

# Követelmény Menedzsment

Dr. Gergely Tamás

Szegedi Tudományegyetem  
Informatikai Intézet  
Szoftverfejlesztés Tanszék

2024

(v0910)



## A Kurzusról

- Bemutatók
- Az RE szerepe

## 1 Bevezetés

- Mit?
- Miért?
- Hol?
- Hogyan?
- Szerepkörök és feladatok
- Mit kell tudni?

## 2 Az RE alapelvei

- Az alapelvek áttekintése
- Az alapelvek elmagyarázva

## 3 Munkatermékek és dokumentáció

- Munkatermékek a követelménytervezésben
- Természetes nyelv alapú munkatermékek
- Sablon alapú munkatermékek
- Modell alapú munkatermékek
- Szójegyzékek
- Követelménydokumentum szerkezete
- Prototípusok
- Minőségi kritériumok

## 4 A követelménytervezés lépései

- A követelmények forrásai
- Követelmények feltárása
- Konfliktusfeloldás
- Követelmények validálása

## 5 Folyamatok és munka

- Befolyásoló tényezők

- RE folyamat szempontok
- Az RE folyamat konfigurálása

## 6 Követelmények menedzsmentje

- Mi az a követelménymenedzsment?
- Életciklus menedzsment
- Verziókövetés
- Konfigurációk
- Tulajdonságok és nézetek
- Nyomonkövethetőség
- Változások kezelése
- Priorizálás

## 7 Tool Support

- Eszközök a követelménytervezésben
- Eszközök bevezetése

- A szükséges dokumentáció mértéke függ
  - a fejlesztési modelltől,
  - a rendelkezésre álló időtől,
  - a „megrendelő” és „fejlesztő” kapcsolatától,
  - a különféle vonatkozó szabályoktól, szabályozástól,
  - a rendszer biztonsági elvárásaitól,
  - ...
- Nincs egy minden helyzetben tökéletesen működő módszertan/sablon
- A szükséges dokumentációt az adott projekt és körülmények függvényében kell meghatározni

## A Kurzusról

- Bemutató
- Az RE szerepe

## 1 Bevezetés

- Mit?
- Miért?
- Hol?
- Hogyan?
- Szerepkörök és feladatok
- Mit kell tudni?

## 2 Az RE alapelvei

- Az alapelvek áttekintése
- Az alapelvek elmagyarázva

## 3 Munkatermékek és dokumentáció

## • Munkatermékek a követelménytervezésben

- Természetes nyelv alapú munkatermékek
- Sablon alapú munkatermékek
- Modell alapú munkatermékek
- Szójegyzékek
- Követelménydokumentum szerkezete
- Prototípusok
- Minőségi kritériumok

## 4 A követelménytervezés lépései

- A követelmények forrásai
- Követelmények feltárása
- Konfliktusfeloldás
- Követelmények validálása

## 5 Folyamatok és munka

- Befolyásoló tényezők

- RE folyamat szempontok
- Az RE folyamat konfigurálása

## 6 Követelmények menedzsmentje

- Mi az a követelménymenedzsment?
- Életciklus menedzsment
- Verziókövetés
- Konfigurációk
- Tulajdonságok és nézetek
- Nyomonkövethetőség
- Változások kezelése
- Priorizálás

## 7 Tool Support

- Eszközök a követelménytervezésben
- Eszközök bevezetése

# Munkatermékek jellemzői

- Munkatermék: bármi ami valamilyen követelményt fejez ki
- A munkatermékek a következő szempontok szerint kategorizálhatók
  - Cél (*purpose*)
  - Méret (*size*)
  - Reprezentáció (*representation*)
    - **természetes nyelv, sablon, modell**, ábra, prototípus, ...
  - Élettartam (*lifespan*)
    - ideiglenes: kommunikációhoz, közös megértéshez; használat után megsemmisül, nincs követve
    - fejlődő: iterációkon keresztül növekszik, fejlődése meta-adatokon keresztül nagyjából követve van, a változtatása néha csak megadott eljárások szerint történhet
    - időtálló: release-elt, hivatkozott munkatermék, mindenféle metaadattal ellátva/nyilvántartva, csak szigorú folyamatok mentén változhat
    - ideiglenes → fejlődő → időtálló
  - Tárolás (*storage*)
    - elektronikusan fájlként, adatbázisban, RE eszközökben, ...
    - papíron, cetliken, táblán, ...

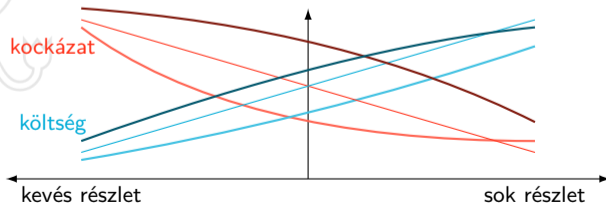
# Példa munkatermékek

- Egyedi követelmény: egy követelmény (általában szöveges) leírása (S)
- User story: funkció vagy viselkedés egy érdekelt fél szemszögéből (S)
- Use case: a rendszer funkciója a felhasználó szemszögéből (S-M)
- Grafikus modell: követelmények valamely vetületei (M)
- Feladatleírás: a rendszer feladatának leírása (S-M)
- Interfész leírás: a rendszer és egy aktor között cserélt információ leírása (M)
- Epic: Egy érdekelt fél igényeinek magasszintű leírása (M)
- Követelmény-specifikáció: a követelmények leírása (L-XL)
- Sprint backlog: Megvalósítandó követelmények listája (M-L)
- Story map: user story-k kapcsolatának grafikus megjelenítése (M)
- Szójegyzék: a használt kifejezések magyarázata (M)
- Prototípus: a követelmények példán keresztüli specifikációja (S-L)

- Magas szintű üzleti követelményektől alacsony szintű, nagyon részletes leírásokig sokféle lehet
- Tipikus szintjei
  - Üzleti szint: üzleti, iparági, felhasználói követelmények
  - Rendszer: magasabb szintű rendszerkövetelmények
  - Komponens: alacsonyabb szintű rendszerkövetelmények
  - *Lineáris fejlesztésnél általában egyenként, egymás után, egymásra épülve kerülnek meghatározásra*
  - *Iteratív fejlesztésnél általában egymással párhuzamosan fejlődnek*
- A megfelelő szint kiválasztása függ attól, hogy mit kell specifikálni
- Az egyedi követelmények szinteken keresztül tovább finomíthatók
- A különböző absztrakciós szintű követelményeket el kell különíteni
  - Kis és közepes munkatermékekben általában csak egy szint jelenik meg

# Követelmények részletessége

- Azt, hogy milyen részletességű legyen egy követelmény nehéz általánosan megfogalmazni, sok minden befolyásolhatja
  - nehezebb, speciálisabb probléma, kontextus → nagyobb részletesség
  - kevesebb közös megértés → részletesebb követelmények
  - nagyobb fejlesztői szabadság → kevesebb részlet
  - rövidebb visszajelzési ciklus fejlesztés alatt → kevesebb részlet
  - követelmény értéke nagyobb → „drágább”, részletesebb lehet
  - standardok és szabályozás előírhatnak részletesebb specifikációt, mint ami egyébként szükséges lenne
- Részletesebb követelmény = kisebb kockázat + nagyobb költség





- Az absztrakciós szint és a követelmények részletessége alapvetően független egymástól
  - Az üzleti követelmények feltárásának kezdetekor általában először nagyívű igényekkel találkozunk,
  - ezek kifejtésekor, finomításakor egyre több részlet fog előkerülni.
  - Egy komponens specifikációja általában örökli a rendszerspecifikáció részletességét,
  - ugyanakkor ha például a fejlesztők jól ismerik a kontextust, a komponens bizonyos saját követelményeit kevésbé kell részletezni.



# Szemponatok, amikre figyelni kell

- Le kell fedni
  - a funkcionális követelményeket,
  - a minőségi követelményeket, és
  - a megszorításokat is.
- A minőségi követelményeket gyakran adottnak tekintik, így elmaradnak
- A megszorítások gyakran funkcionális követelményként vannak megadva
- A megfelelő besorolásra oda kell figyelni!



- A funkcionális követelmények három fontosabb vetülete
  - Szerkezet és adatok
    - A rendszer és az általa kezelt adatok statikus szerkezete
  - Funkciók és folyamatok
    - A rendszer által nyújtott funkciók és az ezek közötti adatáramlás és vezérlés
  - Állapotok és viselkedés
    - A rendszer állapot-alapú viselkedése, milyen eseményekre hogyan reagál?



- Az ISO/IEC 25010:2011<sup>5</sup> szabvány jól használható ellenőrzőlistaként
  - performancia, hatékonyság, kompatibilitás, használhatóság, megbízhatóság, biztonság, (*pontosság*)
- A performancia kiemelten fontos
  - idő, adatmennyiség, gyakoriság, feldolgozási kapacitás, erőforráshasználat (CPU, memória, háttértár, hálózat, akkumulátor)
  - A minőségi jellemzőket is – ha csak lehet – mérhető jellemzőkkel kell megadni
    - Az átlagérték nem mindig elég, minimum és/vagy maximum, illetve 95%-os értékekre is szükség lehet
- A többi minőségi követelmény dokumentálása kemény dió

- A minőségi követelmények dokumentálásának módjai
  - **Qualitatív** reprezentáció: nem egyértelmű, szubjektív
  - **Quantitatív** reprezentáció: mérhető, nagy előny, ha sikerül, de néha nagyon nehéz és költséges lehet
  - **Funkcionalizált** reprezentáció: a minőségi célt funkciók hozzáadásával biztosítjuk, ez korai tervezési döntésekre kényszeríthet
- A „csak a kvantitatív minőségi követelmény jó” nem jó vezérelv
- A kvalitatív megadás is elfogadható
  - megfelelő implicit közös megértés esetén
  - ha a felek és fejlesztők megegyeznek egy ismert, kielégítő megoldásban
  - ha a felek csak irányelveket adnak és megbíznak a fejlesztőkben
  - rövid visszajelzési ciklusok esetén
- Megfelelő, általánosítható (kvantitatív) példák listája is elegendő lehet
- Csak nagy kockázat, különösen biztonságkritikus fejlesztés esetén kell mindent kvantitatív módon megadni

- A megszorítások lehetséges kategóriái:
  - Technikai: interfészek, protokollok, komponensek, keretrendszerek, ...
  - Jogi: törvények, szerződések, standardok, szabályozás, ...
  - Szervezeti: struktúra, folyamatok, policy-k, ...
  - Kulturális: különösen fontos a figyelembe vétele, ...
  - Környezeti: hőmérséklet, páratartalom, sugárzás, vibráció, energiafogyasztás, hőtermelés, ...
  - Fizikai: felhasznált anyagok, erőhatások, tehetetlenség, a fizika törvényei, ...
  - Egyedi, speciális: a fontosabb érdekelt felektől származó igények, ...



- „Kontextus és határ”: a követelmények csak a kontextusban érthetők
  - Szakterületi, iparági követelmények
  - A kontextus szakterületre, iparágra vonatkozó feltételezései
  - Külső interfészek
- A szempontok nem függetlenek
  - Egy külső felhasználói (kontextus) kérés valamely interfészen (határ) érkezik a rendszerbe, megváltoztatva annak állapotát (állapot) miközben a rendszer több lépésen (funkció) keresztül feldolgozza (flow) a megfelelő formában kapott adatot (adat) és határidőn belül (performancia) visszajuttatja a felhasználónak az eredményt
- Vannak munkatermékek, amik egy-egy szempontra koncentrálnak (pl. a modellek)
- Más munkatermékek több (akár az összes) szempontot lefedhetnek
- Figyelni kell arra, hogy a különböző szempontok szerint is konzisztens legyen a végeredmény

# Általános dokumentációs irányelvek

- Olyan munkaterméket válasszunk, amely megfelel a kitűzött céloknak
- Kerüljük a redundanciát, használjunk inkább hivatkozásokat
- Figyeljünk rá, hogy a munkatermékek konzisztensek legyenek
- A fogalmakat konzisztensen, a szójegyzék szerint használjuk
- Figyeljünk a munkatermékek megfelelő szerkezetére (pl. használjunk standardokat)





- Minden projekt, minden szakterület más és más
- Miben kell az RE-nek, az érdekelt feleknek és a projektvezetésnek megegyeznie?
  - Melyik munkatermékekben milyen célból dokumentáljuk a követelményeket?
  - Milyen absztrakciós szinte(ke)n dolgozzunk?
  - Az egyes absztrakciós szinteken mi az elvárt részletesség?
  - Milyen formában/módszerrel dokumentáljuk a követelményeket?
- A megfelelő munkatermékek korai definíciója
  - segíti az erőforrástervezést,
  - biztosítja a megfelelő jelölések használatát,
  - biztosítja, hogy minden eredmény a megfelelő helyen kerül rögzítésre,
  - segít, hogy ne kelljen utólag átszervezni/átszerkeszteni az elkészült anyagot,
  - segít kivédeni redundanciát.

## A Kurzusról

- Bemutató
- Az RE szerepe

## 1 Bevezetés

- Mit?
- Miért?
- Hol?
- Hogyan?
- Szerepkörök és feladatok
- Mit kell tudni?

## 2 Az RE alapelvei

- Az alapelvek áttekintése
- Az alapelvek elmagyarázva

## 3 Munkatermékek és dokumentáció

- Munkatermékek a követelménytervezésben

## • Természetes nyelv alapú munkatermékek

- Sablon alapú munkatermékek
- Modell alapú munkatermékek
- Szójegyzékek
- Követelménydokumentum szerkezete
- Prototípusok
- Minőségi kritériumok

## 4 A követelménytervezés lépései

- A követelmények forrásai
- Követelmények feltárása
- Konfliktusfeloldás
- Követelmények validálása

## 5 Folyamatok és munka

- Befolyásoló tényezők

- RE folyamat szempontok
- Az RE folyamat konfigurálása

## 6 Követelmények menedzsmentje

- Mi az a követelménymenedzsment?
- Életciklus menedzsment
- Verziókövetés
- Konfigurációk
- Tulajdonságok és nézetek
- Nyomonkövethetőség
- Változások kezelése
- Priorizálás

## 7 Tool Support

- Eszközök a követelménytervezésben
- Eszközök bevezetése

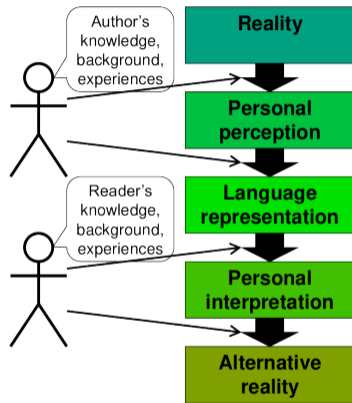
# A természetes nyelvű dokumentáció jellemzői

## • Előnyök

- Nincs szükség speciális ismeretekre a dokumentáláshoz
- Nagy kifejezőerő és rugalmasság
- Szinte bármilyen követelményt le lehet írni benne

## • Hátrányok

- Közvetlen, kontextusban történő, közös háttérismerettel rendelkezők közötti kommunikációra fejlődött ki
- Írásban hiányzik a gyors visszajelzés, a „hibák” azonnali detektálása
- Az írott szöveg ellenőrzése lassú, költséges



- Rövid, jól strukturált mondatok
  - ökölszabály: egy követelmény – egy mondat
  - *phrase template*-ek használata
- Jól strukturált munkatermékek
  - megfelelő fejezetek, alfejezetek
  - dokumentum–sablonok használata
- Egységesített terminológia konzisztens használata
  - szójegyzék
- Kerüljük a homályos, zavaros kifejezéseket
- Ismerjük és kerüljük a jól ismert csapdákat
- Ismerjük és lehetőleg kerüljük a buktatókat

- Csapdák: kerüljük őket!
  - hiányos leírások (*incomplete descriptions*)
  - általános elnevezések (*unspecific nouns*)
  - hiányos feltételek (*incomplete conditions*)
  - hiányos összehasonlítások (*incomplete comparisons*)
- Buktatók: körültekintéssel használjuk!
  - Szenvető forma (*passive voice*)
  - Általános névmások (*universal quantifiers*)
  - Főnevesítés (*nominalizations*)



# Hiányos leírások

- Az igéknek általában vannak paramétereik:
  - pl. „átad”: ki, mit, kinek?
- A követelményekben ezeket mind meg kell adni!
  - *Lekéri az adatokat a szerverről.*
  - **A modul** lekéri az adatokat a szerverről.



# Általános elnevezések

- Az általános elnevezések mindenféle pontosítás, hivatkozás nélkül túl tág teret adnak az értelmezésnek
  - a felhasználó, a rendszer, az adat, a funkció, ...
- A követelményekben adjunk hivatkozásokat ezekhez
  - *A modul lekéri az adatokat a szerverről.*
  - Az **időzítő** modul lekéri az **időzónára vonatkozó** adatokat a **Location** szerverről.



- Hajlamosak vagyunk csak a normális, rendes esetekre gondolni, és kihagyni a kivételes eseteket
  - ha ...akkor; amikor ...; amennyiben ...; ...
- A követelményekben figyeljünk az „egyébként” részekre is
  - *Ha van kapcsolódott Location szerver, akkor a modul lekéri az időzónára vonatkozó adatokat a Location szerverről.*
  - *Ha van kapcsolódott Location szerver, akkor a modul lekéri az időzónára vonatkozó adatokat a Location szerverről. Ha nincs kapcsolat Location szerverrel, akkor a modul UTC időzónát feltételezve működik tovább.*





- Élőszóban az összehasonlítás referenciája gyakran lemarad
  - néha a kontextusból egyértelmű
  - máskor nem is igazán összehasonlítást akarunk kifejezni vele
- Adjuk meg a viszonyítási, összehasonlítási alapot
  - **Korábbi időpontra nem lehet időzítést beállítani.**
  - **Az időzítés mentési időpontjánál** korábbi időpontra nem lehet időzítést beállítani.



- A szenvedő (angol nyelvben *passive*) forma elrejtheti a cselekmény alanyát, felelősét, részleteit
  - „A fa meg lett mászva.”
- Használjunk cselekvő (angol nyelvben *active*) igéket
  - Ha a Location szerver kapcsolódás sikeres, akkor az időzóna adatlekérés megtörténik.
  - Ha a Location szerver kapcsolódás sikeres, akkor az időzítő modul lekéri az időzónára vonatkozó adatokat a Location szerverről.



# Általános névmások

- Élőszóban az általános névmások gyakran olyan általánosításokat fejeznek ki, ami alól azért vannak kivételek
  - mindig, minden, soha, összes, ...
- A követelmények definíciója esetén ezeket a kivételeket pontosan meg kell adni, vagy
- az általános névmások helyett máshogyan kell specifikálni a követelményt.
  - **A felhasználói profil oldalon az aktuális felhasználó összes adata megjelenik.**
  - **A felhasználói profil oldalon a jelszó kivételével az aktuális felhasználó összes eltárolt adata megjelenik.**



- Amikor valamilyen eseményt, cselekményt névvel hivatkozunk, ez az elnevezés utalhat további (nem specifikált) követelményekre
- Ilyen esetekben az RE felelőssége, hogy a hivatkozott követelmények is dokumentálva legyenek
  - Ha a *Location szerver kapcsolódás* sikeres, akkor az időzítő modul lekéri az időzónára vonatkozó adatokat a Location szerverről.
  - *Location szerver kapcsolódás: ...*



## A Kurzusról

- Bemutatók
- Az RE szerepe

## 1 Bevezetés

- Mit?
- Miért?
- Hol?
- Hogyan?
- Szerepkörök és feladatok
- Mit kell tudni?

## 2 Az RE alapelvei

- Az alapelvek áttekintése
- Az alapelvek elmagyarázva

## 3 Munkatermékek és dokumentáció

- Munkatermékek a követelménytervezésben
- Természetes nyelv alapú munkatermékek

## • Sablon alapú munkatermékek

- Modell alapú munkatermékek
- Szójegyzékek
- Követelménydokumentum szerkezete
- Prototípusok
- Minőségi kritériumok

## 4 A követelménytervezés lépései

- A követelmények forrásai
- Követelmények feltárása
- Konfliktusfeloldás
- Követelmények validálása

## 5 Folyamatok és munka

- Befolyásoló tényezők

- RE folyamat szempontok
- Az RE folyamat konfigurálása

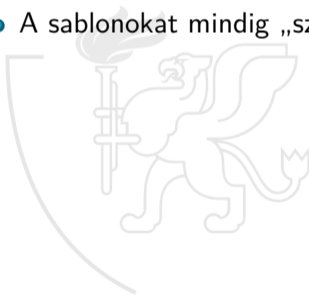
## 6 Követelmények menedzsmentje

- Mi az a követelménymenedzsment?
- Életciklus menedzsment
- Verziókövetés
- Konfigurációk
- Tulajdonságok és nézetek
- Nyomonkövethetőség
- Változások kezelése
- Priorizálás

## 7 Tool Support

- Eszközök a követelménytervezésben
- Eszközök bevezetése

- Természetes nyelvű dokumentációnál a sablonok nagyon hasznosak a különféle, természetes nyelv használatából eredő kockázatok csökkentésére
- A sablonok három szintjét különböztetjük meg
  - mondat sablon (*phrase template*)
  - űrlap sablon (*form template*)
  - dokumentum sablon (*document template*)
- A sablonokat mindig „személyre” (cégre, projektre) kell szabni használat előtt!



**Phrase template** *A template for the syntactic structure of a phrase that expresses an individual requirement or a user story in natural language.*

- A mondat sablonok olyan minták, amelyekből az RE a megfelelő helyek kitöltésével a követelményeket kifejező, jól strukturált, egységes mondatokat tud gyártani.
  - Sablon egyedi követelményekhez
  - Sablon user story-khoz



# Egyedi követelmény sablonok (angol nyelvhez)

- Az ISO/IEC/IEEE 29148:2011<sup>2</sup> szabvány ad egy általános mintát a követelményekre:  
[<Condition>] <Subject> <Action> <Objects> [<Restriction>].
- <Subject>
  - Kire vonatkozik a követelmény?
    - pl: the system <name>
- <Action>
  - Jelleg és tevékenység:
    - pl.: shall|should|may|will <activity>
    - shall: kötelező követelmény
    - should: erősen igényelt követelmény
    - may: javasolt követelmény
    - will: inkább tény rögzítése, mint követelmény



# Egyedi követelmény sablonok (angol nyelvhez)

- **<Objects>**
  - A tevékenység paraméterei és azok jellemzői
    - pl. az adat, amivel dolgozik, az interfész, amit használ, ...
- **<Condition>**
  - Időbeli és logikai feltételek
    - logikai pl: `if <cond>`
    - időbeli pl: `as soon as <cond>`
- **<Restriction>**
  - A tevékenység jellemzői
    - pl: pontosság, sebesség, szabályokból adódó kötelezvények, ...

# Egyedi követelmény sablon példák (angol nyelvhez)

- Általános forma:

The <system-name> shall <system-response>.

- Eseményvezérelt követelmény:

When [<preconditions>] <trigger> the <system-name> shall <system-response>.

- Elkerülendő szituáció:

If [<preconditions>] <trigger>,  
then the <system-name> shall <system-response>.

- Állapotvezérelt követelmény:

While <in-a-specific-state> the <system-name> shall <system-response>.

- Opcionális követelmény:

Where <feature-is-included> the <system-name> shall <system-response>.

# User story sablonok (angol nyelvhez)

- A klasszikus forma:

As a <role> I want <requirement> so that <benefit>.

- Minden user story mellé kell valahány elfogadási kritérium (*acceptance criteria*)
  - Ezek teszik konkrétabbá a user story-t
  - Az implementáció csak ezek teljesülése esetén lehet elfogadva



## Form template

*A template providing a form with predefined fields to be filled in.*

- Általában közepes méretű munkatermékekhez használják ezeket
  - használati esetek
  - feladatléírások
  - mérhető minőségi követelményekhez



# Ürlap sablon példa

Használati esethez

Id	Egyedi azonosító
Name	Egy rövid kifejezés
Precondition	A használati eset indításának előfeltétele(i)
Success end condition	Sikeres végrehajtás utáni állapot leírása
Failed end condition	Sikertelen végrehajtás utáni állapot leírása
Primary actor	Az elsődleges szereplő megnevezése
Other actors	További aktorok megnevezése(i)
Trigger	Az esemény, ami elindítja a leírt folyamatot
Normal flow	A fő folyamat leírása lépések sorozatával
Alternate flows	Az alternatív és kivételes folyamatok leírásai, hivatkozva a fő folyamat lépéseire
Extensions	A fő folyamat kiegészítései, hivatkozások az eredeti lépésekre
Related information	Egyéb információ (performancia, gyakoriság, referenciák, ...)

# Használati eset példa

1/2

Id	UC-8577-001
Name	Set alarm clock
Precondition	Phone is switched on, Clock application has been started
Success cond.	Alarm is set, alarm icon is shown in the main screen of the phone
Failed cond.	No new alarm is set
Primary actor	User
Other actors	Location server
Trigger	The user selects „New Alarm” in the Clock application
Normal flow	<ol style="list-style-type: none"><li>1) User sets the time of the alarm → <b>UC-8577-008 User input</b></li><li>2) User sets the date of the alarm → <b>UC-8577-008 User input [NO DATE SET] [CANCEL]</b></li><li>3) User presses OK button <b>[TIME BEFORE] [CANCEL]</b></li><li>4) Clock application asks the Location server about daylight saving data in the current location</li><li>5) Location Server answers with daylight saving data <b>[TIMEOUT]</b></li><li>6) Clock application checks if daylight save changing is involved <b>[DAYLIGHT SAVE]</b></li><li>7) Clock application shows the remaining time until the alarm fires to the user</li><li>8) Clock application arms the new alarm and puts the firing date and time to the armed alarms list</li></ol>

Alternate flows	<p><b>[NO DATE SET]</b> – <i>&lt;User does not change the date of the alarm&gt;</i>: Step 2 is skipped UC continues from step 3 date is the date of the actual day.</p> <p><b>[CANCEL]</b> – <i>&lt;User presses the Cancel button&gt;</i>: No alarm is set. UC ends.</p> <p><b>[DAYLIGHT SAVE]</b> – <i>&lt;There will be a time change because of daylight saving if the timezone of the user&gt;</i>: Alarm firingtime is adjusted to the new time → <b>UC-8577-005 Adjust time</b>. Clock application notifies the user about the change. UC continues from step 7.</p> <p><b>[TIME BEFORE]</b> – <i>&lt;The set time is before the actual time&gt;</i>: OK button changes to inactive. UC continues from step 1.</p> <p><b>[TIMEOUT]</b> – <i>&lt;Location Server does not answer within 1s&gt;</i>: Clock application does not consider daylight saving. Step 6 is skipped. UC continues from step 7.</p>
Related information	QR.004 – Reaction time upon user input

# Ürlap sablon példa

Mérhető minőségi követelményekhez

Id	A követelmény egyedi azonosítója
Goal	A cél kvalitatív módon megadva
Scale	Milyen skálán, mértékegységgel mérjük a követelményt?
Meter	Hogyan mérjük a követelményt, mi a mérési módszer?
Minimum	A legkisebb elfogadható minőség, ha nem éri el, akkor bukik
Ok range	A megcélzott intervallum, nem tökéletes, de elfogadható
Desired	A legjobb esetre vonatkozó elvárt minőség

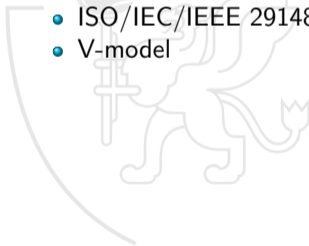




## Document template

*A template providing a predefined skeleton structure for a document.*

- Ezek a sablonok segítenek megszerezni az egyes követelményeket
- A standardok gyakran nyújtanak mintákat
- Néhány standard sablon példaként
  - IEEE 830-1998 standard
  - ISO/IEC/IEEE 29148:2011
  - V-model



- Kifejezetten a szoftver követelmények dokumentálásához tervezték
- A standard három fő fejezet használatát javasolja
  - Bevezető (Introduction)
  - Általános leírás (General description)
  - Követelmények (Specific requirements)
    - pl. funkcionális követelmények, interfészek, performancia, . . .



- Háromféle dokumentációs sablont definiál:
  - Magas szintű követelmények (Stakeholder Requirements Spec. – StRS)
  - Rendszerkövetelmények (System Requirements Spec. – SyRS)
  - Szoftver követelmények (Software Requirements Spec. – SRS)
- A dokumentumokban közös fejezetek
  - Bevezetés (Introduction)
  - Referenciák (References)
  - Követelmények leírása
    - Üzleti (Business management/operational)
    - Felhasználói (user)
    - Rendszer (system)
    - Specifikus (specific)
  - Ellenőrzés (Verification)
  - Függelék (Appendix)

- Különbféle struktúrákat definiál attól függően, ki a követelmények szerzője
  - Ügyfél követelmény-specifikáció (Customer requirement specification)
    - Az ügyfél készíti
    - A szerződő fél összes igényét rögzíti a szerződés tárgyával kapcsolatban
  - Rendszer követelmény-specifikáció (System requirement specification)
    - Az ügyfél követelmény-specifikációban megadott követelmények és megszorítások finomítása



## I. Bevezető

1. A rendszer célja (*purpose*)
2. A fejlesztés hatóköre
3. Érdekeltek felek

## II. Rendszer áttekintése

1. A rendszer víziója és céljai (*goals*)
2. A rendszer kontextusa és határai
3. A rendszer felépítése
4. Felhasználók jellemzői

## III. Rendszerkövetelmények

A rendszer felépítése szerint hierarchikusan szervezve, a követelmények hierarchikus számozásával

## • Alrendszerenként / komponensenként:

- Funkcionális követelmények (szerkezet és adatok, funkciók és folyamatok, állapotok és viselkedés)
- Minőségi követelmények
- Megszorítások
- Interfészek

## • Referenciák

## • Függelékek

- Szójegyzék (ha nem önálló munkatermék)
- Feltevések és függőségek

- Előnyök

- Segít jó minőségű követelmények, specifikációk előállításában, csökkenti egyes hibák kockázatát
- Az RE a lényegre koncentrálhat
- Könnyen újrahasznosítható követelmények
- Könnyebben validálható dokumentumok/követelmények

- Hátrányok

- Mechanikus használat veszélye: ha az RE a tartalom helyett a formára koncentrál, a sablonba nem illő követelmények kimaradhatnak



## A Kurzusról

- Bemutató
- Az RE szerepe

## 1 Bevezetés

- Mit?
- Miért?
- Hol?
- Hogyan?
- Szerepkörök és feladatok
- Mit kell tudni?

## 2 Az RE alapelvei

- Az alapelvek áttekintése
- Az alapelvek elmagyarázva

## 3 Munkatermek és dokumentáció

- Munkatermek a követelménytervezésben
- Természetes nyelv alapú munkatermek
- Sablon alapú munkatermek
- **Modell alapú munkatermek**
- Szójegyzékek
- Követelménydokumentum szerkezete
- Prototípusok
- Minőségi kritériumok

## 4 A követelménytervezés lépései

- A követelmények forrásai
- Követelmények feltárása
- Konfliktusfeloldás
- Követelmények validálása

## 5 Folyamatok és munka

- Befolyásoló tényezők

- RE folyamat szempontok
- Az RE folyamat konfigurálása

## 6 Követelmények menedzsmentje

- Mi az a követelménymenedzsment?
- Életciklus menedzsment
- Verziókövetés
- Konfigurációk
- Tulajdonságok és nézetek
- Nyomonkövethetőség
- Változások kezelése
- Priorizálás

## 7 Tool Support

- Eszközök a követelménytervezésben
- Eszközök bevezetése

## Model

*An abstract representation of an existing part of reality or a part of reality to be created. The notion of reality includes any conceivable set of elements, phenomena, or concepts, including other models.*

- A természetes nyelvű dokumentációnak vannak korlátai
  - Természetes többértelműség
  - Személyes interpretáció
  - A „nagy kép” elvesztése
    - Sok követelmény
    - Komplex követelmények
    - Követelmények közötti összefüggések
    - Absztrakció
  - Elsőre nem biztos, hogy sikerül teljesen megérteni
- A modellek segíthetnek:
  - Egyértelműek
  - Könnyebben átláthatók
  - Gyorsan megérthetőek



**REQ001:** The customer places an order

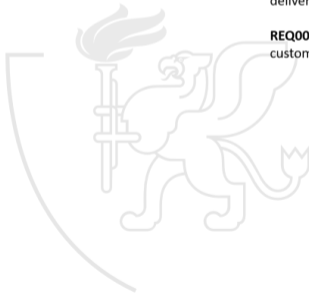
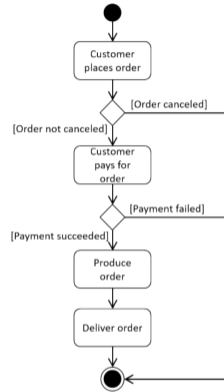
**REQ002:** The customer pays for the order

**REQ003:** If the payment succeeded, than the order is produced

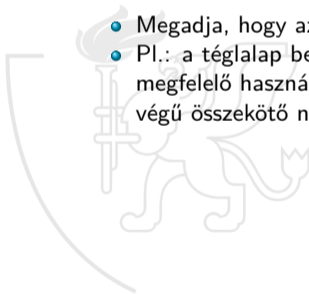
**REQ004:** If the payment failed, then the order is canceled

**REQ005:** After the order is produced it is delivered to the customer

**REQ006:** The order can be canceled if the customer does not pay for it



- A modellező nyelv is egy nyelv (még ha főként grafikus is)
  - Szintaxis
    - Megadja, hogy a nyelvben milyen elemeket használhatunk, és azokat hogyan kapcsolhatjuk egymáshoz
    - Modellező nyelv esetén a használható grafikai elemeket, pl.: téglalap benne más elemekkel, ellipszisek felirattal, összekötő vonalak dobozok között, összekötő nyilak háromszög fejjel, stb.
  - Szemantika
    - Megadja, hogy az egyes elemeknek, elemkombinációknak mi a jelentése
    - Pl.: a téglalap benne más elemekkel a rendszer határát jelzi, az ellipszisek a feliratnak megfelelő használati esetet jelölik, egy sima összekötő vonal asszociációt jelöl, a háromszög végű összekötő nyíl öröklődést jelöl, stb.

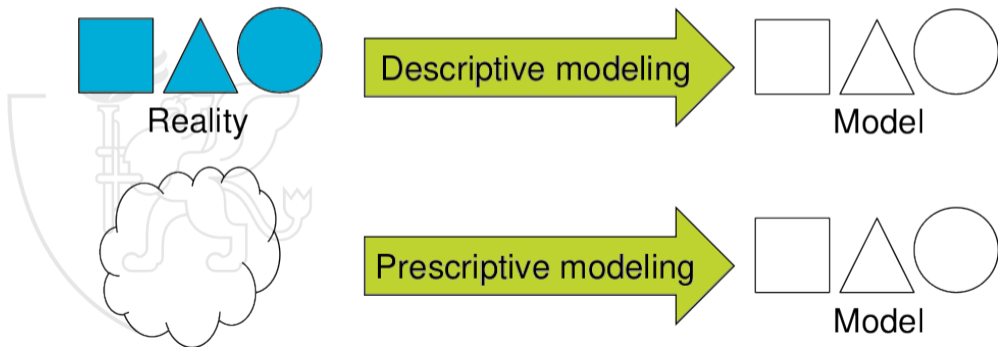


# Modellek és tulajdonságaik

- A követelménymodellek olyan fogalmi (*conceptual*) modellek, melyek a következő három tulajdonsággal rendelkeznek
  - Reprezentációs tulajdonság
  - Redukciós tulajdonság
  - Gyakorlatias tulajdonság
- A modellező nyelv lehet standardizált, formális modellező nyelv, mint amilyen például a UML (*Unified Modelling Language*)



- A modell mindig a létező vagy elképzelt valóság absztrakt reprezentációja
  - A leíró (*descriptive*) modellek mindig az aktuális valóságot, a már teljesült követelményeket írják le („eredeti” modell)
  - A perspektív (*perspective*) modellek az elképzelt, elvárt valóságot, az ezután teljesítendő követelményeket írják le

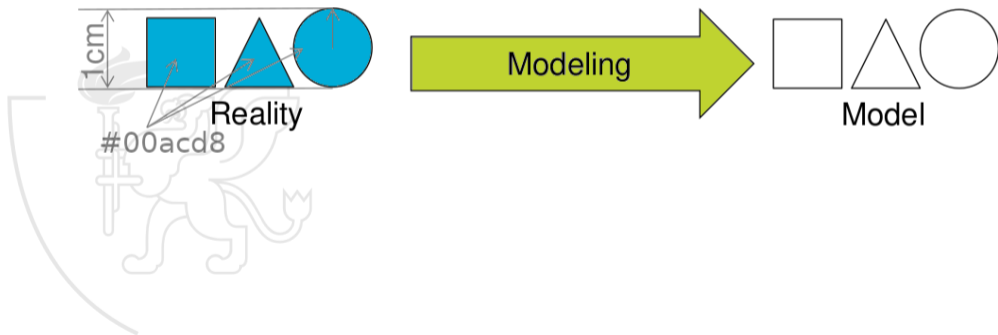


# Redukciós tulajdonság

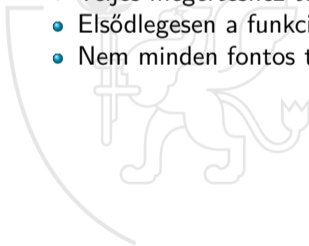
- A modellek mindig a valóság egy egyszerűsített változatát írják le (ez segít a lényegre koncentrálni, könnyebben megérteni a valóságot)
  - Kiválasztással (*selection*) a valóságnak csak egy részét, bizonyos elemeit modellezzük („széltében”)
  - Tömörítéssel (*compression, aggregation*) a valóságnak csak bizonyos részleteit, jellemzőit modellezzük („mélységben”)



- A modellek mindig egy speciális célból/célra készülnek, a valóságot mindig a megadott cél érdekében, annak megfelelően modellezik
  - Az egyes modellek csak a megfelelő célból alkalmazhatók
  - A modell csak az adott cél érdekében szükséges információkat tartalmazza, de azokat tartalmazza



- A modellek előnyei
  - Könnyebb érthetőség (kép vs. szöveg)
  - Fókuszált, egyféle nézet/aspektus
  - Egyértelmű (szigorú szintaktika és szemantika)
  - Nagyobb fokú automatikus ellenőrzés
- A modellek hátrányai
  - Konzisztencia megtartása (különböző nézetek között)
  - Teljes megértéshez több modellt át kell nézni
  - Elsődlegesen a funkcionális követelményekre koncentrálnak
  - Nem minden fontos tényező/információ fejezhető ki a modelleken



- Általános/speciális modellek
  - Általános, a kontextusban is használható modellek
  - Speciális, az adott kontextusra kifejlesztett modell
- Standardizált/egyedi modellek
  - Standardizált: „közismert” szintaxis és szemantika (pl. UML, BPML)
  - Egyedi: a szintaktikát és szemantikát magyarázni kell az olvasónak
- Rendszerfejlesztés követelményeinél használt vetületek (*perspectives*)
  - Szerkezet és adatok
  - Funkciók és folyamatok
  - Állapotok és viselkedés
- A rendszer jellege segíthet a megfelelő modell kiválasztásában, de alapvetően a vetületek kombinációját kell alkalmazni
- A követelmények feltárása előtt a magasszintű célokat is lehet modellezni



- Mire jók a modellek?
  - Követelmények specifikálására, akár a szöveges követelmények teljes kiváltásával
  - Egy komplex valóság szétbontásával (specifikus modellekkel) segít a megértésben
  - A szöveges követelmények „átfogalmazásával” javítja azok érthetőségét
  - Segítenek a szöveges követelmények validálásában, felfedve azok hiányosságait, kétértelműségeit, következetlenségeit
  - Segítenek a tudás rendszerezésében, a rendszer és a kontextus jobb megismerésében

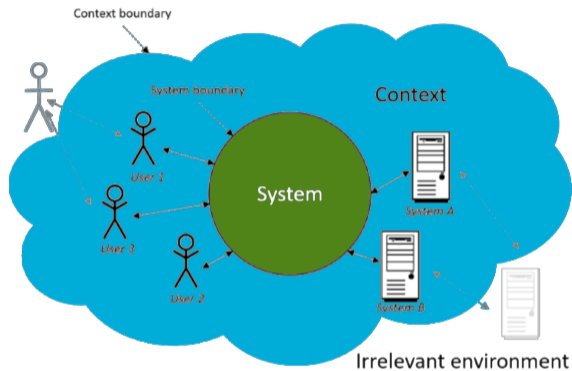


- Szintaktikus minőség
  - A modell mennyire felel meg a szintaktikai szabályoknak?
  - Vannak-e a modellben rosszul használt elemek?
  - Vannak-e a modellben (szintaktikailag) extra elemek?
- Szemantikus minőség
  - Egy elem, diagramm vagy modell mennyire korrekten reprezentálja a tényeket?
  - Mennyire lehet félreérteni a modellt és elemeit?
- Gyakorlatias minőség
  - Egy adott elem, diagramm vagy modell mennyire alkalmas a kitűzött célok elérésére?

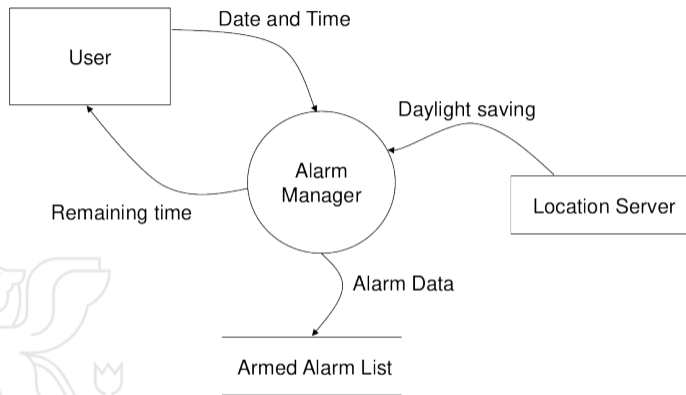
- Mindkét reprezentációnak vannak előnyei és hátrányai
- Együttes használatuk eltüntetheti a hátrányokat
- A modellek kiegészíthetők szöveges leírásokkal, pl. magyarázatokkal vagy minőségi követelményekkel
- Szöveg alapú követelmények esetén is használhatók ábrák, modellek egyes követelmények magyarázatához



- Hogyan ágyazódik be a rendszer a kontextusba?
  - Felhasználói interakciók
  - Interakciók más rendszerekkel
- Nem követelményeket definiál, hanem forrásokat segít beazonosítani
- Használható
  - Data flow diagrammok
  - UML use case diagrammok
  - Doboz-vonal diagrammok



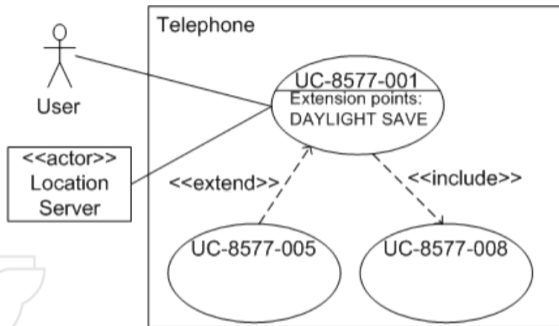
# Data Flow Diagram



- Rendszerelem / funkció
- Forrás / nyelő

- Belső tároló
- Adatáramlás

# UML Use Case Diagram



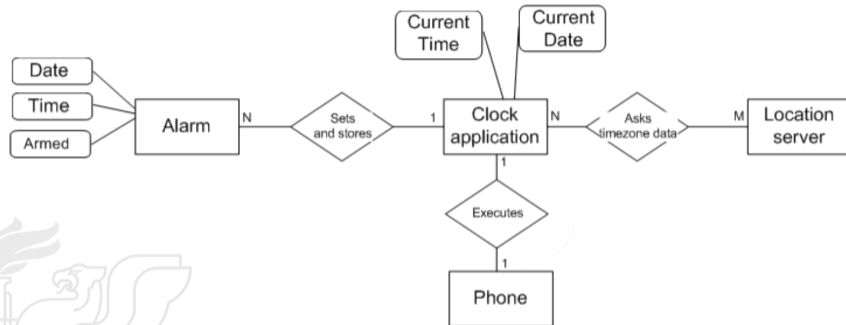
- Rendszer határa
- Használati eset
- Aktorok

- Bővítés (feltételes)
- Kiegészítés (feltétlen)
- Kommunikáció

- A valóság entitásainak és azok kapcsolatainak, összefüggéseinek modellezése
  - Milyen adatokkal, azok milyen jellemzőivel dolgozik a rendszer?
  - Ezek hogyan kapcsolódnak, viszonyulnak egymáshoz?
- Használható
  - Entity Relationship diagrammok
  - UML Class diagrammok
  - SysML Block Definition diagrammok
  - Building Information Models (BIM)



# Entity Relationship Diagram

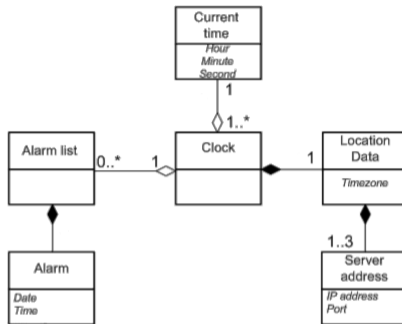


- Egyed
- Reláció

- Attribútum
- Számosság



# UML Class Diagram



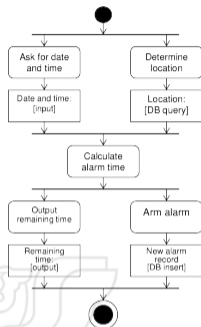
- Asszociáció
- Név / Szerep
- Multiplicitás

- Osztály
- Aggregáció / Kompozíció
- Öröklődés

- Hogyan alakítja át a rendszer az input adatokat output adatokká?
  - Több szemszögből is modellezhető
- Használható
  - UML Use Case diagrammok
  - UML Activity diagrammok
  - UML Sequence diagrammok
  - Data Flow diagrammok
  - Domain Story Models



# UML Activity Diagram

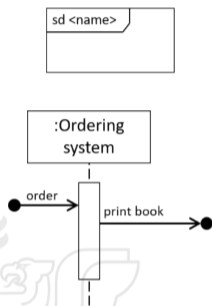


- Kezdő- / Végpont
- Akció / Tevékenység
- Control flow

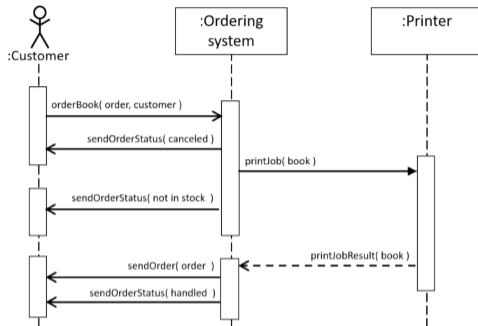


- Döntési / merge pont
- Object flow
- Szinkronizációs vonalak

# UML Sequence Diagram



- Aktor / Objektum
- Életvonal / Terminálás (×)
- Aktiváció
- Diagram keret

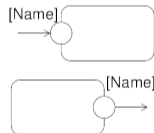
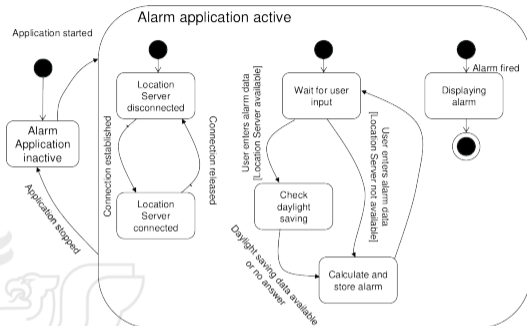


- Szinkron üzenet
- Aszinkron üzenet
- Válasz üzenet
- Külső üzenet

- Viselkedés, állapotok, átmenetek
  - Milyen állapotai vannak a rendszernek?
  - Milyen események hogyan hatnak a rendszerre?
- Használható
  - Statecharts
  - UML State diagrammok



# Statechart / UML State Diagram



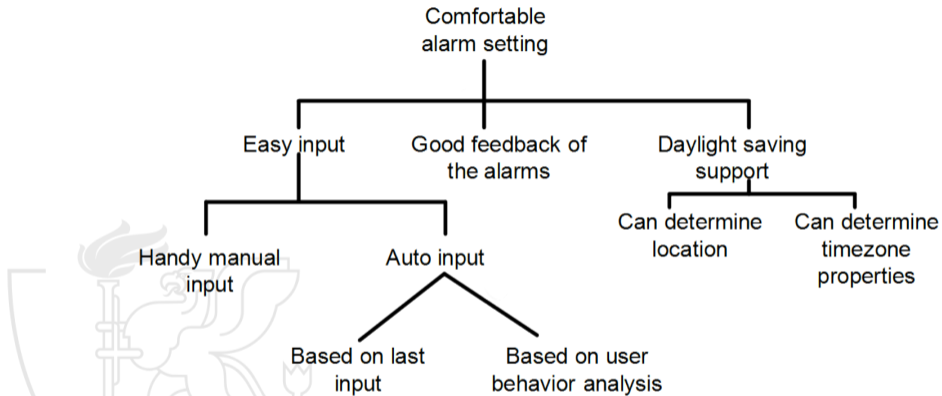
- Állapot
- Kezdő- / Végállapot
- Hierarchia
- Konkurencia

- Átmenet
- Esemény / Feltétel / Aktivitás
- Belépési pont
- Kilépési pont

- A magas szintű célokat, igényeket általában természetes nyelven szokás rögzíteni
- Az ilyen célok gyakran tovább bonthatók, finomíthatók
- Az így keletkezett célok kapcsolatát lehet akár modellekkel is ábrázolni
- Használható
  - And / Or Trees



# And / Or Trees



• És

• Vagy



## A Kurzusról

- Bemutató
- Az RE szerepe

## 1 Bevezetés

- Mit?
- Miért?
- Hol?
- Hogyan?
- Szerepkörök és feladatok
- Mit kell tudni?

## 2 Az RE alapelvei

- Az alapelvek áttekintése
- Az alapelvek elmagyarázva

## 3 Munkatermek és dokumentáció

- Munkatermek a követelménytervezésben
- Természetes nyelv alapú munkatermek
- Sablon alapú munkatermek
- Modell alapú munkatermek
- Szójegyzékek
- Követelménydokumentum szerkezete
- Prototípusok
- Minőségi kritériumok

## 4 A követelménytervezés lépései

- A követelmények forrásai
- Követelmények feltárása
- Konfliktusfeloldás
- Követelmények validálása

## 5 Folyamatok és munka

- Befolyásoló tényezők

- RE folyamat szempontok
- Az RE folyamat konfigurálása

## 6 Követelmények menedzsmentje

- Mi az a követelménymenedzsment?
- Életciklus menedzsment
- Verziókövetés
- Konfigurációk
- Tulajdonságok és nézetek
- Nyomonkövethetőség
- Változások kezelése
- Priorizálás

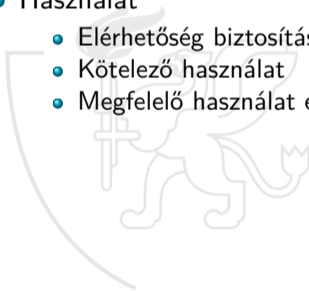
## 7 Tool Support

- Eszközök a követelménytervezésben
- Eszközök bevezetése

- Miért?
  - Egységes terminológiát, nevezéktant ad
  - Segít elkerülni a félreértéseket
- Mit tartalmaz?
  - Kontextus-specifikus fogalmakat
  - Mindennapi, de az adott környezetben speciális jelentőséggel bíró fogalmakat
  - Rövidítéseket
  - Betűszavakat
  - Szinonimákat
  - Homonimákat



- Létrehozás és karbantartás
  - Központilag menedzselte a teljes fejlesztés alatt
  - Egyszemélyi (max pár ember) felelős
  - Egységes stílus és szerkezet
  - Forrásmegjelölés
  - „Közös megegyezés” minden érintett fél bevonásával
- Használat
  - Elérhetőség biztosítása
  - Kötelező használat
  - Megfelelő használat ellenőrzése (a munkatermékekben)



## A Kurzusról

- Bemutató
- Az RE szerepe

## 1 Bevezetés

- Mit?
- Miért?
- Hol?
- Hogyan?
- Szerepkörök és feladatok
- Mit kell tudni?

## 2 Az RE alapelvei

- Az alapelvek áttekintése
- Az alapelvek elmagyarázva

## 3 Munkatermek és dokumentáció

- Munkatermek a követelménytervezésben
- Természetes nyelv alapú munkatermek
- Sablon alapú munkatermek
- Modell alapú munkatermek
- Szójegyzékek
- **Követelménydokumentum szerkezete**
- Prototípusok
- Minőségi kritériumok

## 4 A követelménytervezés lépései

- A követelmények forrásai
- Követelmények feltárása
- Konfliktusfeloldás
- Követelmények validálása

## 5 Folyamatok és munka

- Befolyásoló tényezők

- RE folyamat szempontok
- Az RE folyamat konfigurálása

## 6 Követelmények menedzsmentje

- Mi az a követelménymenedzsment?
- Életciklus menedzsment
- Verziókövetés
- Konfigurációk
- Tulajdonságok és nézetek
- Nyomonkövethetőség
- Változások kezelése
- Priorizálás

## 7 Tool Support

- Eszközök a követelménytervezésben
- Eszközök bevezetése

- Az egyedi követelmények nyilvánvalóan fontosak, de a követelmények összességét is megfelelő módon kell dokumentálni
- Nincs egy univerzális szabvány vagy struktúra, de a különféle dokumentumsablonok jól használhatók
- Hogyan válasszunk?
  - Nem szabad kérdés nélkül újrahasznosítani régi dokumentumszerkezeteket
  - Sok függ a fejlesztési folyamattól
  - A projekt típusát (egyedi eszközfejlesztés, „személyre szabás”, termékfejlesztés) is figyelembe kell venni
  - A szerződések, megállapodások előírhatnak valamit
  - A dokumentáció mérete befolyásolhatja a döntést

- Gyakori dokumentumtípusok, dokumentációs formák:
  - Stakeholder / Customer Requirements Specification
    - Az érdekelt felek / ügyfelek igényei a saját szempontjukból
  - User Requirements Specification
    - Az előző követelményhalmaz felhasználókat érintő részhalmaza
  - System Requirements Specification
    - Az előzőekből származó rendszerkövetelmények
  - Business Requirements Specification
    - Magas szintű üzleti célok és igények
  - Vision Document
    - Az elképzelt jövőbeli rendszer főbb jellemzői, használatának haszna
  - Product backlog
    - Feladatok prioritizált listája, ami az összes ismert követelményt lefedi
  - Sprint backlog
    - A következő iterációban megvalósítandó feladatok prioritizált listája
  - Story map
    - User story-k kapcsolatának grafikus megjelenítése

## A Kurzusról

- Bemutató
- Az RE szerepe

## 1 Bevezetés

- Mit?
- Miért?
- Hol?
- Hogyan?
- Szerepkörök és feladatok
- Mit kell tudni?

## 2 Az RE alapelvei

- Az alapelvek áttekintése
- Az alapelvek elmagyarázva

## 3 Munkatermékek és dokumentáció

- Munkatermékek a követelménytervezésben
- Természetes nyelv alapú munkatermékek
- Sablon alapú munkatermékek
- Modell alapú munkatermékek
- Szójegyzékek
- Követelménydokumentum szerkezete
- **Prototípusok**
- Minőségi kritériumok

## 4 A követelménytervezés lépései

- A követelmények forrásai
- Követelmények feltárása
- Konfliktusfeloldás
- Követelmények validálása

## 5 Folyamatok és munka

- Befolyásoló tényezők

- RE folyamat szempontok
- Az RE folyamat konfigurálása

## 6 Követelmények menedzsmentje

- Mi az a követelménymenedzsment?
- Életciklus menedzsment
- Verziókövetés
- Konfigurációk
- Tulajdonságok és nézetek
- Nyomonkövethetőség
- Változások kezelése
- Priorizálás

## 7 Tool Support

- Eszközök a követelménytervezésben
- Eszközök bevezetése

## Prototype

*Software and systems eng.: A preliminary, partial realization of certain characteristics of a system.*

*Design: A preliminary, partial instance of a design solution.*

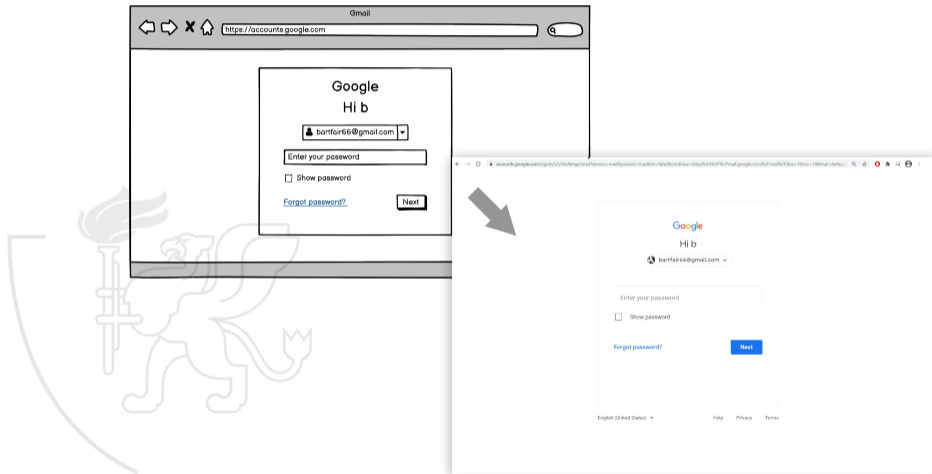
- „Felfedező” (exploratory) prototípus
  - Közös megértés elősegítésére, követelmények tisztázására, validációjára létrehozott ideiglenes munkatermék
  - Követelmények definiálására létrehozott fejlődő vagy tartós munkatermék
- Kísérleti (experimental) prototípus
  - Technikai megoldások kipróbálására, ellenőrzésére; ideiglenes munkatermék (az RE-ben nem használjuk)
- Evolúciós (evolutionary) prototípus
  - Folyamatosan fejlődő rendszer, melynek egyes (fejlesztői) verziói használhatók és egymásra épülnek



- Az RE-ben leginkább exploratory prototípusokat használunk
  - Feltáráskor: specifikáció példán keresztül („*specification by example*”)
  - Validáláskor: a követelmények megfelelőségének („*adequacy*”) ellenőrzésére
- Az exploratory prototípusok kategóriái (*fidelity* szerint)
  - Wireframes (low-fidelity): egyszerű elemekből, ötletek megbeszéléséhez, gyorsan elkészíthető és módosítható, egy-egy kulcstulajdonságra fókuszáló prototípus
  - Mock-ups (medium-fidelity): realiztikus kinézetű, valós funkcionalitással nem rendelkező, főként user interfészek validálására vagy specifikálására használt prototípus
  - Native (high-fidelity): a rendszer kritikus részeit valósítja meg, megfelelő viselkedéssel, funkcionalitással, esetleg változatokban, specifikációra és validációra is használható
- Egy prototípus költséges mulatság lehet, de vannak előnyei, értékei: mérlegelni kell!

# Prototípusok

## Wireframe és Mockup példa



## A Kurzusról

- Bemutató
- Az RE szerepe

## 1 Bevezetés

- Mit?
- Miért?
- Hol?
- Hogyan?
- Szerepkörök és feladatok
- Mit kell tudni?

## 2 Az RE alapelvei

- Az alapelvek áttekintése
- Az alapelvek elmagyarázva

## 3 Munkatermékek és dokumentáció

- Munkatermékek a követelménytervezésben
- Természetes nyelv alapú munkatermékek
- Sablon alapú munkatermékek
- Modell alapú munkatermékek
- Szójegyzékek
- Követelménydokumentum szerkezete
- Prototípusok

## • Minőségi kritériumok

## 4 A követelménytervezés lépései

- A követelmények forrásai
- Követelmények feltárása
- Konfliktusfeloldás
- Követelmények validálása

## 5 Folyamatok és munka

- Befolyásoló tényezők

- RE folyamat szempontok
- Az RE folyamat konfigurálása

## 6 Követelmények menedzsmentje

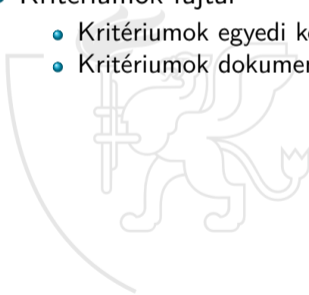
- Mi az a követelménymenedzsment?
- Életciklus menedzsment
- Verziókövetés
- Konfigurációk
- Tulajdonságok és nézetek
- Nyomonkövethetőség
- Változások kezelése
- Priorizálás

## 7 Tool Support

- Eszközök a követelménytervezésben
- Eszközök bevezetése

# Követelmények minőségi kritériumai

- A követelményeket „megfelelő” minőségben kell elkészíteni
- A megfelelőséget különböző kritériumok szerint vizsgálhatjuk
- Nincs egy általánosan elfogadott teljes kritériumhalmaz
  - Nem kell mindig minden kritériumot kielégíteni
  - Néhány kritérium fontosabb, mint a többi
- Kritériumok fajtái
  - Kritériumok egyedi követelményekhez
  - Kritériumok dokumentumokhoz



- **Megfelelő (adequate)**
  - Valódi, egyeztetett követelményt ír le
- **Érthető (understandable)**
  - A célközönség számára teljesen érthetően van megfogalmazva
- Szükséges (necessary)
  - Valamely érdekelt fél céljaiból ered
- Egyértelmű (unambiguous)
  - A közös megértés alapján minden érintett számára ugyanazt jelenti
- Teljes (complete)
  - A követelmény önmagában érthető, nem hiányzik hozzá információ
- *Ellenőrizhető (verifiable)*
  - A követelmény megvalósulása egyértelműen eldönthető

# Dokumentumok minőségi kritériumai

- Következetes (consistent)
  - A követelmények között nincs ellentmondás (ugyanabban és különböző munkatermékekben sem)
- Nem redundáns (non-redundant)
  - Minden követelmény csak egy példányban van dokumentálva és nem átfedőek
- Teljes (complete)
  - A munkatermék minden, *az adott időpontban* releváns követelményt (funkcionális, minőségi, megszorítás) tartalmaz
- Módosítható (modifiable)
  - A dokumentum a minőség romlása nélkül módosítható, bővíthető (pl. verziókövetés, atomi követelmények, ...)
- Nyomonkövethető (traceable)
  - Ismertek a követelmények forrásai, a ráépülő munkatermékek, valamint a követelmények közötti függőségek
- Előírás szerinti (conformant)
  - Ha vannak kötelező formai, szerkezeti elemek, akkor annak megfelel (pl. fejezetek, jegyzékek, képaláírások, ...)