrán látható képernyő az „Engedélyező” szerepkörrel bejelentkezett felhasználó által látott felületet mutatja, mivel elképzelésem szerint ez a legbővebb jogosultsági körrel rendelkező felhasználó. Természetesen a látható nevek és adatok fiktívek, kizárólag teszt információk. Az első körös prototípusok lényege a letisztultság és egyszerűség, mely szemmel látható.

Ahogy korábban említettem, létezik néhány konvenció, melyek betartása nem kötelező, csak ajánlott. Ebből az indíttatásból került a menüsáv a bal oldalra, illetve a bal felső sarokba a logó. Szintén ebben a sávban található a keresési funkció, az értesítések és a profil.

A keresés során munkavállalókra lehet keresni, természetesen a megjelenített információknak a mélysége függ a jogosultsági szinttől. A harang ikonra kattintva a felhasználó megtekintheti a rendszer által küldött értesítéseket, melyek a szabadság igénylés, jóváhagyás, engedélyezés folyamatához kapcsolódnak. Magától értetődő, hogy az alkalmazás minden szabadságot érintő interakcióról értesítést küld. Például, ha leadjuk igényünket, értesítést küld a jóváhagyónak, illetve az engedélyezőnek, hogy várható egy jóváhagyandó/engedélyezendő szabadságigény.



14. ábra: "Engedélyező Endre" a tesztfelhasználó

*Forrás: Saját szerkesztés*

A jobb felső sarokban található ikon egy lenyíló ablak, mellyel a fiókunkat kezelhetjük. Itt kap helyet a profil- és értesítési beállítások, illetve a kijelentkezés is. Az első két említett funkció megtalálható a főmenüben is. A főmenü felépítése egyszerű és átlátható, természetesen a megjelenített menüpontok a jogosultsági szinttől függenek.

Bejelentkezést követően a kezdőlapot látjuk, melyről több másik lapra átnavigálhatunk. A „Szabadságok” menüpontra kattintva láthatjuk a saját – korábban leadott – szabadságigényeinket, illetve itt rögzíthetünk újat. A jóváhagyás és engedélyezés két külön almenüpontra került, mivel mindkettő külön folyamat és különböző felhasználó végzi. De vannak kivételes esetek is, mint például egy igazgató szabadsága esetén a jóváhagyó és engedélyező – a Zalasám esetében – ugyanaz a személy. Mindazonáltal a későbbiek során a folyamatoptimalizálás szempontjából meg kell oldani, hogy ebben az esetben egyszerre történhessen mindkét művelet.

Feladatok menüpontra rögzíthető, hogy milyen projektben dolgozik munkatárs és mi ott a feladata. Ez azért fontos, mert a kezdőlapon megjelenítésre került, melyik kolléga van éppen szabadságon és milyen projektben érintett és mi a feladata. Ezt a funkciót fontosnak vélem, mivel az információáramlás a szervezetek többségénél nem 100 százalékos hatékonyságú és előfordulhat, hogy keresünk egy munkatársat, de nem találjuk, mert szabadságon van.

A későbbiek folyamán ezt a funkciót össze lehet kötni interfész segítségével egy projektkövető szoftverrel. A Zalasám esetében ez a Jira webes alkalmazás, melyet az

ausztrál Atlassian Co. szoftverfejlesztő cég fejlesztett Java nyelven. Szintén ez az alkalmazás segíti az agilis szoftverfejlesztést és a Scrum alkalmazását. Szintén az agilis fejlesztésnél alkalmazott módszer az MVP (Minimal Viable Product) alapján nem került bele ez funkció a prototípusba. Az MVP azt jelenti, hogy legelső „életképes” verzió bemutatásra kerül az ügyfélnek, ami alapján meghatározható, hogy az elképzelésünk jó vagy sem. [6]

A Törzsadatok menüpontban kezelhetjük az alkalmazottakat és adataikat, melyhez – a Zalaszám esetében – a pénzügyi osztálynak lesz teljes jogosultsága. Itt kerül rögzítésre a munkavállalók minden olyan adata, amely a szabadságokhoz szükséges. A dolgozók csak a saját adataikat szerkeszthetik, a módosítások érvénybelépéséhez visszaigazolás szükséges.

A kezdőképernyőre visszatérve táblázatos a megjelenítés, ahol – ahogy korábban említettem – látható, hogy melyik kolléga van éppen szabadságon adott időponthoz viszonyítva. Természetesen ebben az esetben is jogosultságtól függ, ki mit lát, hiszen egy felsővezetőnek látnia kell, kinek mennyi szabadsága van még, főleg az év végéhez közeledve. Ellenben egy alap jogosultsággal rendelkező felhasználónak elég annyit látnia, hogy a kolléga jelen pillanatban szabadságon van.

A Szabadság igények részen láthatók a jóváhagyandó/engedélyezendő szabadságok, illetve néhány információ róluk. Itt is lehetőség van jóváhagyni/engedélyezni egy adott szabadságot, de ha több információra van szükségünk az igénnyel kapcsolatban, akkor megfelelő menüpontba kell navigálnunk.

## Szabadság engedélyezés

Valóban engedélyezi Kovács Gábor szabadságigényét?

Engedélyezés

Elutasítás

Mégse

15. ábra: Felugró ablak engedélyezés esetén

Forrás: Saját szerkesztés

Amennyiben már ezen a felületen szeretnék a műveletet végrehajtani a felugró ablakok egyik fajtája, az úgynevezett modal<sup>6</sup> jelenik meg. Ezen a felületen a felhasználó három interakció közül választhat, döntésének függvényében. Ha információra van szükségünk az igénnyel kapcsolatban, akkor a megfelelő menüpontot kell kiválasztani, ami jelen esetben az Szabadság engedélyezés menüpont.

Kezdőlap / Szabadságok / Szabadság engedélyezés

Szabadság engedélyezés

Engedélyezésre vár Engedélyezett

Név	Szabadság kezdete	Szabadság vége	Típus	Kivett napok száma	Kivehető napok száma	Műveletek
Kovács Gábor	2019.10.25. (P)	2019.10.25. (P)	normál	1	12	
Kis Artúr	2019.10.28. (H)	2019.10.31. (Cs)	normál	4	11	

Copyright 2019 © Zalaszám Informatika Kft.

16. ábra: Szabadság engedélyezés menüpont

Forrás: Saját szerkesztés

A menüpont hasonló adattartalommal és ugyanolyan felépítéssel rendelkezik, mint a jóváhagyó felület. Itt van lehetősége a felhasználónak bővebb információkat kapni a szabadságról, mint például ki hagyta jóvá, milyen projektet és feladatot érint a szabadság. Szintén a korábbiakhoz hasonló felugró ablakos megoldást választottam az interakciók kezelésére, mind a megtekintés mind a szerkesztés esetében. Engedélyezést vagy elutasítást követően az igény átkerül a – képernyőn látható – másik lapra, ahol terveim szerint visszanezhetőek lesznek a szabadságok, főleg annak tükrében, hogy ezeket ellenőrizhetik a hatóságok. Eleinte mivel még viszonylag kevés adatot kell megjeleníteni célszerű az összes adatot felsorolni, viszont a későbbiek során racionális döntés egy

<sup>6</sup> A modal a weboldalak előterében hirtelen megjelenő – elsősorban – információs ablak. Fontos, hogy mögötte az eredeti oldal ugyanúgy látszik. [16]

előszűrési funkció beépítése, hiszen egy oldal betöltési sebessége nagyban függ a betöltendő adatok mennyiségétől.

A prototípus másik része az igénylő szerepkör szerinti folyamat. Törekedtem minden esetben egy lineárisan felépülő modellt alkotni, hogy gördülékenyen történjen a későbbi tesztelés. A felület lényegi elemeiben megegyezik a másik két szerepkörrel, ebben az esetben az igénylési ablakra fektettem a hangsúlyt. A megtekintési lehetőség itt is adott, illetve az új igény rögzítése is. Szintén egy felugró ablakot terveztem, melynél ugyanúgy törekedtem a jelenlegi folyamatot lemodellezni.

**Szabadság igénylés**

**Kivehető napok száma:**

**Szabadság kezdete:**

**Szabadság vége:**

**Napok száma:**

**Igénylés után maradt napok:**

**Szabadság típusa:**

**Megjegyzés:**

17. ábra: Szabadságigénylő űrlap a prototípusban

Forrás: Saját szerkesztés

A képernyőn látható szürke háttérű mezők számított mezők, nem szerkeszthetők. A kivehető napok száma adott évben még maradt napok számát mutatja, melynél többet nem vehetünk ki. A másik két számított mező értelemszerűen megmutatja, hogy hány napot jelöltünk ki és levonja a „Kivehető napok száma” mezőből az értéket. A szabadság kezdete és vége mezők dátum választó típusú mezők, mellyel kiválaszthatjuk az időintervallumot.

A prototípus tervezése során képernyőterveket rajzoltam, de a későbbiek során ez továbbfejleszhető. Akár egy HTML alapú prototípust is létre lehet hozni. Egy egyszerű példa HTML alapú dátumválasztóra:

```
<!DOCTYPE html>
<html>
<body>
  <input type="date">
</body>
</html>
```

Természetesen ezt még bőven lehet bonyolultabbá tenni és designolni.

A „Szabadság típusa” egy legördülő lista, angolul drop-down list, melyre szintén van egyszerű HTML megoldási opció – a rajzolt képernyőn túl – a <select> paranccsal:

```
<!DOCTYPE html>
<html>
<body>
<select>
  <option value="default" ></option>
  <option value="norm" >Normál</option>
  <option value="rendkivuli" >Rendkívüli</option>
  <option value="betegsz" >Betegszabadság</option>
  <option value="fiznelk" >Fizetés nélküli</option>
</select>
</body>
</html>
```

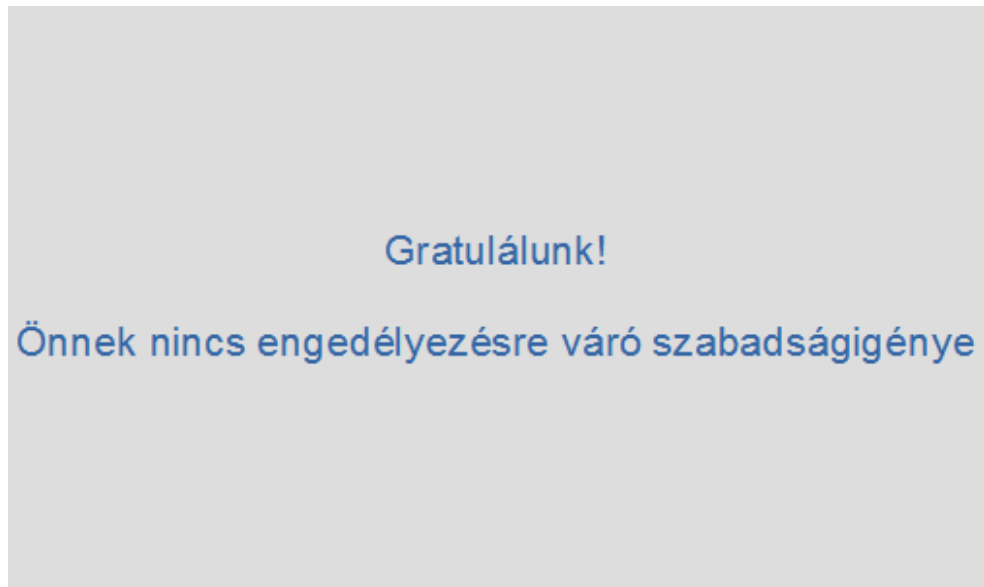
A „Megjegyzés” egy szabadszöveges mező, ahova bármely – a szabadsággal kapcsolatos – információt beírhatjuk, például rendkívüli szabadság esetén a rendkívüliség okát. Rögzítést követően értesítést küld a feletteseknek, mely után megtörténhet a jóváhagyás és engedélyezés. Természetesen ez nem azt jelenti, hogy az igénylési folyamat előtt nem kell szóban egyeztetni.

A prototípus kialakítása során fontos volt egy – korábban már említett – belső készítés megalapozása, előkészítése, hogy a prototípus alapján várják a kollégák az alkalmazás megvalósítását. Ennek az egyik legnépszerűbb módszere a felhasználó „megjutalmazása” az alkalmazás használata során. Ezt többféle módon el lehet érni, például kitüntetéssel, vicces képpel vagy animációval. A prototípusnál nem ezeket használtam, mivel úgy gondolom, hogy ezek inkább játékokhoz vagy vicces hangvételű alkalmazásokhoz valók. Az imént említettek helyett gratuláló üzenetet és a – Zalaszámnál más webes



alkalmazások során használt – jobb alsó sarokban megjelenő értesítéseket alkalmaztam.

[11]



*18. ábra: Gratuláló üzenet a prototípusban*

*Forrás: Saját szerkesztés*

A felhasználót ilyen, vagy ehhez hasonló – tartalmilag eltérő – képernyők, értesítések fogadják a prototípusban.

A kész, első verziójú prototípust az InVisionApp Inc. Invision prototípus tervező, készítő és animáló alkalmazásával „kattintható” prototípussá alakítottam, melyet ezt követően tesztelni lehet a kollégák által.

## 6. A prototípus tesztelése, eredmények értékelése

### 6.1. A usability teszt

Miután megtörtént az UX kutatás és elkészült a prototípus sor kerülhetett a használhatósági tesztelésre. Ennek a lényege, hogy olyan kollégákat kell toborozni, akik még nem használták az alkalmazást, illetve nem velük történt – a korábban már részletezett – interjúzás. Leülnek a „kattintható” prototípus elé, feladatokat adunk nekik és megfigyeljük, hogyan boldogulnak velük. Többek között arra kell figyelni, hogy hol akadtak el, hol lepődtek meg, mikor milyen gesztus és nonverbális jelet adtak a használat során. A user tesztelés során kiderül, hogy a felhasználó mit ért meg és mit nem. Ez a módszer segít megtalálni a prototípus problémás pontjait. A user tesztelés egy kvalitatív módszer, mely a miérteket mondja meg. [11]



19. ábra: Usability teszt

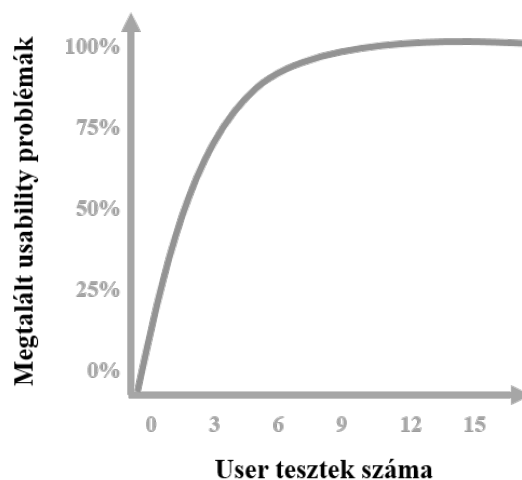
Forrás: ProfitOnline

(URL): <https://profitonline.hu/ux-ergonomia/> Letöltés dátuma: 2019. december 5.

Természetesen, mint mindennek – így a user tesztelésnek is – vannak előnyei és hátrányai, mint például, hogy kevésbé hatékony az első benyomások és érzelmek tesztelésére. Ennek tükrében egy másik UX Design tesztelési módszert is használtam, az 5 másodperces tesztet. Ez azt jelenti, hogy megmutatjuk a felhasználónak a prototípust, vagy alkalmazást és 5-10 másodperc után „elvesszük” előlük és megkérdezzük, mi az első benyomásuk a rendszer kapcsán. Az 5 másodperces teszt szorosan kapcsolódik a

korábban említett kutatáshoz, mely szerint a felhasználó csak pár másodpercet képes várni egy oldalra és annyi idő alatt dönti el, hogy használni fogja-e. [11]

Mielőtt a tesztelésről és annak mivoltáról lenne szó, fontos részletezni azt, hogy hány tesztalany szükséges. A szakirodalom általános „ökölszabályként” úgy vélekedik, hogy legalább 5 felhasználóra szükség van, hiszen ilyenkor a usability (használhatósági) problémák 85 százaléka kiderül. Mindazonáltal az is érvényes, hogy minél többen tesztelik, annál hamarabb kiderülnek a hibák. 9-10 emberre tehető az a szám, amikor elmondható, hogy a hibák 99 százalékát megtaláltuk. Ezt az alábbi ábra is szemlélteti.



20. ábra: Összefüggés a tesztelések száma és a megtalált hibák között

*Forrás: Saját szerkesztés a Nielsen Intézet és Pásztor Dávid UX Design könyve alapján*

A tesztalany toborzásban konzulensem segített, melynek során 5 rendszerszervező kolléga került kiválasztásra. Az egyik legfontosabb szempont, hogy meglegyen a korábban már a personában leírt előképzettségük, tehát előzetes tudás a szabadságokkal kapcsolatban.

Az öt kolléga előképzettsége különböző, hiszen hárman közülük idén ősszel kezdtek a Zalaszámnál, egyikük februárban, az ötödik pedig évek óta a cég kötelékében van. Természetesen a tesztelés során ezek az információk is számítanak. A tesztek egyesével zajlottak, mivel a user teszt lényegi eleme, hogy a tesztelő és a tesztet levezénylő kettesben legyen, lehetőleg egy nyugodt helyen. [11]

A usability tesztek egyik eleme, hogy a képernyő képe és a tesztalany hangja rögzítésre kerül. Ettől jelen esetben eltekintettem, mivel nem akartam, hogy a kollégák feszélyezve érezzék magukat, főleg pár hónap munkaviszony elteltével. Ehelyett jegyzeteket írtam, melyek később kidolgozásra kerültek. [11]

Mielőtt elkezdtem a tesztelést, pár fontos információt elmondtam a résztvevőknek. Részleteztem, hogy egy webes felületre tervezett szabadság nyilvántartó rendszer első verziós prototípusát fogják tesztelni, tehát nem fog minden úgy működni, ahogy azt ők elvárnák. A képernyők mögött nincs semmilyen adatbázis, vagyis konkrét adatrögzítésre nincs lehetőség. Elmondtam nekik, hogy feladatokat fognak kapni, melyeket önállóan kell megoldani, bármikor bárhova lehet kattintani, de ahogy korábban is elmondtam, nem feltétlenül lesz minden kattintás mögött interakció.

Fontos szempont, hogy nyugodtan kérdezhetnek, mert bármilyen felmerülő kérdés hasznos lehet a továbbfejlesztés szempontjából, illetve hangsúlyoztam, hogy gondolkozzanak „hangosan”. Ezenfelül még elmondtam, hogy legfeljebb egy órát fog igénybe venni a tesztelés.

Egy-egy user tesztelés felfogható egyfajta szerepjátékként is, hiszen a feladatok zöme olyan, hogy a tesztelőnek egy adott szerepkörbe kell képzelnie magát és úgy véghez vinni a kitűzött tesztfeladatot.

A tesztelés megkezdése előtt összeírtam magamnak, melyek azok a feladatok, amiket mindenképp szerettem volna, ha megcsinálnának a prototípus kipróbálása során. Első körben tágabban értelmezett feladatokkal célszerű kezdeni a UX Design szerint, mint például „Jelentkezz be a rendszerbe!”. Ezt követően lehet olyan – konkrét célhoz kötött – feladattal folytatni, mint a „Igényelj szabadságot a rendszerben!”.

Egy dolgot nem mondtam el a tesztelőknél, csak a prototípus tesztelést követően: azt, hogy a jobb felső sarokban található felhasználóikon „kidobja” a felhasználót a bejelentkezési felületre. Ennek a magyarázata egyszerű. Létezik egy módszer a „fake door” tesztelés, aminek a lényege, hogy szeretnénk megtudni, mit használnának szívesen az emberek. Ennek érdekében egy gomb mögé nem rakunk funkciót, vagy jelen esetben egy teljesen más funkciót, mint ami elvárt lenne és megfigyeljük, hogy a felhasználó hányszor kattint rá. Ha sokan „nyomkodják”, tudni fogjuk, hogy érdemes-e a prototípus későbbi verziójába, vagy a konkrét alkalmazásba belerakni azt a funkciót. [11]

Mindezek tudatában kezdődhetett el a tesztelés. A tesztelőknél kifejezett sorrendje nem volt, ahogy idejük engedte, úgy követték egymást az interjúk.

Az első teszt során egy szeptember óta a Zalaszámnál dolgozó kollégával kezdtem. Ezt követte egy régebb óta itt lévő, majd a két legújabb munkatárs, aztán befejezőként a február óta a cég kötelékébe tartozó alkalmazott következett. Ahogy említettem ez a

sorrend nem előre meghatározott volt, a szerencsének is köszönhető így tudtak időt szakítani a prototípus tesztelésére.

Az a hipotézisem, melyet a bevezetésben vettem fel már a tesztelés elején az „ismerkedési fázisban” bebizonyosodott, miszerint a kollégák nem feltétlenül ismerik a szabadságolás menetét. Érdekes módon nem csak az új kollégák esetében merült ez fel, igaz kisebb-nagyobb információja mindenkinek volt róla. Ennek tükrében ismertettem a folyamatot a kezdetektől a szabadság elszámolásáig, így minden előzetes tudás rendelkezésre állt a usability teszthez.

Az összes tesztelést a korábban leírtak alapján, a prototípus ismertetésével kezdtem. Ezt követően az 5 másodperces teszt eredménye biztató volt minden esetben, tetszett a kollégáknak a bejelentkezési felület, főleg azért, mert „olyan Zalaszámos”. Elmondható tehát a megfigyelések alapján, hogy a kollégák örömmel próbálták ki a prototípust, nem úgy élték meg a tesztet, mint időrabló tevékenység. Ez azért is fontos, mert a UX Design szerint a cél az, hogy a tesztelés során érezze jól magát a felhasználó, ne legyen feszélyezve.

Mind az öt esetben megegyezett az első feladat, miszerint igényelniük kellett a szabadságot. A második feladat az engedélyezés és jóváhagyás közül volt az egyik, mivel lényegében majdnem ugyanaz a folyamat.

Az első teszt során még a bejelentkezési felületnél felmerült, hogy „hol lehet regisztrálni?”, de ez egyedi eset volt és a későbbiek során nem tette fel senki kérdésként. A bejelentkezést követően érdekes módon egyből a „Szabadságok” menüpontot választotta a tesztelők zöme, nem időztek sokat a kezdőlapon. Tetszett nekik, hogy érthetően jelölve van az „Új igény” gomb, nincs eldugva a képernyőn. A felugró ablakon lévő mezők egyértelműek voltak. Például egyből tudták, hogy a szürke mezők nem „szerkeszthetők”, ezt egyfajta megszokásnak tekintem.

Kicsit előre haladva, összességében elmondható, hogy a Naptár funkciónak mindenki „bedőlt”. Mindegyik tesztelő azt hitte, hogy az egy rendesen működő naptár – akár ahogy korábban leírtam példaként – de elmondtam, hogy csak egy képről van szó. Lényegét tekintve ez is felfogható egy fake door tesztelésnek, de itt nem ez volt a szándék.

Visszatérve a rögzítés űrlaphoz örömmel láttam, hogy könnyedén „rögzítették” az adatokat. Az „Igény leadása” nyomógomb többeknél generált kérdéseket, abból a szempontból, hogy akkor biztos rögzíti-e a rendszer vagy még kell egy megerősítés a

felhasználó részéről. Az első terveim alapján, már egyből rögzítené az alkalmazás, de a fejlesztési ötletek, javaslatok hatására – melyekről később lesz szó – átgondoltam ezt az ötletem.

Ahogy korábban említettem, igénylő szerepkörben a tesztelők nem tartózkodtak sokat a kezdőlapon, ezért menetközben próbáltam rávezetni őket, hogy célszerű lenne megnézni, hiszen azzal kapcsolatban is szükséges az észrevételük. Összességében elmondható, hogy tetszett nekik a felület, egy-két észrevétel hangzott el.

Logikai és kronológiai sorrendben a jóváhagyás és/vagy engedélyezés következik igénylést követően, ezért a tesztelések is így folytatódtak. Ezek közül pár során mindkét szerepkört kipróbálták, bár irányított tesztelésről volt szó, mégsem volt megtiltva mit tehetnek, mit nem. Ennek során már „könnyebben” mozogtak a prototípuson belül, mivel már ismerős volt a bejelentkezési felület és a felépítések.

Érdekes módon az 5 usability teszt során csak egyetlen alkalommal kattintottak a bejelentkezési oldalon található (9. ábra) „Maradjon bejelentkezve” checkboxra<sup>7</sup>. Ez a későbbiek során elgondolkodásra késztet, hogy szükség van-e erre a funkcióra.

Bejelentkezést követően mindkét esetben egy felugró ablak fogadta a tesztelőket.



**Önnek két engedélyezendő szabadságigénye van!**

[Ugrás az engedélyezéshez](#)

*21. ábra: Felugró ablak bejelentkezést követően*

*Forrás: Saját szerkesztés*

Az ablak jobb alsó sarkában található egy úgynevezett call-to-action gomb, vagyis valamilyen cselekvésre készítő, felhívás jellegű gomb, mellyel a felhasználók egyből a jóváhagyás/engedélyezés menüpontra „ugorhatnak”. Meglepetésként ért, hogy mindegyik teszt során rákattintottak a tesztelők. Természetesen várható volt, mert ez a

---

<sup>7</sup> Checkbox vagyis választónégyzet, egy bináris értéket felvevő grafikus „ablak”.

gomb lényege, habár az ilyen jellegű gombok megosztják az embereket. Vannak, akik rákattintanak és azok, akik bezárják.

Mindkét szerepkör esetében – ahogy már említettem – hasonló képernyőket láttak, viszont így könnyebben tájékozódtak. Egy-két esetben tétozáást észleltem, melynek oka a „Megtekintés” nézetben hirtelen jött információ mennyiség volt. Mivel prototípusról lévén szó többször elakadtak, mert olyan menüpontra szerettek volna kattintani, ami nem került bele a prototípus megvalósításába. Ezen a felületen is működött az egyszerűsége és átláthatóságra irányuló törekvésem, mert könnyen jóváhagyták/engedélyezték a tesztelők az igényeket és tetszett nekik, hogy megtekinthetőek visszamenőleg azok.

Szintúgy, mint az igénylés esetében rá kellett vezetni őket, hogy a kezdőlapot is használják. Jó ötletnek tartották, hogy az engedélyező „mindent” lát, sőt építő jellegű javaslatokat kaptam.

A tesztek végeztével elmondtam nekik, hogy a jobb felső sarokban lévő felhasználó ikon a későbbiek során lesz fontos és csak „fake-door” tesztelés volt, hányszor kattintanak rá. Természetesen menetközben is rájöttek, hogy nem azt a funkciót nyújtja, amit elvárnak, de sokszor megfélekedtek arról, hogy „kijelentkezteti” őket a rendszer, ezért újra elkövették ezt a „hibát”.

Összességében úgy gondolom, hogy jól sikerült a prototípus. Természetesen sejtettem, hogy sok észrevétel fog érkezni, de a fő cél ez volt. Minél nagyobb, minél teljesebb képet kapjak az ügyféligényekről, hiszen korábban már volt róla szó, hogy jelen esetben az ügyfél egyenlő a kollégákkal.

## 6.2. Tesztelési észrevételek, javaslatok

A usability tesztek során sok észrevétel elhangzott, melyeket úgy gondoltam célszerű egyben, tesztről-tesztre haladva ismertetni, mivel néhány esetben – akár többszörös – átfedés volt. Ennek tükrében, ami többször szerepelt, azt csak egy helyen tüntetem fel.

Rögtön az első esetben a menüsáv került előtérbe és annak racionalizálása, mivel ha a felhasználó ikonra kattintunk ott is ugyanúgy meg lehet jeleníteni a profil beállításokat, illetve az értesítéseket. Ennek köszönhetően még letisztultabb lehet a rendszer. Ezenfelül felmerült, hogy jó lenne, ha minden oldalon lenne egy „Vissza a főoldalra” gomb, mellyel könnyebb lenne a navigálás. Korábban említettem, hogy kérdést okozott igénylőkor az „Igény leadása” gomb. Miszerint rögzíti-e a rendszer ennek megnyomására az igényt, vagy csak tervezettnek tekinti? Ezért ötletként felmerült – mivel mindenképp kell előre tervezni a szabadságokkal – hogy legyen tervezési lehetőség és először csak terveztként jelenjen meg a szabadság, majd később, ha véglegesíti a felhasználó váljon igényeltté az állapot.

A második teszt során megmutatkozott, hogy tapasztalt Zalaszámos kolléga tesztelte a prototípust, hiszen legelső észrevétele az volt, hogy jelölni kellene a kötelező mezőket. Erre egy bevett szokás a piros vagy fekete csillag a jobb felső indexben. Technikai ötletként merült fel, hogy lehessen törölni az igényelt szabadságot igénylő esetében, ameddig nincs jóváhagyva. Ezenfelül a projektvezetőknek jó lenne, ha látnák a saját projektjükhöz tartozó kollégákat, hiszen Zalaszám esetében az első döntést a projektvezetői egyeztetés után hozzák meg a szabadsággal kapcsolatban. A továbbiakban szó esett arról, hogy az engedélyezőknek már az igény leadását követően látniuk kellene a szabadságokat, illetve egy értesítő email sem ártana, mivel így ők is úgy tudnak kalkulálni, hogy a közeljövőben engedélyezni kell. A prototípust úgy valósítottam meg, hogy minden esetben pozitív elbírálás lehetséges, ezért felmerült igényként, hogy elutasítás esetén szükséges lenne egy indoklás mező, amiben leírható az elutasítás miéértje. A prototípuson túl, már a konkrét fejlesztéshez kapcsolódóan hangzott el, hogy abban az esetben, ha nincs a munkavállalónak kivehető szabadsága, akkor a legördülő listában csak a betegszabadság és a fizetés nélküli jelenjen meg, illetve ha elfogyott a betegszabadság is, értelemszerűen csak fizetés nélküli lesz választható. Egyszerű logikai összefüggés, de a működés szempontjából fontos.



A következő teszt során szintén felmerült a szabadság tervezési állapot, mivel megegyezik a korábban említettel, így csak említés szintjén írom. Szintén a szabadságigényléshez kapcsolódóan egy olyan igény hangzott el, miszerint célszerű lenne megjeleníteni minden felhasználó számára, hogy szabadsága miből áll össze. Például a 20 nap alapszabadság és 2 nap életkor alapján járó. Ezenfelül még egy mező, ami megmutatja, hogy hány munkanap van vissza az évből, így el lehet kerülni, hogy túl sok szabadság maradjon év végére. Továbbá, ha szerepelni fog a rendszerben, hogy ki melyik projektben dolgozik, akkor adott projekteken belül a kollégák lássák egymás tervezett szabadságait, ezáltal segítve a saját tervezésüket. Személy szerint ezt az ötletet elvettem, mivel a projektvezető így is látni fogja és első körben mindenképp vele egyeztet a projektben dolgozó. Ezen kívül a vezető beosztásúak mind az igényeknél, mind az alkalmazottak adatlapjánál látnák, hogy időarányosan mennyi szabadságot vettek ki az alkalmazottak. Tehát, ne fordulhasson elő az, hogy valakinek év végén szabadságainak 80 százaléka még bent van, illetve fordított esetben az sem szerencsés, ha valakinek már az első negyedévben szinte teljesen elfogynak a szabadságai.

A negyedik esetben szerencsére sok átfedés volt az előző háromhoz viszonyítva, pedig még idejük sem volt egymással beszélni, hiszen egymást követték a usability tesztek. Természetesen volt olyan észrevétel, amely korábban nem szerepelt, mint például az, hogy jó lenne színekkel jelölni típusonként a szabadságokat és akár alkalmazottanként lehetne egy naptár nézet, amelyben ez alapján jelennének meg az adatok. Ezen felül a vezetőknek egy külön felület, ahol a tervezett állapotú szabadságok jellennek meg és szűrni lehet projektekre, illetve munkakörökre, ezzel is segítve a kapacitásbecslést és új feladatok vállalását.

Az utolsó teszt során – hasonlóan az előzőhöz – már sok korábban elhangzott észrevétel azonos volt. Ez alátámasztja azt a tézist, miszerint legalább 5 usability teszt megtalálja a hibák nagy részét. Ezen felül felmerült a Zalaszámnak szánt alkalmazás kapcsán szükség lenne a Pénzügyi osztály által küldött üzenetek megjelenítésére is. A pénzügy minden hónap vége felé közeledve emlékeztető emailt küld az alkalmazottaknak, melyekben ismertetik a fontosabb határidőket a szabadságokkal kapcsolatban. Ezeket akár ugyanolyan tartalommal meg lehetne jeleníteni a felületen, vagy az értesítések között. Továbbá szükség lenne a nagy számú igényekre felkészülve egy jelölőnégyzet minden sorba, illetve egy „Mindet kijelöl” checkbox, amivel tömegesen ki lehet jelölni az igényeket és akár egyszerre jóváhagyni vagy elutasítani azokat. Ez a kényelmi funkció a

későbbiek során hasznos lehet a nyári vagy a karácsony előtti időszak nagy számú igényeinél. Elhangzott továbbá az is – szintúgy, mint az első tesz során – hogy teljes mértékben áthelyezhető a „Profil beállítások” és az „Értesítések” menüpont a jobb felső sarokba, de kicsit átalakítva. Például az Értesítések helyett Értesítések kezelése, hiszen így beszédesebb, könnyebben értelmezhető a célja.

Minden esetben a tesztelést követően megköszöntem a kollégáknak az idejüket és a hasznos javaslataikat, melyek a későbbiekben segítségemre lesznek egy még jobb prototípus előállításában.

## 7. Összefoglalás

Szakedolgozatom elkészítése során igyekeztem megfelelni a UX design előírásainak és úgy felépíteni azt, hogy az agilitás megmutatkozzon a tervezés során, hiszen úgy gondolom, hogy az agilitás és a UX Design együtt járnak.

Véleményem szerint, maga a diplomamunka is egy agilis folyamat, hiszen felmerül az ötlet, kutatunk, egyeztetünk a konzulensekkel, elkezdjük írni, majd átfogalmazzuk újra és újra és ez ciklikusan ismétlődik. Ezt követően az első verziót elküldjük revízióra és átvezetjük az érkezett változtatási javaslatokat. Folyamatosan javítgatjuk, szépítgetjük mindaddig, amíg elkészül a végleges, „átadható” verzió.

Ebből kiindulva esett a választás egy szabadságnylvántartó rendszer tervezésére interjúkon át a konkrét prototípus létrehozásán keresztül a usability tesztekig, mindezt a UX design felhasználásával. Ezenfelül igyekeztem tiszta képet adni az agilis szoftverfejlesztésről és azon belül a Scrumról, melyet duális gyakorlati helyemül szolgáló Zalaszám Informatika Kft. is két éve alkalmaz egyes projektjeiben.

Az, hogy egy cég 2019-ben agilis módszertan szerint próbál működni nem számít úttörőnek, de mégis más, mint például a korábbi módszertanok. Magyarországon körülbelül a 2010-es évek elején, közepén kezdett el megjelenni és egyre népszerűbb lesz, mi sem bizonyítja jobban, hogy nem csak az IT szektorban a projektek menedzselésénél jelent meg, hanem már a szervezeti struktúrákban is változást hozott. Erre példa az OTP Bank, amely elsőként a szervezeti felépítését is az agilis módszertan alapján szabta át.  
[12]

A dolgozat felépítését tekintve fontosnak tartottam ismertetni a szabadságok jogszabályi hátterét, hiszen tapasztalataim alapján sokan nincsenek teljesen tisztában vele, ezért egyfajta útmutatásként részleteztem. Úgy gondolom, hogy az adatvédelmi törvénynek történő megfelelés magától értetődő, ezért nem tértem ki rá mélységeiben, hiszen a szakdolgozatnak nem az a fő témája.

Ezt követően elmélyültem a dolgozat szakirodalmául szolgáló irodalmakba és ennek hatására számomra egyértelmű, hogy a szoftverfejlesztésnek a jövője az agilis módszertanok, hiszen világunk állandóan változik és a változásra való hajlamban rejlik a versenyképesség kulcsa.

A biztos alapokat követően a tervezés folyamatára tértem rá, mely alatt interjúk során megismertem a felhasználók igényeit és vágyait a rendszerrel kapcsolatban. Ez alapján felírható volt a felhasználó útja és a konkrét prototípus megvalósítása is. A szakdolgozat elkészítése során véleményem szerint megtaláltam a bevezetésben feltett kérdésre a választ, miszerint mire vágyik a felhasználó és ennek következtében jobban megismertem munkatársaim humán oldalát is.

A prototípus megvalósítása során ismerttettem alapszabályokat és konvenciókat, melyek alapján az elkészült prototípust a kollégák tesztelték egy usability teszt keretein belül, ami szintén a UX Design szerinti tervezés része. Ennek köszönhetően egy még tisztább kép alakult ki a – később – megvalósítandó rendszerrel és elkészült a prototípus második verziója.

Úgy gondolom az elkészített prototípus jó kiindulási alap a rendszer kapcsán, hozzátevé, hogy mindenképp szükséges egy logikai rendszerterv is, mely alapján a backend is megvalósítható.

Összegzésemet követően fontosnak tartom, hogy szemléltessem az A-ból B-be jutást, a haladási folyamatot:



22. ábra: A prototípus papíron: skicc

Forrás: Saját szerkesztés

**SzabiZZ** Engedélyező Endre

Kezdőlap

**Szabadságok**  
Szabadság jóváhagyás  
Szabadság engedélyezés

**Feladatok**

**Törzsadatok**  
Alkalmazottak

**Szabadságon lévő kollégák**

Név	Szabadság kezdete	Szabadság vége	Kivett napok száma	Kivehető napok száma	Műveletek
<input type="text" value=""/> <input type="button" value="Q"/>					
Kovács Antal	2019.10.25. (P)	2019.10.25. (P)	1	12	
Lantos Béla	2019.10.28. (H)	2019.10.31. (Cs)	4	11	
Tamás Tamás	2019.10.28. (H)	2019.10.28. (H)	1	7	
Horváth László	2019.10.28. (H)	2019.10.31. (CS)	4	3	
Németh Kitti	2019.10.31. (CS)	2019.10.31. (CS)	1	9	

**Szabadság igények**

	Igénylő	Szabadság kezdete	Szabadság vége	Kivett napok száma	Kivehető napok száma	Jóváhagyva?	Műveletek
<input type="checkbox"/>	<input type="text" value=""/> <input type="button" value="Q"/>					<input type="text" value=""/> <input type="button" value="v"/>	
<input type="checkbox"/>	Kovács Gábor	2019.10.25. (P)	2019.10.25. (P)	1	12	igen	
<input type="checkbox"/>	Kis Artúr	2019.10.28. (H)	2019.10.31. (Cs)	4	11	igen	

Copyright 2019 © ZalasZám Informatika Kft.

23. ábra: Az átdolgozott prototípus egyik képernyője

*Forrás: Saját szerkesztés*

## 8. Irodalomjegyzék

[1] **2012.** évi I. törvény a munka törvénykönyvéről [online]. Letöltés időpontja: 2019. szeptember 25. (URL): <https://net.jogtar.hu/jogszabaly?docid=A1200001.TV>

[2] **Agile Alliance:** What is Agile Software Development? [online] Letöltés időpontja: 2019. szeptember 26.

(URL): <https://www.agilealliance.org/agile101/>

[3] **Csutorás Zoltán, Kocsis Árpád:** Projektmenedzsment és az agilis szoftverfejlesztés. In: Híradástechnika LXVI. Évfolyam 2011/4, p. 51-55.

[4] **Chris Mears:** User Journeys – The Beginner’s Guide [online] 2013. április 8.

Letöltés időpontja: 2019. november 2.

(URL): <https://theuxreview.co.uk/user-journeys-beginners-guide/>

[5] **Ficsor Lajos, Dr. Kovács László, Dr. Kuser Gábor, Krizsán Zoltán:** Szoftvertesztelés, Miskolci Egyetem, Digitális Tankönyvtár, 2011 (elektronikus jegyzet, PDF) Letöltés időpontja: 2019. szeptember 11

(URL): <http://aries.ektf.hu/~gkuser/SzoftverTeszteles.pdf>

[6] **David Lowe:** What is an MVP? [online] Letöltés időpontja: 2019. november 25.

(URL): <https://scrumandkanban.co.uk/what-is-an-mvp/>

[7] **Katarina Karagianni:** Optimizing the UX honeycomb [online] 2018. december 6.

Letöltés időpontja: 2019. november 3.

(URL): <https://uxdesign.cc/optimizing-the-ux-honeycomb-1d10cfb38097>

[8] **Ken Schwaber, Jeff Sutherland:** The Scrum Guide – The Definitive Guide to Scrum: The Rules of the Game, 2017 (elektronikus jegyzet, PDF) Letöltés időpontja: 2019. szeptember 12.

(URL): <https://www.scrum.org/resources/scrum-guide>

[9] **Manifesto for Agile Software Development** [online]. Letölés időpontja: 2019. szeptember 13.

(URL): <http://agilemanifesto.org/>

[10] **Mike Cohn**: Agile Estimating and Planning, Upper Saddle River, New Jersey 07458 USA: Pearson Education Inc., 2006.

ISBN: 0-13-147941-5

[11] **Pásztor Dávid**: UX Design – Hogyan tervezz felhasználóbarát és szerethető alkalmazásokat? Budapest: UX Studio Zrt., 2016.

ISBN 978-963-12-5259-0

[12] **Portfolio**: Hazai vállalatoknál is terjed az agilis szervezeti modell [online] 2018. október 17. Letöltés időpontja: 2019. december 6.

(URL): <https://www.portfolio.hu/uzlet/20181017/hazai-vallalatoknal-is-terjed-az-agilis-szervezeti-modell-301284>

[13] **Project Management Institute**: Agile Practice Guide, Newtown Square Pennsylvania 19073-3299 USA: Project Management Institute Inc., 2017.

ISBN: 978-1-62825-199-9

[14] **Webshark**: Milyen egy jó weboldal? [online] Web Hosting Kft. 2019.

Letöltés időpontja: 2019. november 6.

(URL): <https://webshark.hu/hirek/milyen-egy-jo-weboldal/>

[15] **Webshark**: Weboldalsebesség: miért fontos, hogyan javíts rajta? [online] Web Hosting Kft. 2019. Letöltés időpontja: 2019. november 13.

(URL): <https://webshark.hu/hirek/weboldalsebesseg/>

[16] **Webshark**: Hogyan tervezz felugró ablakokat? [online] Web Hosting Kft. 2019. Letöltés időpont: 2019. november 23.

(URL): <http://blog.webshark.hu/2018/08/15/felugro-ablakok/>

[17] **w3schools**: What is Full Stack? [online] Letöltés időpontja: 2019. szeptember 30.

Hozzáférés (URL): [https://www.w3schools.com/whatis/whatis\\_fullstack.asp](https://www.w3schools.com/whatis/whatis_fullstack.asp)

## 9. Ábrajegyzék

1. ábra: Életkor alapján járó pótszabadság.....	6
2. ábra: A projekt életciklus és folytonosság kapcsolata .....	12
3. ábra: A Scrum Csapat .....	16
4. ábra: Sprintek a Scrumban .....	17
5. ábra: A Product Backlog és a Sprint Backlog kapcsolata.....	19
6. ábra: A UX Design "jéghegye" .....	20
7. ábra: A szabadságigénylés jelenlegi folyamata .....	23
8. ábra: A felhasználó útja .....	26
9. ábra: A UX "méhsejt" .....	28
10. ábra: Egy átlagos jobb kezes felhasználó telefonhasználati "problémái" .....	29
11. ábra: A bejelentkező felület a rendszerben .....	30
12. ábra: Vizuális hierarchia .....	31
13. ábra: A prototípus kezdőképernyője .....	33
14. ábra: "Engedélyező Endre" a tesztfelhasználó.....	34
15. ábra: Felugró ablak engedélyezés esetén .....	35
16. ábra: Szabadság engedélyezés menüpont .....	36
17. ábra: Szabadságigénylő űrlap a prototípusban.....	37
18. ábra: Gratuláló üzenet a prototípusban .....	39
19. ábra: Usability teszt .....	40
20. ábra: Összefüggés a tesztelések száma és a megtalált hibák között .....	41
21. ábra: Felugró ablak bejelentkezést követően .....	44
22. ábra: A prototípus papíron: skicc.....	50
23. ábra: Az átdolgozott prototípus egyik képernyője .....	51





**BGE**

BUDAPESTI GAZDASÁGI EGYETEM  
ALKALMAZOTT TUDOMÁNYOK EGYETEME

GAZDÁLKODÁSI KAR ZALAEGERSZEG

## SZERZŐI NYILATKOZAT

Alulírott, **Kocsis Gábor** 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.

Zalaegerszeg, 2019. december 11.

hallgató aláírása

## ÖSSZEFOGLALÁS

Szabadságnylvántartó rendszer tervezése Agilis módszerrel és UX design felhasználásával

**Kocsis Gábor**

**Nappali tagozat / Gazdaságinformatikus / Logisztikai informatikus**

Agilis módszertan és UX Design? Felmerülhet a kérdés, hogy függ össze a kettő. A válasz egyszerű, hiszen a UX Design és a kutatási módszerek segítenek sokkal agilisabbá tenni a fejlesztési folyamatot. Ezért is esett a választásom ezekre a témakörökre a diplomamunka elkészítése során.

A másik ok az az volt, hogy véleményem szerint maga a diplomamunka is egy agilis folyamat, hiszen felmerül az ötlet, kutatunk, egyeztetünk a konzulensekkel, elkezdjük írni, majd átfogalmazzuk újra és újra és ez ciklikusan ismétlődik. Ezt követően az első verziót elküldjük revízióra és átvezetjük az érkezett változtatási javaslatokat. Folyamatosan javítgatjuk, szépítgetjük mindaddig, amíg elkészül a végleges, „átadható” verzió.

Az, hogy egy cég 2019-ben agilis módszertan szerint próbál működni nem számít úttörőnek, de mégis más, mint például a korábbi módszertanok. Magyarországon körülbelül a 2010-es évek elején, közepén kezdett el megjelenni és egyre népszerűbb lesz, mi sem bizonyítja jobban, hogy nem csak az IT szektorban a projektek menedzselésénél jelent meg, hanem már a szervezeti struktúrákban is változást hozott. A UX Design viszont új keletűnek mondható az informatika világában, főleg a 21. század ezen szakaszán, mikor a weboldalak zöme ez alapján készül vagy újul meg.

Az ötlet, miszerint a választott projekt egy szabadságnylvántartó rendszer legyen saját és a – szakmai gyakorlati helyemet adó – Zalasám Informatika Kft. ötlete volt.

Ezen indíttatásokból készült el a dolgozat, melyet felépítésében és tartalmában is úgy készítettem el, hogy könnyen érthető legyen mind az Agilis módszertan, mind a Scrum, mind a UX Design is.

Ezek tudatában megvizsgálom, milyen jogszabályi háttere van a szabadságolásnak, ezt követően betekintést nyújtok az Agilis módszertanba, azon belül is a Scrumba, melyet a Zalasám Informatika Kft. 2017 óta alkalmaz.

A gyakorlatorientáltság jegyében a UX Design segítségével megvizsgálom mire van szüksége a leendő felhasználóknak, mely alapján felírható a felhasználó útja és a konkrét prototípus megvalósítása is elkezdhető. Az elkészülését követően elkezdődhet az úgynevezett usability teszt, ami kvalitatív módon vizsgálja a prototípust.

Úgy gondolom az elkészülő prototípus jó kiindulási alap a rendszer kapcsán, hozzátéve, hogy mindenképp szükséges egy logikai rendszerterv is, mely alapján a backend is megvalósítható.