**ARCH**

**Automated Remote Controlled Home**

**Termékterv**

|  |  |
| --- | --- |
| *Szerzők:* | *Balog Szabolcs*  *Bodnár István*  *Czetin Zoltán*  *Gyimesi Gábor*  *Hangay Zsolt*  *Várhegyi Tamás* |
| *Verzió:* | *1.0* |

Tartalom

[1. Termékspecifikáció 4](#_Toc353838752)

[1.1 Termékdefiníció 4](#_Toc353838753)

[1.1.1. Cél 4](#_Toc353838754)

[1.1.2. Tárgyterület 4](#_Toc353838755)

[1.1.2.1. In scope 4](#_Toc353838756)

[1.1.2.2. Out of scope 5](#_Toc353838757)

[1.1.3. Alapfogalmak 6](#_Toc353838758)

[1.1.4. Áttekintés 6](#_Toc353838759)

[1.1.5. Termékváltozatok 6](#_Toc353838760)

[1.2. A megoldás ismertetése 7](#_Toc353838761)

[1.2.1. Használati esetek 7](#_Toc353838762)

[1.2.2. Objektumok 8](#_Toc353838763)

[1.2.3. Állapotok és események 10](#_Toc353838764)

[1.2.4. SW architektúra 11](#_Toc353838765)

[1.2.5. HW architektúra 12](#_Toc353838766)

[1.3. Speciális követelmények 12](#_Toc353838767)

[1.3.1. Felhasználói felület 13](#_Toc353838768)

[1.3.2. Performancia 15](#_Toc353838769)

[1.3.3. Robosztusság (hibakezelés) 16](#_Toc353838770)

[1.3.4. Skálázhatóság 16](#_Toc353838771)

[1.3.5. Egyéb feltételek (feltételezések, függőségek) 17](#_Toc353838772)

[2. Projektterv a termék kifejlesztésére 18](#_Toc353838773)

[2.1. Elvégzendő feladatok 18](#_Toc353838774)

[2.1.1.Fejlesztési fázisok (Részletes feladatlebontási terv, ütemezés, kapacitás és átfutási idők becslése) 18](#_Toc353838775)

[2.2. Elkészítendő termékek 20](#_Toc353838776)

[2.2.1. Programok 20](#_Toc353838777)

[2.2.2. Dokumentáció 23](#_Toc353838778)

[2.2.3. Bemutató 24](#_Toc353838779)

[2.2.4. Oktatóprogram 24](#_Toc353838780)

[2.2.5. Kézikönyv 24](#_Toc353838781)

[2.3. Erőforrások, költségek 24](#_Toc353838782)

[2.3.1. Erőforrások 24](#_Toc353838783)

[2.3.2. Költségek 25](#_Toc353838784)

[2.4. A projekt módszertana 26](#_Toc353838785)

[2.4.1. Szereplők 26](#_Toc353838786)

[2.4.2. Eljárások 28](#_Toc353838787)

[2.4.3. Projektdokumentáció 30](#_Toc353838788)

[2.4.4. Minőségbiztosítás 34](#_Toc353838789)

# 1. Termékspecifikáció

## 1.1 Termékdefiníció

### 1.1.1. Cél

Az **ARCH** (Automated Remote Controlled Home) nevű termék kifejlesztését tűztük ki célul. Ez nem más, mint egy ún. okos-ház, amelynek irányítását automatizált és manuális formában egyaránt végezhetjük. A termék biztosítja a ház védelmét, környezetbarát energiafelhasználását és széles körű vezérelhetőségét. Automatikus módon végzi a hőszabályozást, a levegőztetést, az áramellátást és a nyílászárók vezérlését. Ezeknek a funkcióknak a hatékony irányítását a két legelterjedtebb mobil operációs rendszer (Android és iOS) segítségével végezhetné a felhasználó.

A termék részét képezné

* a megfelelő vezérlő szoftver, amely segítségével központosíthatjuk a különböző, az épület részét képező irányítható egységek működtetését
* a felhasználót és az épületet vezérlő rendszert összekötő kliensalkalmazás, amelynek segítségével az ügyfél testre szabhatja a saját lakásának vagy házának a működését
* azoknak a hardvereknek, irányító egységeknek a biztosítása és beszerelése, amelyek képesek ellátni az alapvető vezérlési feladatokat (pl.: központi szervergép, ablakot és/vagy redőnyt mozgató motorok, termosztát, biztonsági berendezések)

További cél az is, hogy a termék egy teljesen akadálymentes üzemmódot is támogasson, mivel egy ilyen „öngondoskodó” ház hatalmas segítség tud lenni a fogyatékkal élők számára is.

### **1.1.2. Tárgyterület**

A megoldás határait a következőkben definiáljuk. Külön választjuk a projekt, illetve a termék tekintetében a megoldáson belülre (in scope) és a megoldáson kívülre (out of scope) eső tényezőket.

### 1.1.2.1. In scope

A terméket elsősorban magánszemélyeknek szeretnénk értékesíteni, mivel bennük látjuk a potenciális vevőkört. Kínálatunk egyaránt érvényes társasházi lakások, egyszintes és többszintes kertes házak részére. A telepítéskor figyelembe vesszük a már meglévő eszközöket oly módon, hogy ha azok kompatibilisek az **ARCH** rendszerrel, akkor minden további nélkül telepítjük azt, ha pedig nem kompatibilisek, akkor olyan kiegészítő egységek felhasználásával szállítjuk a terméket, amelyek kiküszöbölik az inkompatibilitási problémákat.

A rendszer a következő egységek irányítását tenné lehetővé:

* Klímaberendezés
* Redőny, reluxa
* Nyílászárók (ajtók, ablakok)
* Bojler, kazán (elektromos vagy gázzal üzemelő)
* Radiátorok
* Szórakoztató elektronikai eszközök (TV, Hifi, Projektor és vetítővászon)
* Kertes ház esetén további lehetőségek:
  + Garázsajtók, kapuk
  + Autóbeálló fűtési rendszere
  + Öntöző rendszerű
* Lámpák, fényforrások
* Napelemes fűtőrendszer
* Napelemes áramellátó rendszer
* Biztonsági kamerarendszer
* Mozgásérzékelők
* Riasztóberendezés

A terméket szeretnénk bevezetni Észak-Amerikában, Európában, valamint Ázsiában. Ezen területeken kívülre csak speciális feltételek teljesülése esetén áll módunkban eladni és leszállítani a terméket. Ezek a feltételek minden ügyfélre, házra és országra különbözőek lehetnek.

A termékre 1 év teljes körű garanciát vállalunk és az első 2 évben díjmentes 7/24 telefonos ügyfélszolgálat is a rendelkezésére áll az ügyfélnek.

### 1.1.2.2. Out of scope

Kínálatunk nem terjed ki irodaházakra, gyárépületekre és közintézményekre. Nem tervezzük továbbá értékesíteni a termékünket cégek és vállalatok számára.

Az előző fejezetben (In scope) felsorolt egységeken kívül más, házba, lakásba szerelhető mechanikai eszközt nem támogat a rendszer. Így egyedi igényeket kiszolgáló berendezéseket nem áll módunkban beépíttetni, telepíteni és üzemelni.

Az afrikai országok számára jelen pillanatban nem tervezzük a rendszer bevezetését.

### **1.1.3. Alapfogalmak**

A termék összetettsége és mérete miatt elengedhetetlen, hogy néhány alapfogalmat definiáljunk. Az első rövidítés, amit tisztázni kell a termék neve. Az **ARCH**, az Automatizált Távvezérelt Otthon elnevezés rövidítése.

A következőkben definiálunk néhány fogalmat, amely a projektterv és termékterv megértését segíti:

* Automatizált: Jelen dokumentumban ez azt jelenti, hogy az adott eszköz magától működő, de manuálisan is vezérelhető.
* Távvezérelt: A termékkel kapcsolatos fogalom, jelentése, hogy egy kliens alkalmazás segítségével távolról is figyelemmel kísérhetjük az otthonunk működését.
* Vezérlő szoftver: Az a program, amely az eszközöket irányítja egy jól definiált interfészen keresztül.
* Egységek (a házban): Azok az eszközök, amelyeknek az irányítását lehetővé teszi a rendszer. Ilyen például a klíma vagy a redőny az ablakon.

### **1.1.4. Áttekintés**

Megvalósítás szempontjából két részről beszélhetünk. Az egyik, amikor a szoftveres megvalósításról beszélünk. Itt kiemelendő, hogy a szoftveres oldalát a terméknek, valamint a 24 órás ügyfélszolgálatot a GauSSoft Kft. biztosítja. A másik része a megvalósításnak az egységek beszerzésére és beépítésére irányul. A megfelelő egységeket a partnercégek fogják biztosítani, ahogyan azoknak a beszerelését és karbantartását.

A karbantartási munkálatokat a GauSSoft Kft. és a partnercégek együttműködve végzik. A hardveres hibaelhárítási feladatokat a partnercégek végzik, míg a szoftveres hibajavításokat a GauSSoft Kft.

### 1.1.5. Termékváltozatok

A termékből előre meghatározott változatok nem lesznek elérhetőek. Az ügyfél maga válogathatja össze a következő listából, hogy mit szeretne az **alap csomag** mellé, amely csak a központi szervert és a rajta futó alkalmazást tartalmazza.

|  |  |  |
| --- | --- | --- |
| Választható elemek | 1. változat | 1. változat |
| Klímavezérlés | Vezérlőegység | Vezérlőegység + klímaberendezés |
| Redőny, reluxa vezérlés | Vezérlőegység | Vezérlőegység + redőny, reluxa |
| Nyílászáró vezérlés | Vezérlőegység | - |
| Hőszabályozás | Termosztát | - |
| Fényszabályozás | Vezérlőegység | Vezérlőegység + világítástechnika |
| Napelemes fűtőrendszer, áramellátó rendszer | Vezérlőegység | Vezérlőegység +  napkollektorok |
| Biztonsági rendszer 1. | Kamerarendszer | Kamerarendszer + mozgásérzékelők |
| Biztonsági rendszer 2. | Riasztóberendezés | Riasztóberendezés + mozgásérzékelők |
| Szórakoztató rendszer 1. | Vezérlőegység | Vezérlőegység + Samsung SMART TV |
| Szórakoztató rendszer 2. | Vezérlőegység | Vezérlőegység + Samsung Hifi |
| Szórakoztató rendszer 3. | Vezérlőegység | Vezérlőegység + Projektor + vetítővászon |
| Távvezérlés 1. (Android) | Kliensalkalmazás | Kliensalkalmazás + Samsung Note 10.1 táblagép |
| Távvezérlés 2. (iOS) | Kliensalkalmazás | Kliensalkalmazás + iPad |

## 1.2. A megoldás ismertetése

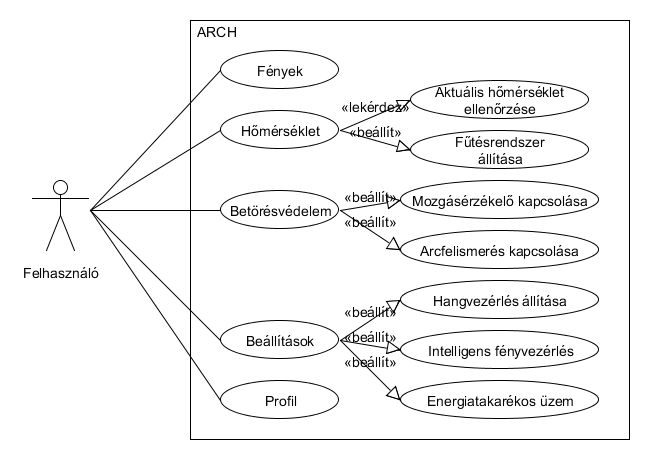
Ebben a fejezetben a szoftver és a hardverek részletes bemutatása található. A szoftver struktúráját UML diagramokon keresztül tanulmányozhatjuk rövid magyarázatokkal. A hardver architektúra fejezet a hardver felépítését segíti megérteni, amin az ARCH szoftver fut.

### 1.2.1. Használati esetek

A felhasználó szempontjából öt főbb használati esetet különböztethetünk meg úgy, mint:

* Fények állítása
* Hőmérséklet állítása
* Betörésvédelem állítása
* Beállítások módosítása
* Profil beállítások

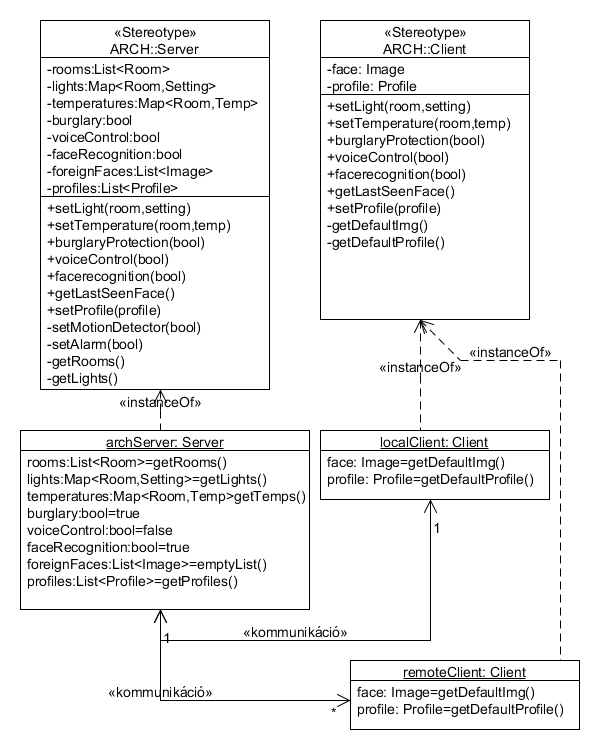
Ezek a lehetőségek elérhetőek mind az otthoni kliensről, mind a mobil kliensről.



**1. ábra: Használati esetek**

### 1.2.2. Objektumok

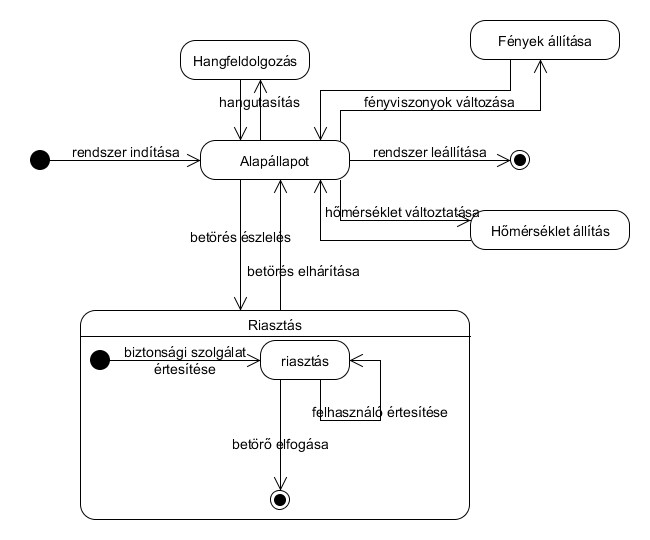
A 2. ábrán látható a szoftver objektumainak kapcsolata. Leolvasható, hogy a szerver funkcióit az ARCH::Server osztály valósítja meg mellyel kommunikálnak a kliensek (helyi kliens, távoli[mobil] kliens).



**2. ábra: Objektumok kapcsolódása**

### 1.2.3. Állapotok és események

A rendszerindítást követően alapállapotba kerül, ahonnan különböző események hatására az eseménynek megfelelő állapotba kerül a rendszer. Például, ha a fényérzékelők a külső fények változását érzékelik, akkor a lakásban lévő fényeket ennek megfelelően módosítják. Az egyik legfontosabb állapot a riasztás állapota (3. ábra). Addig marad a rendszer ebben az állapotban, amíg a biztonsági szolgálat a helyszínre nem érkezik és jelzik ezt a rendszer számára. Eközben a rendszer a tulajdonost is értesíti a betörés tényéről.



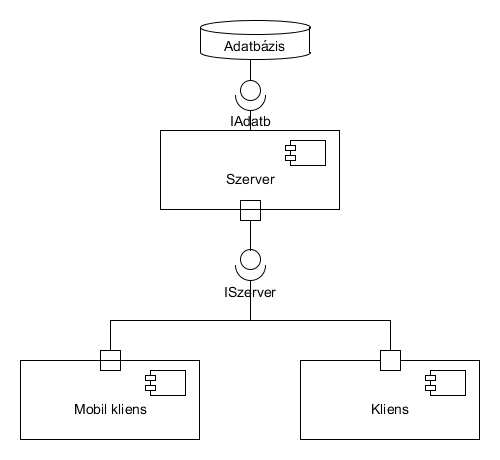
**3. ábra: Állapotok és események**

### 1.2.4. SW architektúra

A szoftver kezdeti verziója négy fő komponensből áll, ezek:

* Adatbázis
* Szerver
* Mobil kliens (távoli kliens)
* Helyi kliens (otthoni)

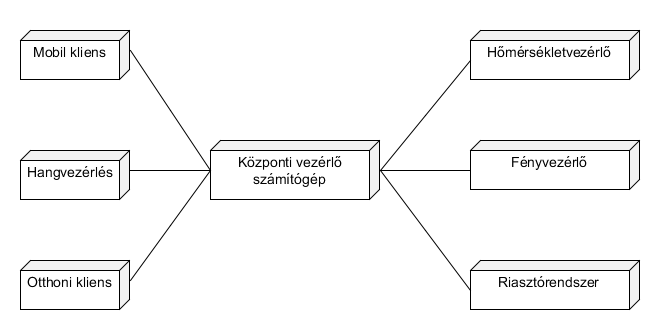
Az adatbázis tárolja a profilokat, amiket létrehozhat, módosíthat, törölhet a felhasználó, valamint itt tárolja a szoftver az arcfelismerő által rögzített képeket. Ha a felhasználó lekéri a legutóbb rögzített képet, akkor a szervernek küld üzenetet, majd a szerver lekéri az adatbázistól a szükséges adatokat és visszaküldi a kliensnek.



**4. ábra: Szoftver architektúra**

### 1.2.5. HW architektúra

Az alábbi ábrán látható a különböző hardverkomponensek kapcsolata. A központi vezérlő számítógép szerepe a legfontosabb, ez tartja a kapcsolatot a kliensekkel és végrehajtja a kliensek által kért változtatásokat a lakásban.



**5. ábra: Hardware architektúra**

## 1.3. Speciális követelmények

Az ARCH telepítéséhez szükséges hardveres valamint szoftveres előfeltételek teljesítése.

**Hardveres követelmények:**

* a lakás megfelelő elektromos vezetékezése
* az elektromos hálózat bővíthetősége valamint egy
* 1 m3-es hely a központi rendszer számára.

Ezen feltételek teljesülése esetén, szakembereink felszerelik a szükséges hardvereket, aminek megfelelő működése esetén telepíthető az ARCH szoftver.

**Szoftveres követelmények:**

* A központi számítógépen futó vezérlőszoftver futtatásához szükséges Linux operációs rendszer
* a rendszer távvezérlését biztosító Androidos kliensprogramhoz Android 4.0 vagy újabb
* iOS kliensprogramhoz iOS 6 vagy újabb

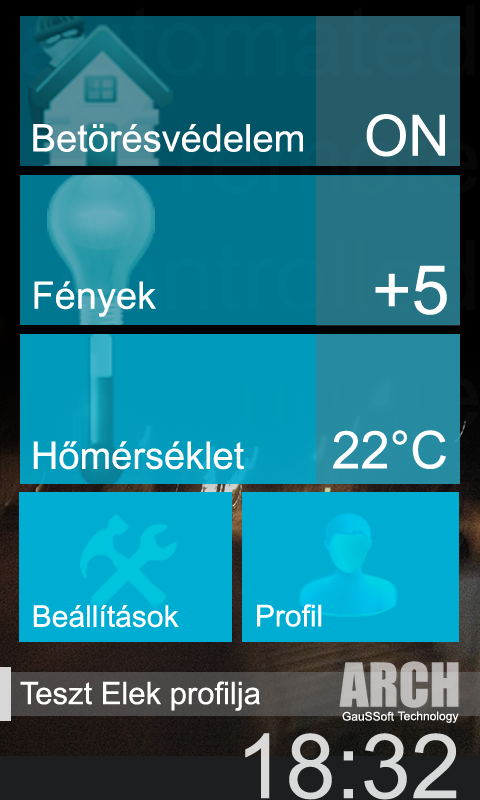
### 1.3.1. Felhasználói felület

A felhasználói felületnek egyszerűnek kell lennie, hogy könnyen átlátható legyen és a szükséges funkciók gyorsan megtalálhatóak legyenek, ezért a főpanelen megjelenő vezérlőprogram menüjére egy terv látható az *6. ábrán*.



**6. ábra: Otthoni kliensszoftver felhasználói felülete**

Mivel a felhasználók nem csak otthonról irányíthatják a rendszert, ezért a távvezérlést megvalósító programnak is felhasználóbarátnak kell lennie. Annak ellenére, hogy a program elérhető iOS és Androidos eszközökről is, egységes felület fogadja mindkét rendszeren a felhasználókat, amit a *7. ábrán* láthatunk.



**7. ábra: Mobil felhasználói felület**

***Fények menüpont:***

* összes fény le/összes fény fel
* helyiségenként kapcsolható fény

***Hőszabályozás:***

* Fűtés kikapcsolása/adott hőfokra állítása
* Automatikus évszaknak megfelelő beállítás

***Betörésvédelem:***

* Mozgásérzékelők bekapcsolása/kikapcsolása
* Riasztók bekapcsolása/kikapcsolása
* Arcfelismerő által rögzített képek megtekintése

***Beállítások:***

* Riasztási hang kiválasztása
* Figyelmeztető jelzések
* Hangvezérlés ki/bekapcsolása

***Profil:***

* Új profil készítése
* Profil kiválasztása

### 1.3.2. Performancia

Mivel az otthoni kliensszoftverünk működéséhez speciális hardverre van szükség, amit megrendelés esetén megkap a felhasználó, ezért az optimalizáció viszonylag egyszerű feladat. Sokkal érdekesebbek a távvezérlést megoldó kliensprogramok. Figyelembe kell venni az eszközök különböző teljesítményének sokszínűségét. A programnak gördülékenyen kell futnia gyengébb hardverrel rendelkező mobilokon/tableteken. Ennek megfelelően semmi számításigényes feladatot nem hajt végre a kliens, csak az otthoni szerverrel kommunikál.

A kommunikáció során fontos a biztonság ezért SSL kapcsolaton keresztül folyik a kommunikáció.

***Elvárások:***

* A riasztások a betörés észlelésétől számított 2 percen belül meg kell érkezzen felhasználó mobiltelefonjára (SMS). Ez lehet gyorsabb a telefonos hálózat terheltségétől függően.
* Ha a felhasználó rendelkezik mobil internet kapcsolattal, akkor szinte azonnal megjelenik a riasztás a felhasználó telefonján. Az internetkapcsolatnak minimum 56kbps-osnak kell lennie a megfelelő kommunikációhoz és sebességhez.
* Az otthoni szerver egyszerre 10 kliens tud kiszolgálni. Általában ennél többre nincs szükség, és biztonsági okokból ésszerűbb korlátozni a kapcsolatok számát.

### 1.3.3. Robosztusság (hibakezelés)

Mivel az ARCH sok komponensből áll, melyek egymással kommunikálnak így a legtöbb hibalehetőség itt léphet fel. Ezen hibák elkerülésére a vezérlő és a kliens szoftver e részei különösen optimalizáltak és minden hibás működést helyesen kezelnek. Ilyen hibás működés lehet például, hogy az elérni kívánt hardver nem található, ekkor a rendszer amellett, hogy a felhasználónak jelzi a hibát, automatikus üzenetet küld a központnak a hibáról. Ezen kívül automatikus hibajavító algoritmusokat helyezünk el a programban, melyek például egy esetleges hibás üzenetet kijavítanak, vagy sikertelen üzenetküldésnél megkeresik a hiba okát és kijavítják vagy tudatják a felhasználóval. A hardvereszközök hibáiért természetesen azok gyártói felelősek, így a javítás is rájuk hárul.

Ezen kívül a felhasználók rendelkezésére áll egy telefonos és egy internetes ügyfélszolgálat, melyen a hibákat bejelenthetik. A telefonos ügyfélszolgálat rendelkezésre állását igény szerint fogjuk kialakítani, azaz kezdetben rövidebb elérhetőséggel főleg a kritikus időintervallumokban fog üzemelni a szolgáltatás – 6 és 8 óra valamint 16 és 21 óra között – azonban ha erre szükség van, tovább fogjuk növelni ezt az időt. Az interneten történő bejelentéseket természetesen a nap többi részében is nyomon követjük, és munkatársaink azonnal intézkednek, valamint erről értesítik a felhasználókat is. A bejelentések alapján a hibát egy munkatársunk irányításával a felhasználó meg tudja oldani, vagy ha nem akkor kiküldünk egy megfelelő képesítéssel rendelkező szerelőt, aki kijavítja a hibát, amennyiben az nem olyan jellegű, hogy cserére van szükség.

### 1.3.4. Skálázhatóság

Termékünk nem igényel nagyméretű adatbázisokat, valamint nem végez bonyolult számításokat és az egy időben történő nagyszámú hardverelemmel való bővítés sem áll fent, ezért a hangsúlyt a rendelkezésre állásra kell fektetnünk. A kezdetben üzembe helyezett vezérlőegységek (a későbbi bővítések során is) teljes mértékben el tudják látni a feladatukat, rendszer hatékonyságát nagymértékben csak az internetkapcsolat sebessége befolyásolja. A vezérlő egységet tartós tesztnek vetjük alá így biztosak lehetünk abban, hogy a folyamatos készenlét csak rendkívüli esetekben nem áll fenn, mint például egy hosszabb áramszünet vagy a készülék amortizációjából adódó meghibásodás, azonban ezt a hibát az előző pontban ismertetett módon gyorsan orvosolni lehet.

### 1.3.5. Egyéb feltételek (feltételezések, függőségek)

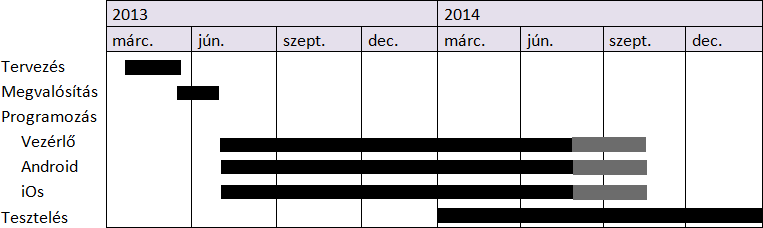
A termék fejlesztésének egyes fázisai nagyban függnek attól, hogy a hardverek gyártói időben leszállítják-e a megrendelt tesztpéldányokat, hiszen ezek nélkül nem kezdődhet meg a vezérlő adott részének programozása vagy a tesztelése. Valamint figyelembe kell venni az egyes eszközök szabványosságát is, amely jelentősen befolyásolhatja a vezérlésének programozására szánt időt, mivel a korábban működő optimalizálási technikák nem feltétlenül jelentenek sebesség növekedést, sőt egyes esetekben akár lassuláshoz is vezethetnek. Az egyes hardver cégek nem állnak kapcsolatban és ezért nem függenek egymástól. A projekt egyes fázisaira sincs hatással a beszállítók munkája, ilyen például a kliens alkalmazások fejlesztése, amelyhez csak a vezérlőprogram specifikációjának ismerete szükséges, ami viszont nem változik az egyes hardverek változásával.

Bár nem szorosan a termék fejlesztéséhez tartozik, érdemes megemlíteni, hogy a gyártók pontossága a későbbiekben is fontos szerepet kap az ügyfelek részére történő szállításkor. Ezen kívül érdemes figyelembe venni a hardverek esetleges későbbi változásait is, akár egy újabb modell vagy egy firmware frissítés esetén is szükség lehet a vezérlő program minimális módosítására.

# 2. Projektterv a termék kifejlesztésére

## 2.1. Elvégzendő feladatok

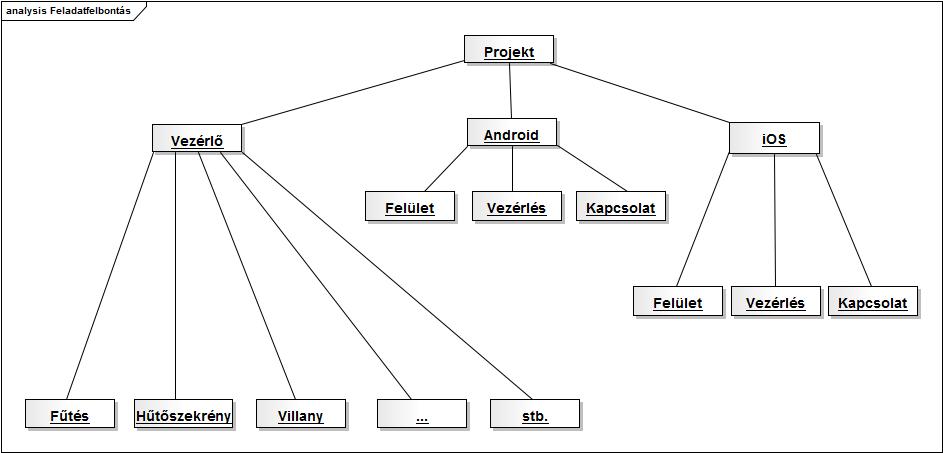
### 2.1.1.Fejlesztési fázisok (Részletes feladatlebontási terv, ütemezés, kapacitás és átfutási idők becslése)



**8. ábra: Fejlesztési fázisok**

***Tervezés (feltételezve a kész követelményspecifikációt)***

A tervezés során, a követelményspecifikáció alapján felépítjük a rendszer tervét, kijelöljük az egyes részekhez tartozó felelősöket. Létrehozunk egy kezdeti erőforrás ütemezést, mely alapján megkezdhető a munka, de természetesen a későbbi eredmények tükrében ez az ütemezés megváltoztatható, az adott időszakra újraírható. Megbecsüljük a projekt és az esetleges csúszások idő, erőforrás és anyagi költségeit.



**9. ábra: Feladatlebontás**

***Megvalósítás***

A megvalósítás során a kiválasztott technológiák segítségével létrehozzuk a projekt vázát, pontosítjuk, és a fejlesztőeszközökre szabjuk a tervezés során elkészült terveket. A vezérlő megvalósításához a gyorsaság miatt a C++ nyelvet és egy erre alkalmas fejlesztői környezetet fogunk használni, ennek eldöntése megbeszélés tárgyát képezi. A kliensek esetében az Eclipse IDE-t és a hozzá telepíthető Android SDK-t valamint az Apple XCode nevű fejlesztőeszközét fogjuk használni.

***Programozás***

A projekt programozási része alapvetően 3 fő részből áll, melyek fejlesztése párhuzamosan fut. Ezek a vezérlő, az Android alapú kliens valamint az iOS alapú kliens programozása. Ezen komponensek függetlenek egymástól, csak a tesztelés során van szükség a vezérlőnek és a két klienssel történő összehangolására. A vezérlő programjának fontos része a hardverekre való optimalizálás, ez a kód írása során folyamatosan történik. A kliens programokat három fő részre oszthatjuk ezek a felület, a vezérlő és a kapcsolat. A felületnek a megtervezése is időigényes, hiszen egy olyan GUI-t kell létrehozni, amely szabadon bővíthető, testre szabható és ergonomikus, bármilyen méretű kijelzőn jól kezelhető és átlátható. A vezérlő a felhasználó által vezérelt eseményeket kezeli le, összeköti a felületet a kapcsolati réteggel. A kapcsolat rész pedig a kommunikációért felelős, ez küldi az ügyfél által kiválasztott adatokat a vezérlőnek, ami továbbítja a megfelelő hardvernek.

Ebben az időszakban sem használjuk ki a teljes kapacitást, hiszen az esetleges csúszásokat figyelembe kell venni és minimalizálni az általuk okozott kárt, ez csak úgy lehetséges, ha vannak szabad erőforrásaink, amelyeket szükség esetén rendelkezésre tudunk bocsátani.

***Tesztelés***

A tesztelés megkezdődhet már azelőtt, hogy a kód teljes mértékben készen lenne, hiszen alap, vagy már meglévő funkciókat így is lehet tesztelni, valamint a már említett vezérlő-hardver és vezérlő-kliens kapcsolatokat is, hiszen itt fordulhat elő a legtöbb hiba, ami nem a kódban található, hanem rajtunk kívülálló okokból lép fel.

A periódus elején kicsivel megemelkedik a projekt erőforrásigénye, hiszen a párhuzamosan folyik a programozás és a tesztelés, azonban ez lehetőséget ad az erőforrások átütemezésére is, így akár idő előtt vagy költségen belül végezhetünk a projekttel vagy az addig keletkezett hátrányokat behozhatjuk vagy csökkenthetjük.

## 2.2. Elkészítendő termékek

A következőkben elsődleges célunk definiálni az előállítandó termékeket, azok előállítandó részegységeit. A fő termék az ARCH (Automated Remote Controlled Home) házkezelő szoftverrendszer, amelynek főbb részegységei mellett, azonosítanunk kell a hozzá tartozó dokumentációkat, bemutató és oktatóprogramokat.

### 2.2.1. Programok

Ebben a fejezetben felsoroljuk az elkészítendő programok listáját. Ezek fejlesztése során az erőforrások megoszlását a következő diagramon láthatjuk:

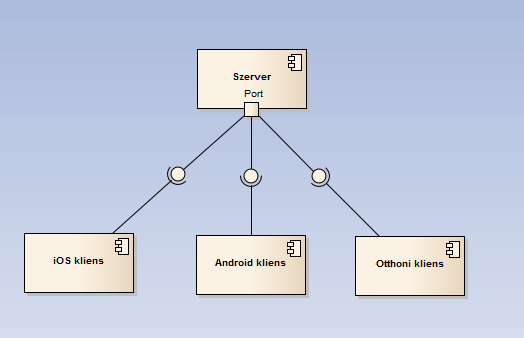
**10. ábra: Erőforráselosztás**

***Szerver***

A központi termék a szerver alkalmazása a programnak. A szerveralkalmazás rendelkezik egy adatbázissal, ahogy azt a komponensdiagramban is láthatjuk. A szerver kezeli a különböző funkcióit az okos-háznak, melyeket a különböző kliensek segítségével érhetünk el. Ehhez implementálnunk kell a szerveren a következő funkciókat:

* Beszédfeldolgozó rész
* Nyílászárók kezelése
* Betörésgátló funkció
* Hőmérséklet kezelése
* Fények kezelése

A szervernek képesnek kell lennie együttműködni a különböző kliensekkel, ehhez egységes interfészt kell nyújtania a kommunikációhoz:



**11. ábra: Komponensek**

***Otthoni kliens***

Az elkészítendő programok listáján következő az otthoni használatra kialakított kliens. Ez közvetlen helyi hálózati kommunikációval kapcsolódik a szerverhez. Az otthonban felszerelt különböző érintőkijelzős felületeken keresztül, vagy otthoni gépről irányíthatjuk ezt a klienst. A kliens hangvezérléssel is működik, melynek feldolgozása a szerveren történik.

***Android kliens***

Ahhoz, hogy távoli eszközről is elérhető legyen az otthonunk, készítenünk kell egy Android alkalmazást, amelyre az otthon megadott profil beállítások segítségével jelentkezhetünk be. Ez a kliens egy biztonságos kommunikációs csatornán fog kommunikálni az otthoni szerverrel.

***iOS kliens***

Az iOS kliens célja ugyanaz, mint az Android kliensé, azonban szükséges ennek elkészítése is, hogy a felhasználók igényeihez alkalmazkodjunk, és ne kelljen függni a telefon vagy operációs rendszerétől.

### 2.2.2. Dokumentáció

***Technikai leírás***

Szükségünk van egy technikai leírásra a fejlesztés befejezte után, mely összefoglalja a termék technikai specifikációját és segítséget nyújtana a rendszer megértése érdekében. A technikai leírás alapján újabb fejlesztőknek nyújtunk segítséget a rendszer megértéséhez. Későbbi újabb fejlesztésekhez is segítséget nyújthat ez a dokumentáció. A leírás tartalmazza a rendszer felépítését, a komponensek leírását.

***Felhasználói leírás***

A felhasználói leírás (User Manual) a felhasználók számára készült dokumentáció, amely a rendszer használatát írja le. Ennek tartalmaznia kell a felhasználó által elérhető kliensek dokumentációját. Ebben megtalálhatjuk a kliens által biztosított funkciók használatát, illusztrációkkal, képernyőképekkel, a különböző jelölések jelentését. Tartalmaznia kell továbbá a gyakran előforduló kérdéseket hozzájuk tartozó válaszokkal. A dokumentáció végén megtalálhatjuk a támogatással és garanciával kapcsolatos információkat, a cég támogatócsapatának elérhetőségeivel.

***Terméktámogatási dokumentáció***

A Terméktámogatási dokumentáció (Support Manual) az ügyfélszolgálat számára elkészítendő dokumentáció. Ebben összefoglaljuk a hasznos információkat a gyakrabban előforduló hibalehetőségekről és ezek elhárításáról. A dokumentáció kevésbé rendelkezik szakmai nyelvezettel, mint egy technikai dokumentáció, inkább gyakorlati szemmel kell megközelítenünk a dokumentációt annak készítése közben. A dokumentációban megtalálhatjuk még az ügyfélszolgálatosok tréning anyagát is, melynek átolvasásával elsajátítható a szükséges technikai tudás. Ez a rendszer alapvető áttekintését jelenti, mely elengedhetetlen a megfelelő szolgáltatás nyújtásához.

***Telepítési dokumentáció***

A telepítési dokumentáció (Install Document) a field engineerek számára előállított dokumentáció. Ebben részben technikai információkat szedünk össze a telepítés komponenseinek megértésére, másrészt a részletes útmutatót, amely a telepítés folyamatához szükséges. A telepítési útmutatónak tartalmaznia kell a szükséges hardver specifikációkat is.

### 2.2.3. Bemutató

A termék bemutatója az érdeklődők számára készülő prezentáció. Ebben a prezentációban a felhasználói szempontból előnyös funkciók bemutatása a cél. Főbb funkciók kiemelése mellett ebben a bemutatóban hangsúlyoznunk kell azokat a funkciókat, melyek cégünk termékét egyedivé teszik a piacon.

### 2.2.4. Oktatóprogram

A felhasználók számára egy gyors és egyszerű oktató programot is biztosítunk a funkciók elsajátítása érdekében. Ebben az oktatóprogramban először végigvezetjük a felhasználót a rendszer felhasználói konfigurációs lehetőségein majd a profilbeállításokon. Miután megtörtént a konfiguráció a felhasználót az oktatóprogram végigkalauzolja a felhasználót a főbb funkciókon, megmutatva animációk segítségével azok elérését és használatát. Az oktatóprogramnak 2 változata lesz elérhető: egy változat a házban található kliens kezeléséhez és egy a mobilos klienshez.

### 2.2.5. Kézikönyv

Az oktatóprogramon kívül elérést biztosítunk egy elektronikus kézikönyvet is a kliensek használatához. A kézikönyvben megtalálhatjuk minden elem leírását a kliensekben illusztrációkkal. A kézikönyv interneten keresztül és helyi letöltéssel is elérhető. A kézikönyvben gyors keresést biztosítunk, tárgymutatókkal és tartalomjegyzék segítségével. A kereséshez hangvezérlést is alkalmazhatunk, amelyet a szerver rendszerében használt hangvezérlésből ültetünk át.

## 2.3. Erőforrások, költségek

### 2.3.1. Erőforrások

A projektet tekintve két nagy csoportra oszthatjuk az erőforrásokat: humán és tárgyi erőforrásokra.

***Humán erőforrások***

A humánerőforrás-menedzsment a projektvezetői feladatok egyike, mely magába foglalja a kommunikációt a humánerőforrások felé. Ennek része a beosztás egyeztetése, azaz hogy ki mennyi szabad kapacitással rendelkezik, valamint az ez alapján elkészített terv megosztása, aminek minden hónap első hetében meg kell történnie.

A humánerőforrások jelzik a projektvezető felé amennyiben szabad kapacitással rendelkeznek, illetve késéssel fejezik be a feladatokat.

#### Tárgyi erőforrások

A tárgyi erőforrások igénylését és szétosztását szintén a projekt-vezetőnek kell lebonyolítania.

A projektcsapat tagjainak a fejlesztésekhez rendelkezésre áll egy-egy hordozható számítógép (notebook), amelyen mindenki a saját feladatait végezheti.

Továbbá a cég szervergépeket bérel a nagy mennyiségű és fontos adatok tárolására és biztonsági mentésekre. A mobil eszközökkel történő használhatóság végett a fejlesztendő alkalmazáscsomag tesztelésére, mobiltelefonokra is szükség van, a folytonos kommunikációról nem beszélve a csapattagok között.

### 2.3.2. Költségek

A projekt során felmerülő költségeket felbecsülve egy táblázatba foglalhatjuk össze, amelyet alább láthatunk:

|  |  |
| --- | --- |
| Tétel | Becsült költség (HUF) |
| Irodabérlés | 250.000 / hó |
| Notebookok (6 db) | 300.000 / db |
| Szervergépek (3-5 db) | 1.500.000 / db |
| Hardver elemek gyártása | 1.000.000 / db |
| Munkabérek havi kifizetése | ~400.000 / fő |
| Telepítési költségek (~10 fő) | 500.000 / fő |
| Mobiltelefonok beszerzése (6 fő) | ~ 200.000 / db |
| Mobilkommunikáció havidíja | 5-8000 / fő |
| Összköltség éves szinten | ~ 100.000.000 / év |

## 2.4. A projekt módszertana

A projekt sikerének egyik záloga kétségkívül a megfelelő előkészítés. Fontos, hogy az alapos helyzetelemzés után, a tervezés és a végrehajtás is előre átgondolt és megtervezett folyamat legyen.

Ezeknek a gondolatoknak a tudatában mi egy folyamat-alapú (agilis) projekt menedzsment megközelítést választottunk, a SCRUM-ot. Ez az a fejlesztési módszertan, amely a legjobban illeszkedik az általunk előállítani kívánt termékhez. Ebben a folyamat-alapú modellben fontos az egyén és a személyes kommunikáció, a működő szoftver az átfogó dokumentációval, az együttműködés az ügyféllel, valamint a változásokra való gyors reagálás.

### 2.4.1. Szereplők

***Termékgazda***

* Gyimesi Gábor

Feladata, hogy megfelelő módon képviselje az ügyfél hangját, vagyis inkább a „mit” kérdésével foglalkozik, mint a „hogyan”-nal. Gondoskodnia kell a termékvízióról és világossá kell tennie a csapat számára azt.

Felelős a kibocsátási dátumok és azok tartalmának meghatározásáért és biztosítja, hogy üzleti szempontból a csapat hasznosan töltse az idejét, azaz felelős a termék nyereségességéért. Végül pedig adminisztrálnia kell a termék hátralékot (backlog), azaz, hogy mit kell csinálni.

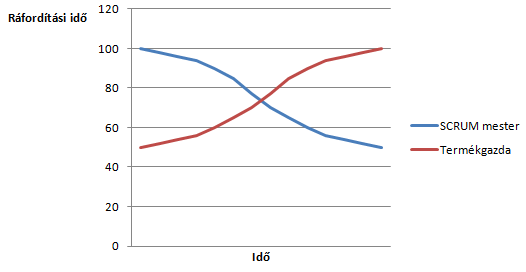
***SCRUM mester***

* Bodnár István

Nevezhetjük akár „team leader”-nek is. Felelős a SCRUM értékek és gyakorlatok meghatározásáért. Résztvevője a napi SCRUM-nak és feladata az fejlesztés során felmerülő akadályok elhárítása. A SCRUM mesternek biztosítania kell a csapat termelékenységét és védeni a csapat munkáját a külső hatásoktól.

Minden Sprint után értékelő megbeszélést kell tartania a csapat tagjaival, ahol a tapasztalatokat és következtetéseket le lehet vonni. Fontos, hogy a csapatot támogató projekt szereplőről van szó.

Ahogy belejön a csapat a munkába, úgy fordul meg a termékgazda és a SCRUM mester ráfordításigénye. (lásd 12. ábra)



12. ábra: Ráfordításigények

***Csapat***

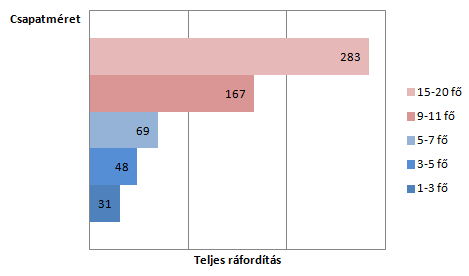
Szoftverfejlesztés elengedhetetlen kelléke egy jól működő SCRUM csapat, ahol a csapattagok határozzák meg a munka menetét és kiosztását és a csapat nagymértékben önszerveződő.

Csapattagok:

* Balog Szabolcs
* Czetin Zoltán
* Hangay Zsolt
* Várhegyi Tamás

A csapat létszámát tekintve kis csapatról van szó, annak minden előnyével. Ilyen például, hogy kevesebb idő szükséges a koordinációhoz, gyakoribb és könnyebb a konstruktív interakció a csapatban, valamint általában kisebb csapatban jobban érzi magát az ember. Utóbbi nagyon igaz a csapatra.

Viszont a legfontosabb, hogy egy kis csapatnál a ráfordítási idő ugyanakkora feladatra kevesebb. (lásd 13. ábra)



13. ábra: Ráfordítás

### 2.4.2. Eljárások

A projekt során a SCRUM jól ismert eljárásait és folyamatait szeretnénk szem előtt tartani. Ezek az egyszerű eljárások fogják segíteni a mindennapi munkát a projekt életében.

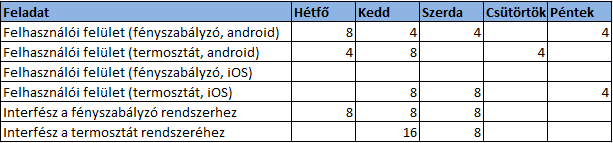
Egyik ilyen fontos fázis a futam (sprint) fázis, amikor a csapat 2-4 hetes munkaszakaszban dolgozik. Ebben a munkaszakaszban zajlik a termék tervezése, kódolása és tesztelése. A fázis végén, annak eredménye az üzleti értéket képviselő működő kód. Betartjuk azt az eljárást, miszerint a sprint fázis alatt nem lehet változás a feladatban.

Minden nap ugyanabban az időben a csapat a SCRUM mesterrel együtt egy rövid megbeszélést tart, ez az ún. napi SCRUM. Ezeknek a megbeszéléseknek a célja a haladást gátló akadályok meghatározása és nem feltétlenül azoknak a felszámolása. Ezek a megbeszélések egymásnak szólnak és nem a „team leader”-nek, vagyis egymás felé vállalunk kötelezettséget a munkánk során.

Következő folyamat a backlog létrehozása, amelyet a termékgazda végez. Összegyűjti az igényeket, szegmensekre bontja, majd priorizálja és az így kapott adatokból egy időben változó to-do listát készít. Egy sprint kezdeténél a to-do lista legelső elemeit befagyasztja és a SCRUM csapat elé terjeszti. Minden sprint kezdetén a priorizálás újra megtörténik. A termék backlog nem más, mint a követelmények és a projekt tervezett munkáinak listája.

A csapattagok választanak munkát maguknak a backlogból és nem kiosztják nekik, így mindenki olyan feladatot fog elvégezni, amelyet vélhetően szívesen választott. Bármelyik csapattag hozzáadhat a sprint backloghoz, törölhet belőle, változtathat rajta. A hátralevő munkát naponta újra kell becsülni, és ha egy munka nem világos, akkor több időt kell allokálni rá a backlogban és csak később lebontani kisebb elemekre.

A sprint backlogot táblázatos formában kezeljük, ahol a meghatározott munkák és az azokra szánt napi emberóra szerepel. (lásd 14. ábra)



14. ábra: Sprint backlog

Minden sprint fázist egy széles körű bemutatóval végződik, amelyen a termékgazda, a felhasználók és a menedzsment is részt vesz. Ezen a bemutatón kell demonstrálni, hogy mit haladt a csapat a sprint fázisban és annak milyen hatása van a következő sprint fázisra. Tehát ez a bemutató lesz az alapja a csapat kiértékelő, valamint a következő sprintindító megbeszélésének.

### 2.4.3. Projektdokumentáció

***Helyzet- és szükségfelmérés***

*1. Problémafa*

Utólag bonyolult beszerelni, drága, nem tartjuk fontosnak

Hanyagság, régimódiság, figyelmetlenség, sok mindent kell észben tartani

Zárlatos eszközök, elavult eszközök, karbantartásra szoruló eszközök, hanyagság

Riasztók hiánya, kamerák hiánya, biztonsági őrök hiánya

Égve hagyott lámpa, bekapcsolva maradt fűtés, magas vízfogyasztás

Elektromos tüzek, gázrobbanás, égve felejtett gyertya

Növekvő lakástűz esetek, magas rezsiköltség, betörések számának növekedése

1. *Célfa*

Lakástüzek megelőzése, rezsi csökkentése, magasabb vagyonvédelem

Égve hagyott lámpákra való figyelmeztetés, automatikus fűtésszabályozás, takarékos vízhasználat

Fejlett tűzjelzők, karbantartott modern elektromos rendszerek, szén-monoxid- és gázjelzők

Profi szakemberek által utólag is könnyen beépíthető informatikai megoldások

Automatikus rendszerek, a legújabb technológiák használata

Fejlett riasztórendszerek, mozgásérzékelők, kamerarendszerek éjjellátó kamerákkal

Állandó karbantartás, távvezérelt eszközök, távellenőrzés

1. *Stakeholder elemzés*

A projektben közvetlenül vagy közvetve érintett személyek, szervezetek:

* **Ügyfél**: Érdeke, hogy minél hamarabb, minél olcsóbban elérje a projekt a célját, azaz megkapja a terméket. Az állandó konstruktív kommunikáció elengedhetetlen. A projekt végső kedvezményezettje.
* **Menedzsment**: Közvetve érdekelt a célok elérésében. Ők finanszírozzák a projektek, amíg az ügyfél nem fizette ki a munkát. A pénzügyi kérdésekről is ők döntenek.
* **SCRUM csapat:** Aktív résztvevői a projektnek, ők fejlesztik le a terméket. Érdekeltek a célok elérésében.
* **SCRUM mester:** Közvetlenül befolyásolhatja a projekt megvalósítását, érdeke a célok elérése.
* **Termékgazda:** Közvetve befolyásolhatja a projekt megvalósítását, és érdekelt a projekt sikerességében.
* **Partnercégek:** Közvetlenül befolyásolhatják a projekt megvalósítását, mivel az általuk beszállított eszközökre fejlesztjük a terméket. A projektben részt vesznek és közvetlenül érdekeltek a célok sikeres elérésében.
* **Konkurencia:** Nem érdekük a projekt célok elérése, a projekt végső kedvezményezettjeit szeretnék megszerezni és közvetve befolyásolhatják a projekt megvalósulását.

***Tevékenység, idő és erőforrás tervezés***

#### Tevékenység fa és WBS diagram

TEVÉKENYSÉG

ARCH fejlesztés

Beszerzés

Tervezés

Helyzetfelmérés

Licenszek

Architektúra

Piaci igények

Komponensek

Funkcionális megfelelés

Irodai bútorok

Felhasználói felület(ek)

Fejlesztő eszközök (számítógépek, stb.)

Konkurencia

Demó ház

Prototípus hardverek, modulok

A tevékenységekhez később felelősök is rendelhetők. Ebben segít a WBS szerkezete. A táblázatban a tevékenységek kerülnek felsorolásra a tervezett időbeli költségükkel.

|  |  |  |
| --- | --- | --- |
| WBS | Tevékenység | Idő (óra) |
| 1 | **Helyzetfelmérés** | **350** |
| 2 | **Szoftverfejlesztés** | **1500** |
| 2.1 | Tervezés | 250 |
| 2.2 | Fejlesztés | 900 |
| 2.3 | Tesztelés | 150 |
| 2.4 | Hibajavítás | 200 |
| 3 | **Ügyféltámogatás (2év)** | **500** |
| 3.1 | Rendszertelepítés | 100 |
| 3.2 | 7/24 ügyelet | 300 |
| 3.3 | Hibajavítás | 100 |
| 4 | **Ügyféllel kapcsolattartás** | **100** |
| 5 | **Workshop-ok** | **300** |
| 6 | **Licenszek** | **200** |
| 7 | **Projekt menedzsment** | **350** |
| Összesen |  | **3300** |

#### Mérföldkövek

Az összetettsége a projektnek megköveteli, hogy mérföldköveket is meghatározzunk ahhoz, hogy a fontos határpontokat a projekt megvalósítása során jelezhessük. Ilyen jellegű határpontok lehetnek azok az időpontok, amikor meg szeretnénk nézni, hogy hogyan is halad a projekt, milyen állapotban van.

|  |  |  |
| --- | --- | --- |
| Mérföldkő | Leírás | Időpont |
| 1. | Helyzetfelmérés kiértékelés | 2013.07.01. |
| 2. | Fejlesztés megkezdése | 2013.09.15. |
| 3. | Állapotjelentés | 2013.11.02. |
| 4. | Tesztelés kiértékelés | 2014.01.30. |
| 5. | Állapotjelentés | 2014.02.20. |

#### Gantt-diagram

Nagyméretű projektről van szó, ezért nem könnyű áttekinteni az összefüggéseket a különböző tevékenységek és azok időbeli és erőforrásbeli vonzatuk között. A projekt ugyan elég nagy, viszont a tevékenységek száma nem olyan nagy, így nem kell a bonyolultabb hálótervezési módszereket alkalmaznunk, mint például a CPM vagy a PERT.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tevékenység** | **Idő** | | | | | | | | | | | |
| **2013** | | | | | | | | **2014** | | | |
| 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 01 | 02 | 03 | 04 |
| Helyzetfelmérés |  |  |  |  |  |  |  |  |  |  |  |  |
| Workshop-ok |  |  |  |  |  |  |  |  |  |  |  |  |
| Licenszek |  |  |  |  |  |  |  |  |  |  |  |  |
| Üzleti megbeszélés |  |  |  |  |  |  |  |  |  |  |  |  |
| Partnerek |  |  |  |  |  |  |  |  |  |  |  |  |
| Tervezés |  |  |  |  |  |  |  |  |  |  |  |  |
| Fejlesztés |  |  |  |  |  |  |  |  |  |  |  |  |
| Tesztelés |  |  |  |  |  |  |  |  |  |  |  |  |
| Átadás |  |  |  |  |  |  |  |  |  |  |  |  |

### 2.4.4. Minőségbiztosítás

#### 2.4.4.1. Követelménykezelési terv.

|  |  |  |  |
| --- | --- | --- | --- |
| Ki készíti? | Mi a tartalma? | Ki ellenőrzi a tervezetet? | Ki és hogyan ellenőrzi a végrehajtást? |
| Gyimesi Gábor | Az adott termékkel szemben felmerülő követelmények (akár belső, akár külső forrásból érkezik) kezelése, azaz, hogyan biztosítja majd vállalatunk azt, hogy a termék megfelel a követelményeknek. | A tervezési osztály. | A minőségbiztosítási osztály végzi, a végrehajtás alatt állapotjelentések kérésével, a tervezet végrehajtása után szigorú minőségellenőrzési eljárásokkal. |

**Hogyan kezelünk egy esetleges követelményváltozást?**

Egy termékkel szemben felmerülő követelményváltozás esetén a következő eljárás kerül végrehajtásra:

1. Igénybejelentés. Az igénybejelentések a Támogatói Részleghez érkeznek és innen továbbítódnak a megfelelő osztályokhoz, jelen esetben a Tervezői Osztályhoz.
2. Felmérés. Mekkora változtatást igényel a terméken? Szükséges-e a megrendelővel konzultálni? Végrehajtó: tervezési osztály vezetése.
   1. A hatékony ügyintézés érdekében a felmérés 2 napnál többet nem vehet igénybe.
3. Jóváhagyás. Új követelmény beépítése a termék specifikációjába. Végrehajtó: tervezési osztály vezetése.
   1. Kisméretű változás esetén specifikáció kiegészítése.
   2. Ellenkező esetben újratervezés.
4. Érdekeltek értesítése. Az új követelmény fejlesztése és tesztelése. Végrehajtó: fejlesztési + tesztelési osztály.
   1. Mindennemű változtatás esetén mély, teljes tesztelés, hogy a korábbi funkciók pontos működése is garantálva maradjon.

#### 2.4.4.3. Konfiguráció és változáskezelési terv.

|  |  |  |  |
| --- | --- | --- | --- |
| Ki készíti? | Mi a tartalma? | Ki ellenőrzi? | Ki és hogyan ellenőrzi a végrehajtást? |
| Bodnár István | Teendők listája egy esetleges hardware és/vagy software változás esetére. | Tervezési osztály |  |

**Konfiguráció megváltozásának kezelése**

1. Igénybejelentés.
   1. Ez indoklást is tartalmaz, legyen szó a cégen belüli változás (elromlott/elavult hardware, elavult software, új biztonsági mentések készítéséhez új software/hardware stb.), vagy egy külső felhasználó esetében.
2. Jóváhagyás. Új software/hardware beszerzésének jóváhagyása.
   1. Csak a hiba/elavultság megerősítése után.
      1. Ennek ellenőrzése nem vehet igénybe többet fél napnál.
      2. Software hiba esetén csak külsős esetben szükséges megerősítés, cégen belül feltételezzük, hogy a fejlesztők tudják, mikor hibás egy software.
      3. Hardware hiba esetén külsős esetben az azt gyártó cég küld ki egy embert a hiba megerősítésére és elhárítására.
      4. Hamis igénybejelentés esetén, cégen belüli esetben szankciók.
3. Felmérés. Milyen változtatás szükséges és mekkora költséggel jár ez?
   1. Magas költség esetén külön egyeztetések a vezetőséggel.
4. Beszerzés.
   1. A beszerzés prioritások szerint történik és elsősorban a vállalattal szerződésben álló cégektől történik. Egyéb forrásokból történő beszerzés csak indokolt esetben.
5. Érdekeltek értesítése. A beszerzett software/hardware üzembe helyezése.
   1. Software esetén tesztrendszerben történő kipróbálás után.
   2. Hardware-t csak alapos stressz teszt után.

#### 2.4.4.3. Kommunikációs terv

|  |  |  |  |
| --- | --- | --- | --- |
| Ki készíti? | Mi a tartalma? | Ki ellenőrzi? | Ki és hogyan ellenőrzi a végrehajtást? |
| Balog Szabolcs Dávid | Projekt kommunikáció módja. | Fejlesztési osztály. | A legfelsőbb vezetés. |

#### Hogyan kommunikálunk a projektben?

1. Agilis fejlesztés történik, azon belül a csapatok SCRUM-ot használnak.
   1. 2-3 hetes sprintek.
   2. Minden másnap SCRUM meeting a projekt előrehaladásának követése érdekében.
   3. Sprint tervezési fázisok minden hét végén.
   4. Hét végi sprintjelentés.
2. A projekten belül a csapatok számára havi 2 „nagy” megbeszélés, ahol a csapat tisztázza az elért eredményeket, ezek kerülnek a vezetőség felé átadandó állapotjelentésekbe.
   1. Az állapotjelentés tartalma:
      1. A csapat és a projekt neve, kódja, amin dolgoznak.
      2. A projekt/csapatvezető neve.
      3. A kitűzött célok állapota (késés esetén becsült idő).
      4. A felmerült problémák pontos leírása.

#### 2.4.4.4. Kockázatkezelési terv

|  |  |  |  |
| --- | --- | --- | --- |
| Ki készíti? | Mi a tartalma? | Ki ellenőrzi? | Ki és hogyan ellenőrzi a végrehajtást? |
| Hangay Zsolt | Felmérés a projekt kockázatairól, utasítások egy esetleges új vagy módosult kockázat kezelésére. |  |  |

#### 2.4.4.5. Tervezési útmutató

A tervezés során az objektumorientált programozási paradigmát használjuk. Fontosabb lépések:

**Brainstorming a termékről.**

Főbb funkciókra helyezzük a hangsúlyt, a felületről is essék szó. Legyenek jelen: Tervezési Osztály vezetői, a cég felső vezetése, a Fejlesztési Osztály vezetői. Ez a lépés szükség esetén megismételhető: új funkció bevezetése esetén.

**A feladat komponensekre bontása.**

Ezt a feladatot Enterprise Architect-tel oldjuk meg. A Tervezési Osztály vezetői és a Fejlesztési Osztály vezetői együtt dolgoznak ezen a problémán, hogy a fejlesztés gördülékeny lehessen. Az eredmény egy komponensdiagram legyen.

**A komponensek osztályainak megtervezése.**

Erre a feladatra is az Enterprise Architect az erősen javasolt megoldást segítő eszköz. Eredmény a komponensek osztálydiagramjai.

#### 2.4.4.6. Implementációs/Programozási előírások

**1. Programnyelv**

A termékek fejlesztése C++/C# nyelven történnek a terméktől függően. A vállalat fő profilja a C#, de nagy hatékonyságú, platformfüggetlen, erősen gépközeli kódot igénylő program esetén C++ nyelvet használunk. A nyelv kiválasztásáért és betartásáért a Fejlesztési Osztály a felelős.

**2. Fejlesztőkörnyezet**

A GauSSoft dolgozói számára a Microsoft Visual Studio 2010 Professional szoftver áll rendelkezésre, ez a cég által licenszelt fejlesztőkörnyezet így a használata garantált. Vállalatunk a minél hatékonyabb fejlesztést helyezi előtérbe, a fent nevezett fejlesztőkörnyezet pedig kiváló hibakövetési és refaktorálási eszközökkel rendelkezik.

**3. Forráskódok elhelyezése**

A projekteknek rendelkezniük kell SVN Repository-val, ide kommitolhatják a megírt és unit teszteken átesett kódjaikat a dolgozók. A unittesztelésért minden dolgozó maga felel. Használata kötelező, ezért a programozó maga felelős, de a kommit számossága helyett a minőségi kód a prioritás. Erről a következő szekcióban olvashat többet.

**4. Kóddal szemben támasztott követelmények**

A Fejlesztési Osztályon dolgozók számára a „Clean Code” elv a követendő, a fejlesztők saját maga ellenőrzi a kódját, kommitolás után a többi dolgozó ír ún. code review-t ezzel elősegítve a kód jó minőségének megőrzését, elérését. Az elv elsajátítására cégünk minden dolgozó számára elérhető videók segítségével biztosít lehetőséget.

**5. Fejlesztési metodika**

A Fejlesztési osztályon lévő csapatok SCRUM-ban fejlesszenek 2 hetes sprinteket alkalmazva, 2-3 naponta esedékes megbeszélésekkel. Bővebb információ a Kommunikációs Terv részben olvasható.

#### 2.4.4.7. Tesztelési stratégia

**Fejlesztők által elvégzendő tesztek**

*Egységtesztek*

Az egységtesztekért minden fejlesztő maga felelős, így ahány egységtesztre alkalmas kódrészlet van a projektben, annyi ember szükséges a tesztek elvégzéséhez. Ezek végrehajtása kötelező és a rendelkezésre álló fejlesztőkörnyezettel jól automatizálható, ennek használata is kötelező.

**Rendszerteszt**

*Integrációs tesztek*

Az integráláshoz ún. „Bottom-Up” integrációt végzünk a tesztek implementálásának megkönnyítése végett és az így kapott kód kerül tesztelésre.

**Felhasználói tesztek**

*Bétatesztek*

A bétatesztelésből 3-at kötelező végrehajtani, ezeket a Fejlesztési Osztály választott emberei a Tesztelési Osztály vezetőségével karöltve felügyelik, ezzel garantálható, hogy minőségi program kerül majd kiadásra és az esetleges tervezéskor fel nem merült igények beépíthetőek a termékbe (ez persze az igény méretétől függ). Az első két bétatesztben külsős felhasználó még nem vehet részt, csak a Tesztelési Osztály dolgozói szám szerint 10. A 3. és egyben utolsó nyílt bétatesztben 100-150 felhasználó a várható, a program méretétől és milyenségétől függően. A bétatesztek közül az első kettő 2 hónapot, az utolsó 3 hónapot vesz igénybe.

*Végteszt*

A kiadás előtti utolsó nagy teszt, amit a felhasználók végeznek, a felügyeleti jog ugyanaz, mint a bétateszteknél ám a lélekszám nagyobb, itt már 250-300 ember teszteli a programot 1 hónapig. Az időtartam a 3. bétateszthez képest a program kevés változása és a megnövelt emberi erőforrás miatt csökken, hiszen nagy változás/hiba esetén a program nem kerülhet végtesztelésre.