

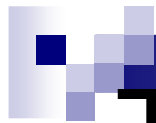


Szoftvertchnológia

4. Követelmények kezelése

BSc kurzus

Dr. Balla Katalin



Tartalom

- A követelmények fontossága
- Requirement Engineering / Követelmények fejlesztése, követelmények kidolgozása
- Követelmények forrásai
- Követelmények felmérése, elemzése, priorizálása
- Követelmények típusai
- Követelmények modellezése, dokumentálása
- Követelménymenedzsment
 - Kétirányú követhetőség
 - Követelmények változásának kezelése
- Példák különböző követelményekre
- Követelmények agilis környezetben



A követelmény definíciója

- Egy követelmény:

- ☐ 1. Olyan feltétel vagy képesség, amely a felhasználó számára azért szükséges, hogy megoldjon egy problémát, vagy elérjen egy célt.
- ☐ 2. Ezen feltételnek vagy képességnek a rendszer vagy rendszer komponens által is megvalósíthatónak kell lennie úgy, hogy közben a szerződés, szabvány, specifikáció és egyéb formális dokumentumban támasztott követelményeknek is megfeleljen
 - Megj: Ez a definíció a „klasszikus”, az IEEE Std 610.12 szabványból, 1990-ből.

- Alább, egy modernebb definíció:

- Követelmény:

- ☐ 1. Egy érdekelt fél által felismert igény.
- ☐ 2. Olyan képesség vagy tulajdonság, amellyel a rendszernek rendelkeznie kell.
- ☐ 3. Egy igény, képesség vagy tulajdonság dokumentált leírása.



Követelményekkel kapcsolatos problémák

- A vevők / felhasználók nem tudják, hogy valójában mit szeretnének
 - Még ha tudják is, nem tudják elmagyarázni a szoftverfejlesztőknek
- Kommunikációs problémák / félreértések merülnek fel
 - vevők
 - fejlesztők
 - menedzsment között.
- A szoftver működési elemeit „magától értetődőnek” tekintik
- (Majdnem) mindig ott a nyomás, hogy a szoftver „gyorsan” készüljön el.
 - A vevő / menedzsment / sales csapat teljesíthetetlen határidőket erőszakolnak rá a fejlesztőkre
- A követelmények változnak a projekt során
 - A felhasználó nincs tudatában a változások hatásának

Bizonyára ismerős...



How the customer explained it



How the project leader understood it



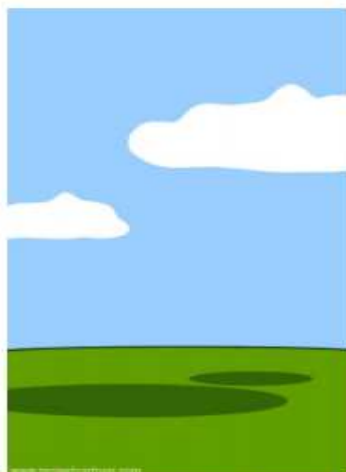
How the analyst designed it



How the programmer wrote it



How the business consultant described it



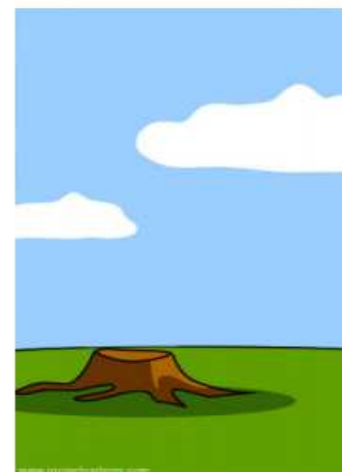
How the project was documented



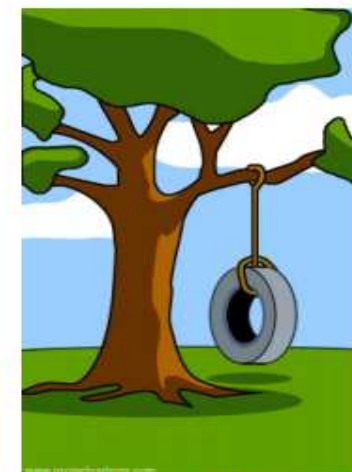
What operations installed



How the customer was billed



How it was supported



What the customer really needed

<http://www.csfieldguide.org.nz/releases/1.9.9/teacher/SoftwareEngineering.html>

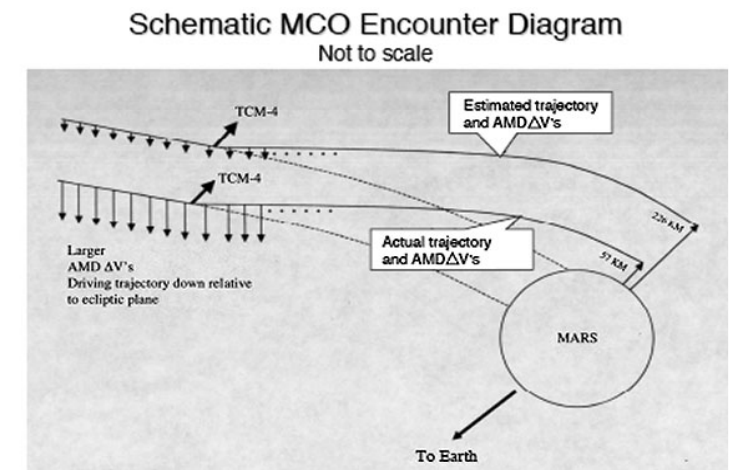


Követelmények miatti „hírhedt” hibák

- Az Ariane–5 első tesztrepülése (Ariane–5 Flight 501) 1996. június 4-én sikertelen volt. A rakéta önmegsemmisítést hajtott végre indítás után 37 mp-el az irányító szoftver hibája miatt.
 - Ez volt a történelem egyik legdrágább számítógépes meghibásodása (kb. 370 millió USD)
- Az önmegsemmisítést teljes információvesztés okozta, ez pedig az IRS szerinti követelmények és tervezés hiányossága miatt következett be.
 - A (belső) hiba egy 64 bites adat 16 bitre történő konvertálásakor fellépő kivétel volt. A 64 bites adat túl nagy volt, hogy 16 biten elférjen.
SRI stands for Système de Référence Inertielle or Inertial Reference System. -
IRS – tehetetlenségi vonatkoztatási rendszer
<http://www-users.math.umn.edu/~arnold/disasters/ariane.html>

Követelmények miatti „hírhedt” hibák

- 1999 szeptemberében a Mars Climate Orbiter misszió sikertelenül végződött. A marsjáró 41 hétnyi utazás után, a Mars körüli pályára állás pillanatában eltűnt.
- A hiba: a Lockheed Martin Astronautics-nál a gyorsulási adatokat angol mértékegységben (fontban) kezelték, a Jet Propulsion Laboratory-nál pedig metrikus rendszerben (newton).
- A NASA egy 50000USD költségű projektet indított annak kiderítésére, hogy ez hogyan fordulhatott elő.
 - Követelmények nem voltak pontosak
 - Nem volt megfelelő az integrációs teszt.



Unit Mixups

<http://www.us-metric.org/unit-mixups/>



Miért szükségesek a pontos, világosan megfogalmazott követelmények?

- Hogy a rendszer határát kijelöljük / megfogalmazzuk
- Hogy biztosítsuk a (jó) tervezési és implementálási tevékenységeket
- Hogy biztosítsuk: a rendszer azt fogja csinálni, amit kell
- Hogy be tudjuk bizonyítani az ügyfélnek: elvégeztük a feladatot
- Hogy rendet tudjunk tartani a (számos) követelmény között; ehhez szükséges a követelményeket megfelelően dokumentálni!
- Hogy a követelményeket leíró dokumentumok felhasználhatók legyenek a következőkre:
 - Tervezés, implementáció
 - Teszt esetek, teszt forgatókönyvek létrehozása
 - Felhasználói dokumentáció létrehozása (pl. oktatási anyag, felhasználói kézikönyv, karbantartási leírások...)

Mit tehetünk, hogy jó követelményeink legyenek?



- A szoftvermérnökök segíthetnek a **követelmények megfogalmazásában**
 - Egyes követelmények a felhasználótól jönnek: **ilyenkor le kell fordítani** őket termék / rendszer követelményekre
 - Egyes követelmények az IT / üzleti tudás alapján kerülnek elő; ezeket **meg kell fogalmazni** (pl. mint nem-funkcionális követelményeket)
 - Néhány követelményt „**feltalálnak**” (nincs rá korábbi explicit kérés, igény ...)
 - Gondoljunk a Facebook-ra!

Requirements engineering (RE)

Követelményfejlesztés (RE)



■ A RE négy alaptevékenysége:

- Meghatározás, egyeztetés, felmérés / elicitation
- Dokumentálás
- Validáció, egyeztetés / Validation/negotiation
- Követelmények menedzsmentje.

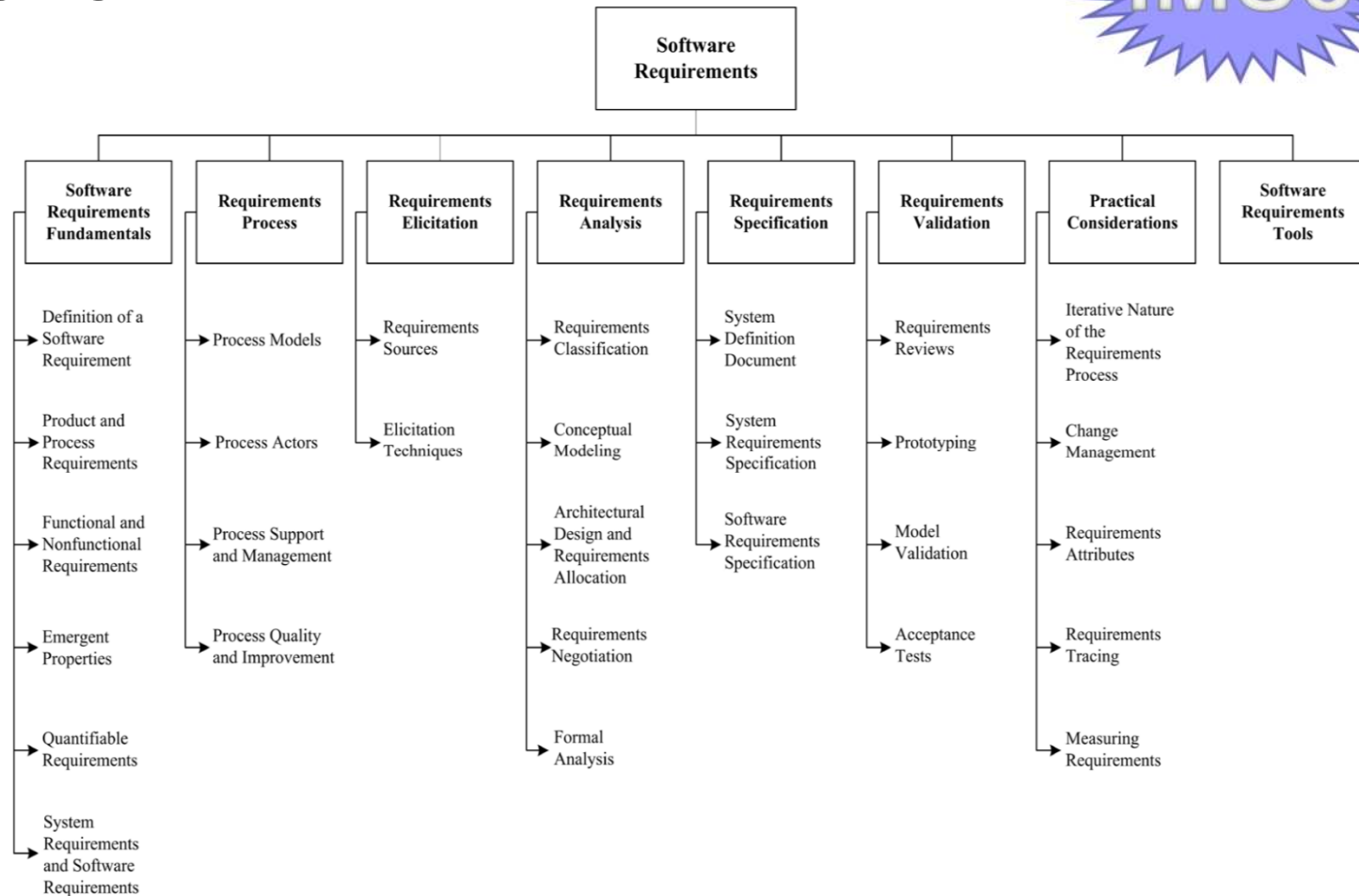
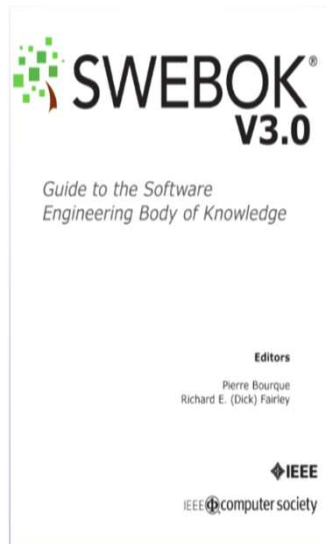
□ Ezeket a tevékenységeket különböző folyamatokba lehet rendezni, ahogyan ezt szabványok, modellek leírják, például:

- ISO/IEC/IEEE 29148:2011 : Systems and software engineering -- Life cycle processes -- Requirements engineering, IREB, CMMI...stb.

□ Ezen tevékenységek elvégzésére **idő és erőforrások szükségesek!**

- A RE a szoftverfejlesztési projekt lényeges eleme!
- **Ha már itt tartunk: Tudja Ön, hogy mennyi időt szokott „követelményekkel” eltölteni? Az Ön szoftverfejlesztésének hány % -a követelményfejlesztés?**

Követelményekhez kapcsolódó témakörök



Breakdown of Topics for the Software Requirements KA. 2004-2005



Követelményfejlesztés (RE)

- A követelmények meghatározásának és menedzselésének szisztematikus megközelítése, a következő célokkal:
 - (1) A helytálló, megfelelő követelmények megismerése, az érdekelt felek egyetértésének elérése ezekre a követelményekre vonatkozóan, a követelmények szabványos dokumentálása, a követelmények folyamatos menedzselése
 - (2) Az érdekelt felek óhajainak és szükségleteinek megértése és dokumentálása
 - (3) A követelmények meghatározása és menedzselése, hogy ezáltal csökkenjen olyan rendszer átadásának kockázata, amely nem felel meg az érdekelt felek elvárásainak és szükségleteinek.
 - Megj: mindhárom cél lényeges eleme a RE-nek: : (1) folyamat-központúság, (2) érdekelt felekre figyelés, és (3) a kockázatokra és értékekre vonatkozó gondolatmenet.
 - Forrás: IREB , International Requirements Engineering Board, <https://www.ireb.org/en>



Követelményfejlesztés a CMMI-ben

- ML2: Requirements management /
Követelménymenedzsment
- ML3: Requirements development /
követelményfejlesztés



Követelménymenedzsment (REQM)

- Célja a projekt termékeihez és termék komponenseihez szükséges követelmények menedzselése, és a követelmények és a projekt tervek és munkatermékek között esetleg fellépő ellentmondások azonosítása.
- SG1 Követelmények menedzselése
 - SP 1.1 Követelmények megértésének elérése
 - SP 1.2 A követelményhez való elkötelezettség elérése
 - SP 1.3 Követelményváltozás menedzselése
 - SP 1.4 Kétirányú követhetőség fenntartása
 - SP 1.5 A projekt munka és követelmények közötti ellentmondások azonosítása
- Ezek a tevékenységek a CMMI 2-es érettségi szintjén szükségesek! Ez azt jelenti, hogy alapvetően fontosak !



Követelményfejlesztés (RD)

- Célja a vevői, termék, és termék-komponens követelmények felmérése, elemzése és dokumentálása.
- SG 1 Vevői követelmények fejlesztése
 - SP 1.1 Szükségletek felderítése
 - SP 1.2 Érdeelt felek igényeinek vevői követelményekké alakítása
- SG 2 Termékkövetelmények fejlesztése
 - SP 2.1 Termék és termék-komponens követelmények meghatározása
 - SP 2.2 Termék-komponens követelmények allokálása
 - SP 2.3 Interfész követelmények azonosítása
- SG 3 Követelmények elemzése és jóváhagyása
 - SP 3.1 Működési elképzelések és forgatókönyvek meghatározása
 - SP 3.2 Az igényelt funkcionalitás és minőségi jellemzők definiálása
 - SP 3.3 Követelmények elemzése
 - SP 3.4 Követelmények elemzése egyensúlyi állapot eléréséhez
 - SP 3.5 Követelmények validálása
- Ezek a tevékenységek a CMMI 3-as érettségi szintjén szükségesek! Mélyebb szakmai tudást feltételeznek!





Requirements Management - REQM

- The purpose of Requirements Management (REQM) is to manage requirements of the project's products and product components and to ensure alignment between those requirements and the project's plans and work products.
- SG 1 Manage Requirements
 - SP 1.1 Understand Requirements
 - SP 1.2 Obtain Commitment to Requirements
 - SP 1.3 Manage Requirements Changes
 - SP 1.4 Maintain Bidirectional Traceability of Requirements
 - SP 1.5 Ensure Alignment Between Project Work and Requirements
- **These activities are needed at ML2! Message: they are basic!**



Requirements development -RD

- The purpose of Requirements Development (RD) is to **elicit, analyze, and establish customer, product, and product component requirements.**
- SG 1 Develop Customer Requirements
 - SP 1.1 Elicit Needs
 - SP 1.2 Transform Stakeholder Needs into Customer Requirements
- SG 2 Develop Product Requirements
 - SP 2.1 Establish Product and Product Component Requirements
 - SP 2.2 Allocate Product Component Requirements
 - SP 2.3 Identify Interface Requirements
- SG 3 Analyze and Validate Requirements
 - SP 3.1 Establish Operational Concepts and Scenarios
 - SP 3.2 Establish a Definition of Required Functionality and Quality Attributes
 - SP 3.3 Analyze Requirements
 - SP 3.4 Analyze Requirements to Achieve Balance
 - SP 3.5 Validate Requirements



These activities are needed at CMMI ML3 ! Message: they are more than basic, but build on the basic ones!



Követelmények forrásai

- Felhasználó
- Más (hasonló) rendszerek
- „Általános” tudás, tapasztalat az üzleti környezettel kapcsolatban és a szoftverfejlesztéssel kapcsolatban
- A követelmények forrására vonatkozóan– ha ők érdekelt felek - legalább a következő információkat szokás feljegyezni:
 - Név
 - Beosztás, szerepkör (role)
 - További személyes vagy kontakt adatok
 - Térben és időben mikor ér rá a projekt időtartam alatt
 - Az érdekelt fél relevanciája
 - Tapasztalat és ismeretek
 - A projekttel kapcsolatos céljaik és érdekeik
- Hozzunk létre szótárt, amely tartalmazza a követelményekben megfogalmazott (alap)fogalmakat!



Követelmények forrásai

- Példák olyan követelményekre, amelyeket a vevő nem szokott specifikálni:
 - ☐ Üzleti alapelvek / Business policies
 - ☐ Szabványok
 - ☐ Korábbi architekturális tervezési döntések és elvek
 - ☐ Üzleti környezettel kapcsolatos követelmények (pl. laboratórium, teszt környezet, IT infrastruktúra...)
 - ☐ Örökölt termékek vagy termék komponensek (újrafelhasznált komponensek)
 - ☐ Törvényi szabályozások ...



Követelmények egyeztetése

- Alapvető a felhasználóval kommunikálni!
- Egyeztetési technikák
 - felmérési technikák Survey techniques (pl. interjúk, kérdőívek)
 - Kreatív technikák (pl. brainstorming, nézőpont változtatás, analógiára alapuló technika)
 - Dokumentum - központú technikák – (pl. rendszer-régészet, követelmények újrafelhasználása)
 - Megfigyelési technikák (pl. helyszíni megfigyelés, tanonckodás)
 - Támogató technikák (pl. mind mapping, workshopok, audio és video felvételek, use case modellezés, prototípusok készítése)

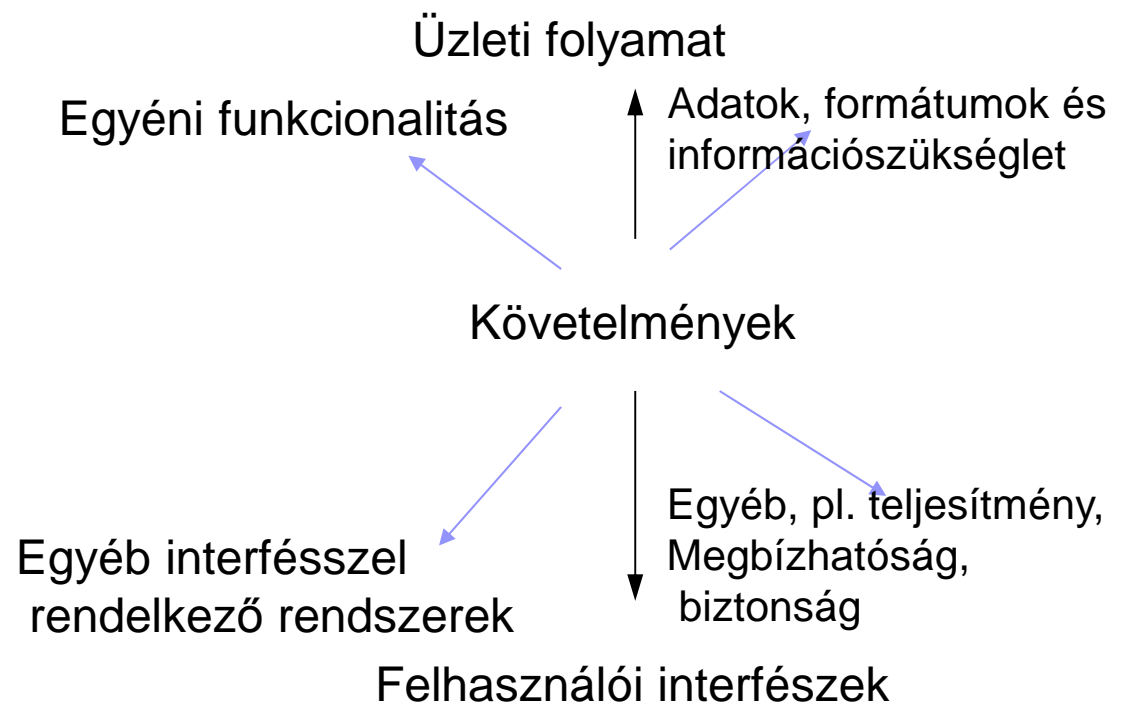
Követelmények egyeztetése és meghatározása

■ Magas szinten:

- Üzleti lehetőségek és üzleti igények
- A projekt létének igazolása
- Szkóp
- Legfontosabb kötöttségek, korlátok
- Legfontosabb funkcionalitás
- Sikertényezők
- Felhasználók jellemzése



■ Alacsony szinten



6 Dimensions of Detailed Requirements Elicitation. From: Essentials of Software Engineering. Frank Tsui, Orlando Karam, Barbara Bernal



Követelmények egyeztetése

- Az egyeztetések célja, hogy egyetértés szülessen a követelményekről a különböző érdekelt felek között.
- Az követelmények egyeztetésének feladatai:
 - ☐ A konfliktus azonosítása
 - ☐ A konfliktus elemzése
 - ☐ A konfliktus megoldása
 - ☐ A konfliktus –megoldás dokumentálása
 - ➔ a következő alkalomra lesznek tapasztalatok!



Követelmények elemzése

- Az egyeztetett követelmények elemzése abból a célból, hogy megértsük és dokumentáljuk őket .
 - (Ezt a tevékenységet néha a követelményfejlesztés szinonímájaként kezelik.)
 - Szükség van gondolkodásra, megbeszélésekre, brainstorming-ra...
- A követelmények csoportosítása
- A követelmények priorizálása



Követelmények típusai

- Funkcionális követelmények
- Nem-funkcionális követelmények / Minőségi követelményeknek is nevezik őket
 - Az IREB egy harmadik követelmény-kategóriát is meghatároz: korlátok
- Követelmény-típusnak tekinthetjük még az alábbiakat is :
 - Üzleti követelmények
 - Felhasználói követelmények
 - Termék követelmények
 - Interfész követelmények



Funkcionális követelmények

- „Amit a rendszer csinál ”
- **Funkcionális követelmény:** olyan követelmény, amely a szoftverrel szemben támasztott funkcionális elvárást írja le. [IEEE 610]. → *functional requirement*
- A funkcionális követelményt leíró dokumentumot Funkcionális Követelményspecifikációnak is nevezik. Ezt Functional Requirements Specification is **széles / általános hallgatóság számára fejlesztik**. Az olvasónak meg kell értenie a rendszert, de semmilyen sajátos vagy technikai tudásra nincs szükség ehhez.
- A funkcionális követelmények nem azonosak az üzleti követelményekkel!
- A funkcionális követelményeknek tartalmazniuk kell:
 - ☐ A rendszer bemenő adatainak leírását
 - ☐ Az egyes képernyők által megvalósított műveletek leírását
 - ☐ A rendszer munkafolyamatának leírását
 - ☐ A rendszer kimenő adatainak, a jelentéseknek a leírását
 - ☐ Annak leírását, hogy ki vihet be adatot a rendszerbe
 - ☐ Annak leírását, hogy a rendszer miképpen felel meg az érvényes törvényi szabályozásoknak



Nem-funkcionális követelmények

- „Ahogyan a rendszer csinálja”
- Minőségi követelményeknek is szokás nevezni őket
- Expliciten dokumentálni kell őket. Különös tekintettel az alábbiakra:
 - ☐ Teljesítmény
 - ☐ Biztonság
 - ☐ Megbízhatóság
 - ☐ Használhatóság
 - ☐ Karbantarthatóság
 - ☐ Hordozhatóság
- Részletesebb leírás : ISO/IEC25010:2011
 - 2011 március 1-én a korábbi ISO/IEC 9126 szabványt az ISO/IEC 25010:2011 szabvány váltotta fel. -Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models.

Nem-funkcionális követelmények alapja



<http://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

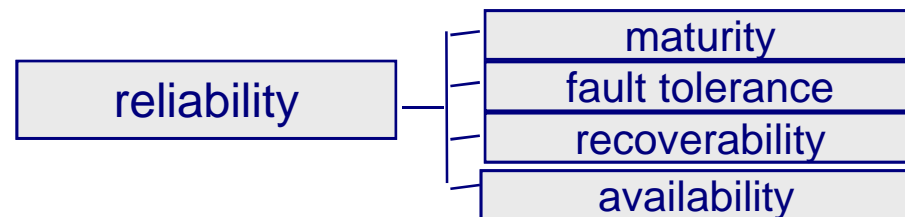


Nem-funkcionális követelmények (az ISO 9126 szerint)

Minőségi jellemző	Meghatározás
Megbízhatóság	Azon attribútumok összessége, amelyek hatással vannak a szoftver azon képességére, hogy teljesítményének szintjét - bizonyos feltételek mellett, meghatározott időre – megőrizze.
Használhatóság	Azon attribútumok összessége, amelyek hatással vannak a szoftver használatához szükséges erőfeszítésre, és a használat bizonyos meghatározott vagy feltételezett felhasználócsoporthoz általi értékelésére.
Hatékonyság	Azon attribútumok csoportja, amelyek hatással vannak az elért eredmény és a felhasznált erőforrás viszonyára, megadott feltételek mellett.
Karbantarthatóság	Bizonyos, meghatározott módosítás elvégzéséhez szükséges erőfeszítésre hatást gyakorló attribútumok összessége.
Hordozhatóság	Olyan attribútumok összessége, amelyek befolyásolják a szoftvernek azt a képességét, hogy adott környezetből másikba átvihető legyen.

Példa nem- funkcionális követelményekre

- Folyamatos, kritikus menedzsment döntések
- a szervezetnek a szoftverterméktől való erős függősége
 - megbízhatóságra / elérhetőségre (availability) vonatkozó követelményeket eredményez



Példa nem- funkcionális követelményekre

Megbízhatóság: A szoftver azon képessége, hogy teljesítményét bizonyos feltételek mellett megőrizze

- *érettség:*

A rendszer azon képessége, hogy “ne omoljon össze” a szoftver hibái miatt

hogyan mérhető: *Mean Time Between Failures =*
működési idő / meghibásodások száma

- *hibatűrés:*

A szoftver azon képesége, hogy bizonyos szinten működőképes maradjon hibák megjelenése esetén is

hogyan mérhető. *Input Error Detection Ratio =*
Azonosított bemeneti hibák száma az összes inputhoz viszonyítva

Példa nem- funkcionális követelményekre

- *visszaállíthatóság:*

A szoftvertermék azon képessége, hogy hiba esetében visszaállítson egy meghatározott működési szintet és a hiba által érintett adatokat.

Hogyan mérhető: Mean Down Time =

A rendszer teljes működésképtelenségének ideje / az azonosított meghibásodások

- *elérhetőség:*

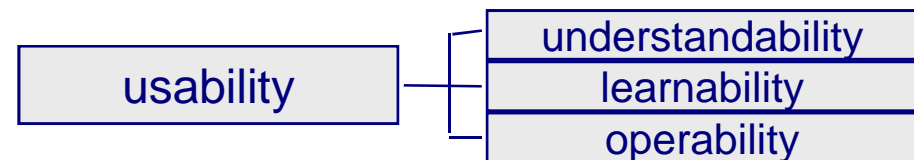
A szoftvertermék azon képessége, hogy egy adott funkciót egy adott pillanatban biztosan végrehajtson

hogyan mérhető: Relative Availability Percentage =

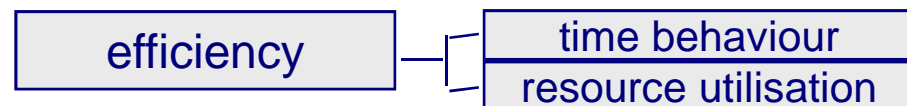
A szoftver elérhetőségének és szükségességének aránya

Példa nem-funkcionális követelményekre

- Sok tapasztalatlan felhasználó
- alacsony iskolázottság
 - könnyen tanulhatóság követelményét eredményezi



- sok menedzser van a felhasználók között
- sok a tapasztalatuk
 - fontossá teszi a rövid válaszidőket (időfüggő viselkedés)





Felhasználói követelmények

- Magas szintűek.

- ☐ Kezdetben kicsit bizonytalanok.

- ☐ Segíteni kell a felhasználónak megfogalmazni őket.

- Pl: „A rendszernek tárolnia kell a vevő megrendelésének minden részletét.”

- ☐ Tehát, azt írjuk le, hogy **MIT** akarunk, és nem azt, **HOGYAN** fogjuk ezt megoldani!

- ☐ <http://www.coleyconsulting.co.uk/require.htm>



Felhasználói követelmények. Példák.

■ Működési korlátok

- ☐ Az információt maximum 3 sec alatt meg kell jeleníteni ill. el kell tárolni
- ☐ A rendszernek hétfőtől péntekig reggel 9-től délután 5-ig rendelkezésre kell állnia.
- ☐ A rendszernek 100,000 felhasználói rekordot kell tárolnia.
- ☐ A rendszernek 10 évig évi 10000 további rekordot kell tudnia tárolni.
- ☐ Egy rekord minden elemének a rendszerben kell maradnia legalább 7 évig.

- ☐ <http://www.coleyconsulting.co.uk/require.htm>



Felhasználói követelmények

- Kerüljük, hogy tervezési vagy implementációs megoldások követelményként kerüljenek leírásra!
- Példák **tervezési megoldásokra** :
 - ☐ A rendszernek a meglévő gépparkunkon kell futnia.
 - ☐ A vevői rekord külön mezőkben tárolja a családnevet és a keresztnévet.
- Példák **implementációs megoldásokra** :
 - ☐ A rendszernek SQL adatbázist kell használnia.
 - ☐ A rendszert Java-ban kell programozni.



Termék követelmények

- Segítenek megérteni, hogy az új elemeket hogyan fogjuk felhasználni a probléma megoldására és célunk elérésére
- Példák egy termék követelményeket leíró dokumentum elemeire: (PRD).
 - Cél
 - Központi elemek
 - Felhasználói adatfolyam
 - Minden egyes felhasználói lépés részletezése
 - Analitika
 - Jövőbeli (várható) kiegészítések
 - <https://www.aha.io/roadmapping/guide/requirements-management/what-is-a-good-product-requirements-document-template>



Termék követelmények

■ Célok

- ☐ Kinek készül a termék / elem ?
- ☐ Milyen problémát hivatott megoldani?
- ☐ Mi a célja ennek az elemnek?
- ☐ Mit fog ez az elem végrehajtani, és mit nem ?

■ Központi elemek

- ☐ Ha pl. egy új e-kereskedelmi elemet fejleszt, mely lehetővé teszi színes ceruzák vásárlását, a központi elemek lehetnek:
 - Keresési eredmények oldal (hogyan találják meg a felhasználók a kívánt színes ceruzákat)
 - Termék részleteit leíró oldal (információ a kedvenc színes ceruzáról)
 - Kilépési oldal (milyen adatokat kell megadni a felhasználónak, hogy rendelhessen)
 - Rendelés visszaigazolási oldal (mi történik a vásárlás után?)



Termék követelmények

■ Felhasználói folyam(at) :

- ☐ Home Page
- ☐ Sam, a színes ceruza rajongó beír egy szót a keresőbe és a keresési oldalra kerül át
- ☐ Keresési oldal
- ☐ Sam ráklikkel a kék színű ceruzára, hogy több információt tudjon meg róla
- ☐ Termék részleteit leíró oldal
 - Sam eldönti, hogy megveszi ezt a ceruzát és rákattint a „kosárba” gombra
 - Sam rákattint a „Kilépés,, gombra
- ☐ Kilépő oldal
- ☐ Sam kitölti a kilépő oldalon kért adatokat és elküldi őket
- ☐ Rendelés visszaigazolása oldal
- ☐ Sam látja a visszaigazolást
- ☐ Rendelést visszaigazoló email
- ☐ Sam email-ben megkapja a rendelése visszaigazolását



Termék követelmények

- Részletek a felhasználó lépéseihez :
 - Pl: A termék részleteit leíró oldal elemei
 - Ár
 - Termék neve
 - Leírás
 - Megosztás közösségi oldalakon
 - Vélemények
 - Vásárlás
 - Fényképek



Termék követelmények

■ Analitika

- Gondoljunk olyan mérőszámokra, amelyek majd megmutatják, mennyire jó a termékünk

- pl: látogatók száma oldalanként

- **Keresési oldal:** 100 fő
- **Termék részletei oldal:** 80 fő
- **Kilépő oldal:** 70 fő
- **Rendelés visszaigazolás:** 10 fő

- Látható, hogy kevesen jártak a rendelés visszaigazolása oldalon a többi oldalhoz képest. Valószínűleg optimalizálni kell a kilépő oldalt.

- Remélhetőleg ezzel növelni tudjuk azok számát, akik rendelés után lépnek ki

■ Jövőbeli elemek

- A csapat ebből tudja megérteni, hogyan fog fejlődni a termék.



Korlátok / korlátozások

- A korlátozás olyan követelmény, amely leszűkíti a megoldási lehetőségeket azon belül, ami ahhoz szükséges, hogy a rendszer a számára elírt funkcionális vagy nem funkcionális elvárásokat teljesítse.
- A projektcsapat a korlátozásokat nem tudja befolyásolni.
 - Pl: „A rendszert web service alkalmazásával kell megvalósítani”
 - Pl: „A rendszernek legkésőbb 2017 novemberében működni kell, azért hogy ...



Követelmények modellezése

- A követelményeket fogalmi modellekben szoktuk leírni; ezek természetes nyelven, grafikai elemekkel vagy a kettő kombinációjával kísérik meg minél jobban leírni a követelményeket.



Követelmények modellezése

■ Modell típusok:

☐ „Cél modellek” / Goal models

- A célok egy érdekelt fél szempontjait írják le. Ezek tipikusan a rendszer funkcióival kapcsolatosak. Természetes nyelven vagy grafikusan is leírhatók. A célok lebontása hasznos .

☐ Use case-ek / használati esetek modelljei

- Segítenek megvizsgálni a tervezett vagy meglévő rendszert a felhasználó szemszögéből. Use-case diagramok és use-case leírások alkalmazhatók.

☐ ...



Követelmények modellezése

- Követelményeket leíró modellek esetében 3 nézőpontból végezhetünk modellezést:
 - ☐ Adatok
 - ☐ Viselkedés
 - ☐ Funkcionalitás
- Ezek mindegyikét leírhatjuk természetes nyelven vagy különböző ábrázolási technikákkal.



Követelmények modellezése

- Adatközpontú követelménymodellezés
 - Tipikusan ERD-k (entity relationship diagrams, entitás relációs diagramok)
- Funkcionális központú követelménymodellezés
 - Alapvetően **az input adatok outputtá alakításával foglalkozik**. Sokféle funkcionális modell létezik: (UML) kontextus diagramok, adatfolyam diagramok, folyamatokat leíró diagramok, adatforrások és adattárak, amelyek az inputot / outputot tárolják
- Viselkedés központú követelménymodellezés
 - Tipikusan **a rendszer állapotai és az ezeket befolyásoló események**. UML állapotdiagramok, kezdő és végállapotok leírásai, állapotátmenetek leírásai , konkurencia



Követelmények modellezése

- Hasznos lehet a következő diákon megjelenő elemeket (is) felhasználni.
- A követelményekben megjelenő hiba igen veszélyes, de sajnós, gyakori! Ne sajnáljuk az időt a **követelmények ellenőrzésére!**
- Jó megközelítés, ha ellenőrizzük, hogy a különböző, követelményeket leíró modellek elemei konzisztensek-e!
 - ☐ Eszközök segíthetnek!



Követelmények modellezése

■ Felhasználható alapelemek:

- ☐ Adatok
- ☐ Akciók
- ☐ Eszközök
- ☐ Események
- ☐ Vezérfonalak

(Jorgenssen alapján)



Követelmények modellezése

■ Adatok

- Ha a rendszert adataival jellemezzük, akkor a hangsúly a rendszer által felhasznált és létrehozott információkon van.
- Adatok: változók, adatstruktúrák, mezők, rekordok, adattárok, file-ok.
- Adatok leírása: pl. magas szinten: entity/ relationship diagramok, alacsony szinten: Jackson diagram
 - Előfordul, hogy közvetlenül az adatmodellekből tudunk vezérfonalakat azonosítani
 - Pl. az egy-egy, egy-sok, sok-egy, sok-sok kapcsolatok értelmezése az adatok feldolgozásában



Követelmények modellezése

■ Akciók

- ☐ Van inputjuk és outputjuk (adatok vagy port-események formájában)
- ☐ Akciók leírása pl: transform, data transform, control transform, process, activity, task, method, service
- ☐ Az akciók lebonthatók

■ Eszközök / Device

- ☐ Minden rendszernek van portja, az I/O események itt jelennek meg
- ☐ A felhasználó a porton megjelenő eseményeket látja



Követelmények modellezése

■ Események

- Adatjellemzők és akciók is kapcsolódnak hozzá
- Olyan rendszerszintű input vagy output, amely a porton jelenik meg
- Diszkrét események: adott hosszúságú idő kapcsolódik hozzájuk

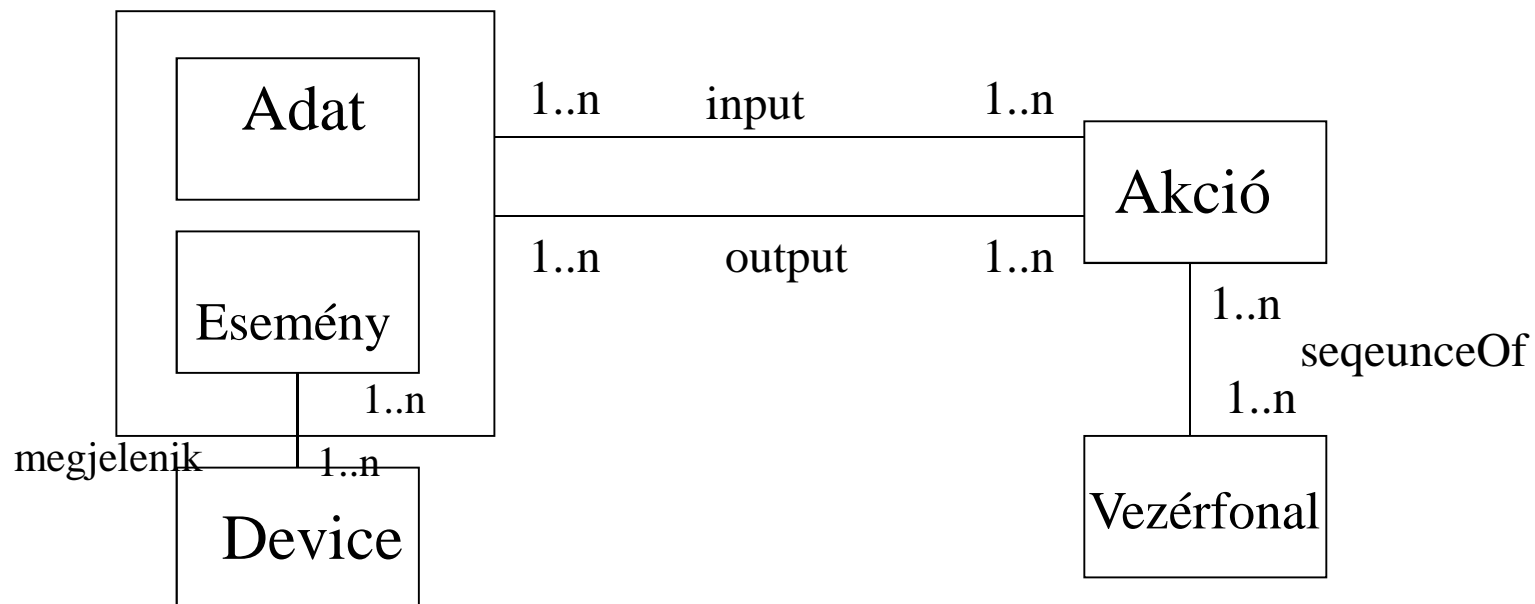
■ Vezérfonalak / szálak / fonalak

- Leginkább a tesztelésben használatosak, ezért általában a tesztelő feladata azonosítani őket
- Követelményspecifikációban prototípus fejlesztésnél hasznosak
- Vezérfonalakat könnyen lehet azonosítani, ha a rendszert és környezetét határoló portokat figyeljük; egy vezérfonal a potról induló input és a portra érkező output közötti „útvonal”

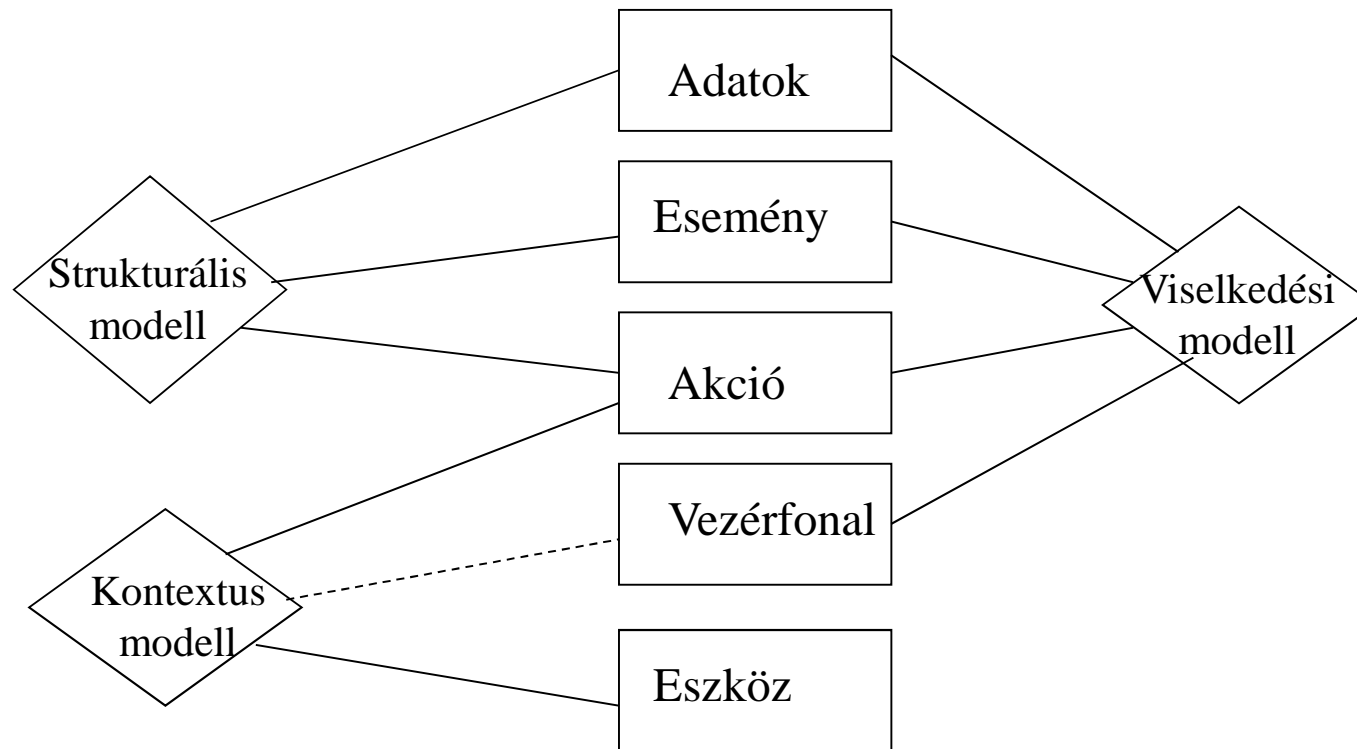
Követelmények modellezése



- Az alapfogalmak közötti összefüggések
 - Elkészíthető az alapfogalmak Entity-relationship diagramja: adatok és események, ezek közötti kapcsolatok
 - Ugyanaz az esemény több porton is megjelenhet
 - Egy akció több vezérfonálon is megjelenhet
- Általában a tesztelő feladata, hogy események és vezérfonalak segítségével biztosítsa, hogy az 5 alapelem közötti sok-sok kapcsolat helyes legyen!



Követelmények modellezése



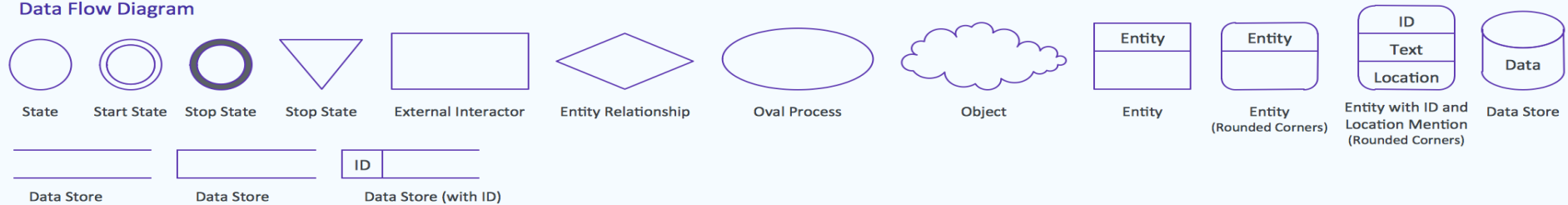


Követelmények modellezése

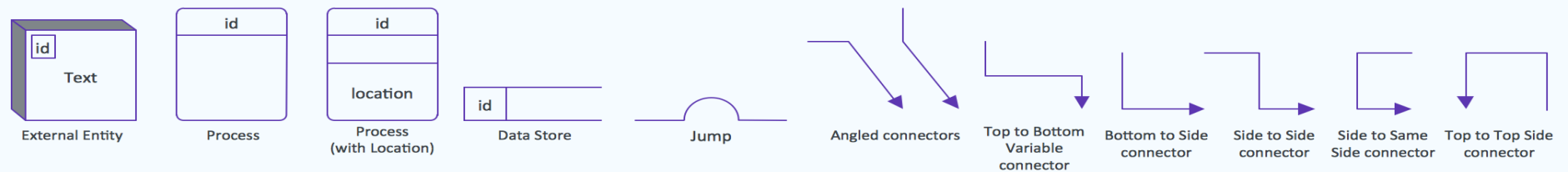
- **DÖNTÉS** kérdése, hogy milyen leírást vagy melyik modellt alkalmazzuk abból a célból, hogy a követelményeket minél jobban megértsük, leírjuk, modellezzük!
- Nincs egyetlen jó megoldás!
- A tapasztalat segíteni fog!
- Tanács: mindig maradjunk nyitottak és tanuljunk új dolgokat! Figyeljünk azokra, akiknek a mieinktől eltérő tapasztalatuk, tudásuk van!
 - Felhasználó, üzleti szakértő, tesztelő...



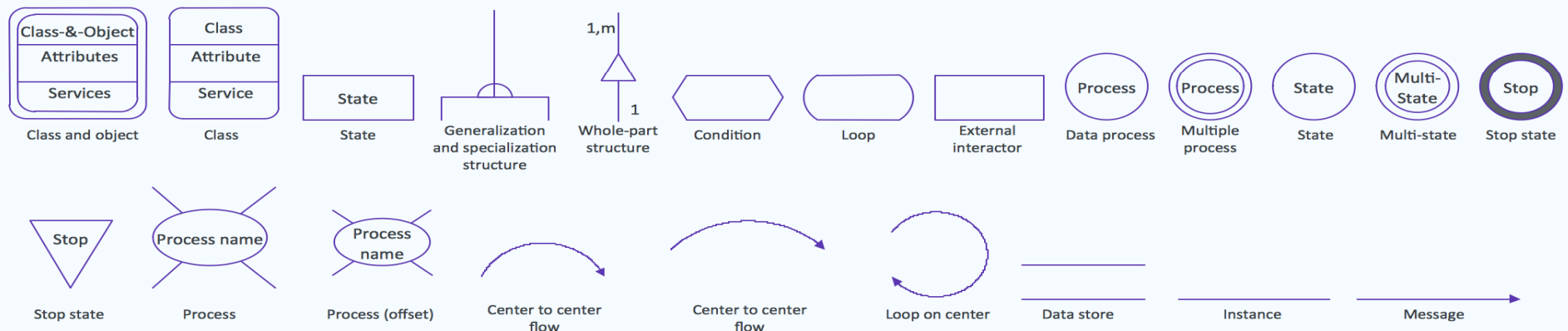
Data Flow Diagram



Gane-Sarson notation



Yourdon and Coad notation



<http://www.conceptdraw.com/How-To-Guide/data-flow-diagram-symbols>

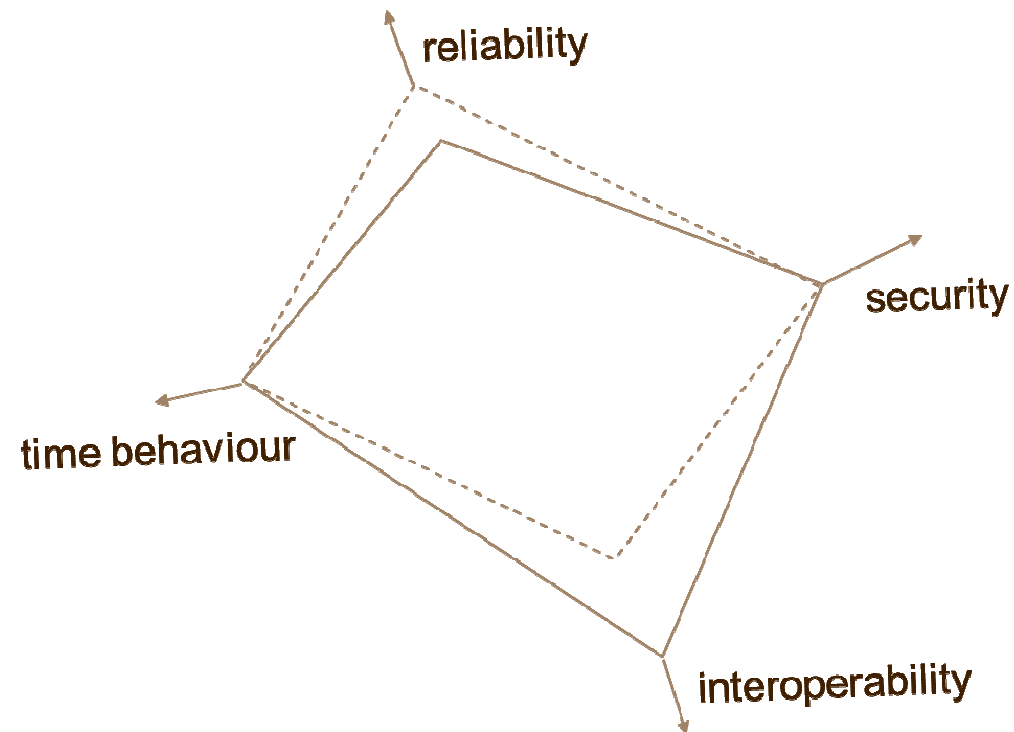


Követelmények priorizálása

- A követelményeket sorrendbe állítjuk, különböző időpillanatokban, különböző tevékenységek során, különböző kritériumok szerint.
- A priorizálás előkészítése :
 - A priorizálás céljának és korlátainak meghatározása
 - A priorizálás kritériumainak meghatározása
 - A releváns érdekelt felek azonosítása
 - A priorizálandó elemek kiválasztása
- Priorizálás: Sorrendet állít fel, valamit előnyben részesít

Követelmények priorizálása

Miért?



Követelmények priorizálása

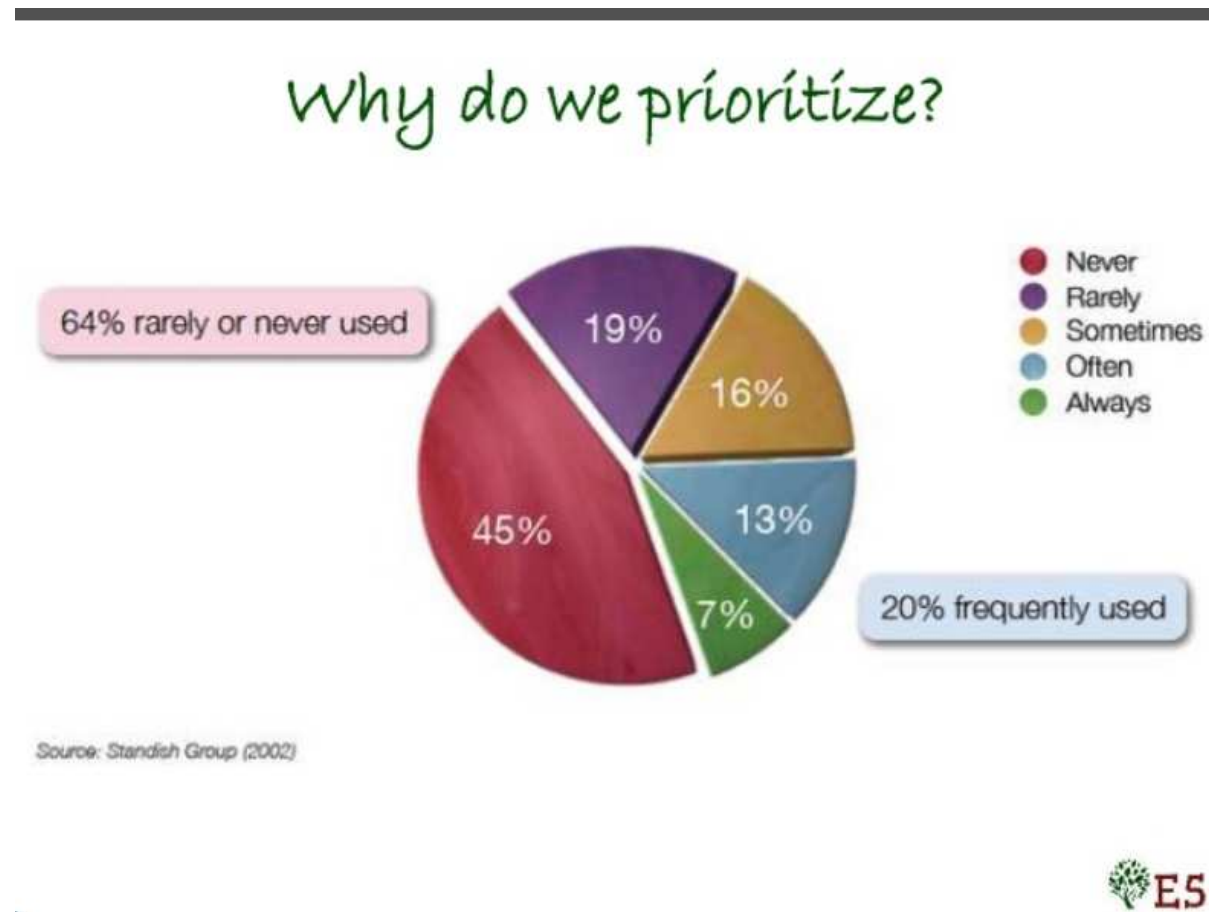
Prioritization

Based on *value*
using MoSCoW:

- M - MUST
- S - SHOULD
- C - COULD
- W - WON'T (but would like)



Követelmények priorizálása





Követelmények dokumentálása

- A korábban felsorolt leírások, modellek mind a követelmények dokumentálására szolgálnak
- Lehetnek különösen magas formalizálásra vonatkozó igények (akár szabványokban leírva)
- **Általános elvárás, hogy a követelményspecifikációból kiderüljön a rendszer célja, alapvető rendeltetése, a szoftver típusa, valamint a szükséges becsült idő és költségek.**



Követelmények dokumentálása

- Példák, szabványok, minták:

- Sokszor emlegetik az **SRS**-t: „**Software Requirements Specification**”
 - 830-1984 — *IEEE Guide to Software Requirements Specifications*. 1984.
- ISO / IEC / IEEE 29148-2011 - Systems and software engineering — Life cycle processes — Requirements engineering. 2011. pp. 1–94. ("This standard replaces IEEE 830-1998, IEEE 1233-1998)
- <http://standards.ieee.org/findstds/standard/29148-2011.html>



Követelmények dokumentálása

- Lehet egyetlen dokumentum, több dokumentum halmaza (szövegesen, diagramokkal...), egy eszközben leírva stb.
- Eszközök (példák):
 - Rational RequisitePro - <https://www.ibm.com/support/knowledgecenter>
 - <https://www.liquidplanner.com/blog/7-tools-to-gather-better-software-requirements/>
 - <http://www.capterra.com/requirements-management-software/> - top 20 requirements management tools
 - <http://www.volere.co.uk/template.htm> - diákoknak ingyenesen letölthető!
 - Stb.
- „Toolbox approach” – integrált eszközkészlet, követelmények, tervek, kód, tesztek kezelésére (pl. RUP eszközkészlete- lásd a további előadásokban)



Követelmények dokumentálása

- A követelményeket leíró dokumentumok alapul szolgálnak a szoftverfejlesztés több, soron következő fázisában, például:
 - ☐ Projekttervezés
 - ☐ Architektúra terv
 - ☐ Implementáció
 - ☐ Tesztelés
 - ☐ Változáskezelés
 - ☐ Működtetés, felhasználói dokumentáció
 - ☐ Szerződéses ügyek
 - ☐ ...
- Ezért is nagyon fontos , hogy a követelményspecifikáció „jó” legyen
- Automatizált eszközök használatával az információk konzisztenssé tehetők a követelmények különböző nézetei, valamint a követelmények és az ezekből származtatott munkatermékek között



Mitől lesz „jó” egy követelményspecifikáció?

- Világosan megfogalmazott minőségi kritériumok teljesítésétől
- Ezeket ellenőrizni / verifikálni / validálni kell!
 - Egyértelmű (két- vagy többértelműségektől mentes)
 - Konzisztens
 - Szerkezete világos
 - Módosítható és kiterjeszthető
 - Teljes
 - Követhető, hogy melyik követelmény hogyan fejlődik, bomlik részekre, milyen egyéb követelményekre van hatással, milyen változásokon megy át, mikor és hogyan törlődik stb.
 - Jól olvasható:
 - Rövid mondatokat tartalmaz
 - Minden mondat egyetlen követelményt ír le
 - “Mint <felhasználó típus>, azt szeretném, hogy <cél> azért hogy <végcél>.”
 - „Mint bankszámla tulajdonos szeretném lekérdezni a számlámat hogy az egyenleget megtudjam.”



A követelmények validálása

- A követelmény validálás során annak (vég)ellenőrzése történik, hogy a követelmények teljesítik-e a velük szemben megfogalmazott minőségi kritériumokat, azzal a céllal, hogy a hibákat – a követelményfejlesztés folyamatában minél előbb megtalálják és kijavítsák.
- A követelmények validálásának technikái:
 - ☐ Kommentelés (szakértői vélemény)
 - ☐ Inspekciók
 - ☐ Walkthrough



A követelmények menedzsmentje

- ...jelenti a követelmények gondos kezelését, kiemelten figyelve az alábbiakra:
 - ☐ Kétirányú követhetőség
 - ☐ Követelmények változásának kezelése

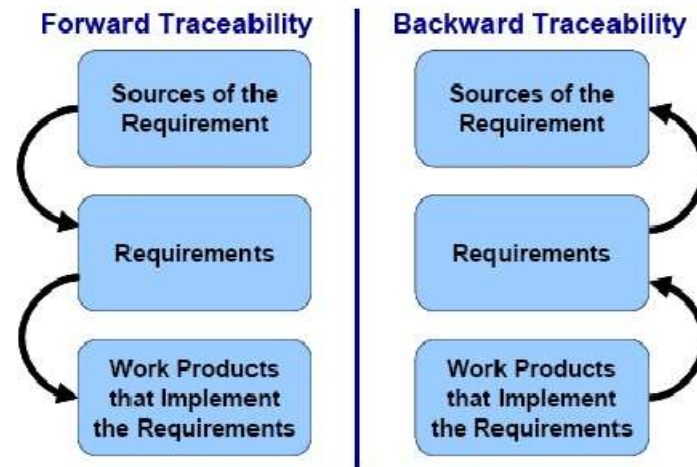


Kétirányú követhetőség

- A követelmények közötti oda-vissza , valamint horizontális és vertikális kapcsolatokat hivatott megjeleníteni
- „Oda-vissza”:
 - Vevői követelmény → termék követelmény → design elem-kód részlet → teszt eset → a működő rendszer eleme → felhasználói dokumentáció eleme ... stb. ÉS visszafele is...
 - Ez a horizontális kétirányú követhetőség
- „Vertikális”
 - Magas szintű követelmény lebontása alacsonyabb szintűekre, melyek azután horizontálisan követhetők lesznek

Kétirányú követhetőség

BI-DIRECTIONAL TRACEABILITY MATRIX

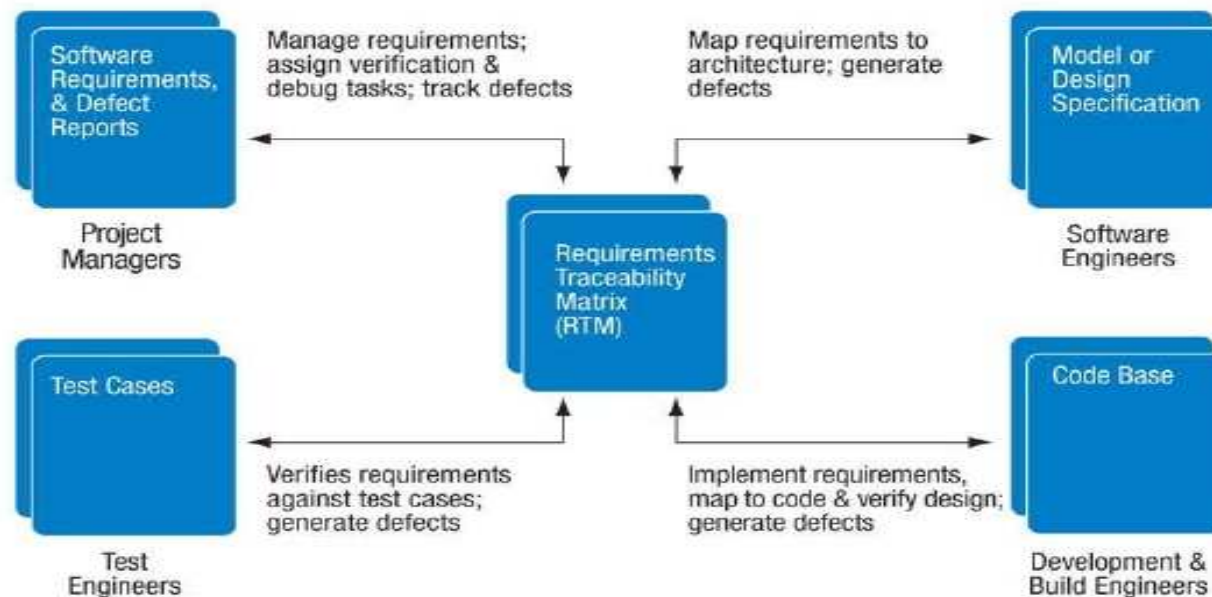


Friday, April 20, 2012

14

Kétirányú követhetőség

TRACEABILITY MATRIX



Friday, April 20, 2012

13

<https://www.slideshare.net/faizayousuf/comparison-of-sdlc-models-and-bidirectional-traceability-matrix>

Kétirányú követhetőség

EXAMPLE OF BI-DIRECTIONAL TRACEABILITY MATRIX

Requirements Source	Requirement	Design Element	Code Function	Test Case
Business Rule # 1: User can register itself on the Website and login itself.	User Registration and Login Module	Use Cases/ Process Flow Diagrams for User Registration and Login	1. Is_user 2. Is_newuser	1. User should be able to register itself on the website. 2. Email address should be unique. 3. User should be able to login after providing correct credentials.
Business Rule # 2: User can buy different products online.	Online Buying Module	Use Cases/Flow Diagrams for Online Buying Module	1. Product_Stock 2. Product_Price 3. Product_Delivery	1. Ensure before checking out that the product is in stock. 2. Checkout the product, add in cart and provide delivery information. 3. Pay dues online by different payment gateways. 4. Product record is updated automatically

Friday, April 20, 2012

Kétirányú követhetőség SPI modellekben



- Nagyon hangsúlyos a CMMI és Automotive SPICE modellekben.

		Bug		New Feature			Sub-task				Technical task		Test	
		MAT-1	MAT-9	MAT-10	MAT-13	MAT-14	MAT-2	MAT-3	MAT-4	MAT-6	MAT-5	MAT-7	MAT-8	MAT-12
Bug	MAT-1		*(1)	*(1)	*(1)		*(1)	*(1)	*(1)	*(1)	*(1)	*(1)		
	MAT-9	*(1)					*(1)						*(1)	
New Feature	MAT-10	*(1)								*(1)		*(1)		
	MAT-13	*(1)				*(1)								
	MAT-14					*(1)			*(1)					
Sub-task	MAT-2	*(1)	*(1)											
	MAT-3	*(1)								*(1)				
	MAT-4	*(1)				*(1)								
	MAT-6	*(1)		*(1)				*(1)					*(1)	
Technical task	MAT-5	*(1)												
	MAT-7	*(1)		*(1)										
Test	MAT-8		*(1)						*(1)					
	MAT-12													



ISO/IEC 15504 - SPICE

* Number of Level
Changes Description
New Added
Added
Changed
Removed
Not Affected

SP 1.4 Maintain Bidirectional Traceability of Requirements

Maintain bidirectional traceability among requirements and work products.

The intent of this specific practice is to maintain the bidirectional traceability of requirements. (See the definition of "bidirectional traceability" in the glossary.) When requirements are managed well, traceability can be established from a source requirement to its lower level requirements and from those lower level requirements back to their source requirements. Such bidirectional traceability helps to determine whether all source requirements have been completely addressed and whether all lower level requirements can be traced to a valid source.

Baselines provide a stable basis for the continuing evolution of configuration items.

An example of a baseline is an approved description of a product that includes internally consistent versions of requirements, **requirement traceability matrices**, design, discipline-specific items, and end-user documentation.

Baselines are added to the configuration management system as they are developed. Changes to baselines and the release of work products built

Configuration Management (CM)

<http://www.confluence.kostebekteknoji.com/display/FB/Create+Bi-directional+Traceability+Matrices+and+Baselines+in+JIRA>



Követelmények változásának kezelése

- Lehetővé teszi, hogy az elkerülhetetlen módosítások miatt ne csússzon ki a kezünkből a szoftver követelményrendszere
- Változások:
 - ☐ Új követelmény
 - ☐ Meglévő követelmény törlése
 - ☐ Meglévő követelmény módosítása
 - Gyakran megesik, hogy egy régi rendszeren kell módosítani „egy kicsit”, de a követelményspecifikáció egészen mást ír le, mint amit a működő rendszer produkál
- Vizsgálni kell a követelmények változásának hatását a többi követelményre



Követelmények változásának kezelése

- Módja: verziókezelés , alapverziók (baseline) létrehozása
- Esetenként nagyon formális lehet
 - ☐ „Change Control Board”
 - ☐ Formális kérelem követelmények változására
 - ☐ Hatáselemzés
 - ☐ A hatás következményeinek, elfogadhatóságának elbírálása
 - ☐ A változaskérelem elfogadása vagy elutasítása
 - ☐ Szigorú dokumentálás



Példák

- A következő diákon követelményeket leíró dokumentumok, modellek láthatók.



Példa követelmény leírására

Leírás természetes nyelven:

- ☐ “A/szereplő (Actor) csináljon (valamit),
azáltal, hogy (hogyan; magyarázzuk el, hogyan indíthatja a
felhasználó ezt a folyamatot),
Azért, hogy / abból a célból, hogy (miért; magyarázzuk el a
követelmény megvalósulásának előnyeit).
- ☐ **Példa:** “A rendszer lehetővé teszi, hogy a felhasználók
regisztráljanak, nevüket és jelszavukat megadva, azért, hogy így
hozzáférést kapjanak a rendszerhez.”

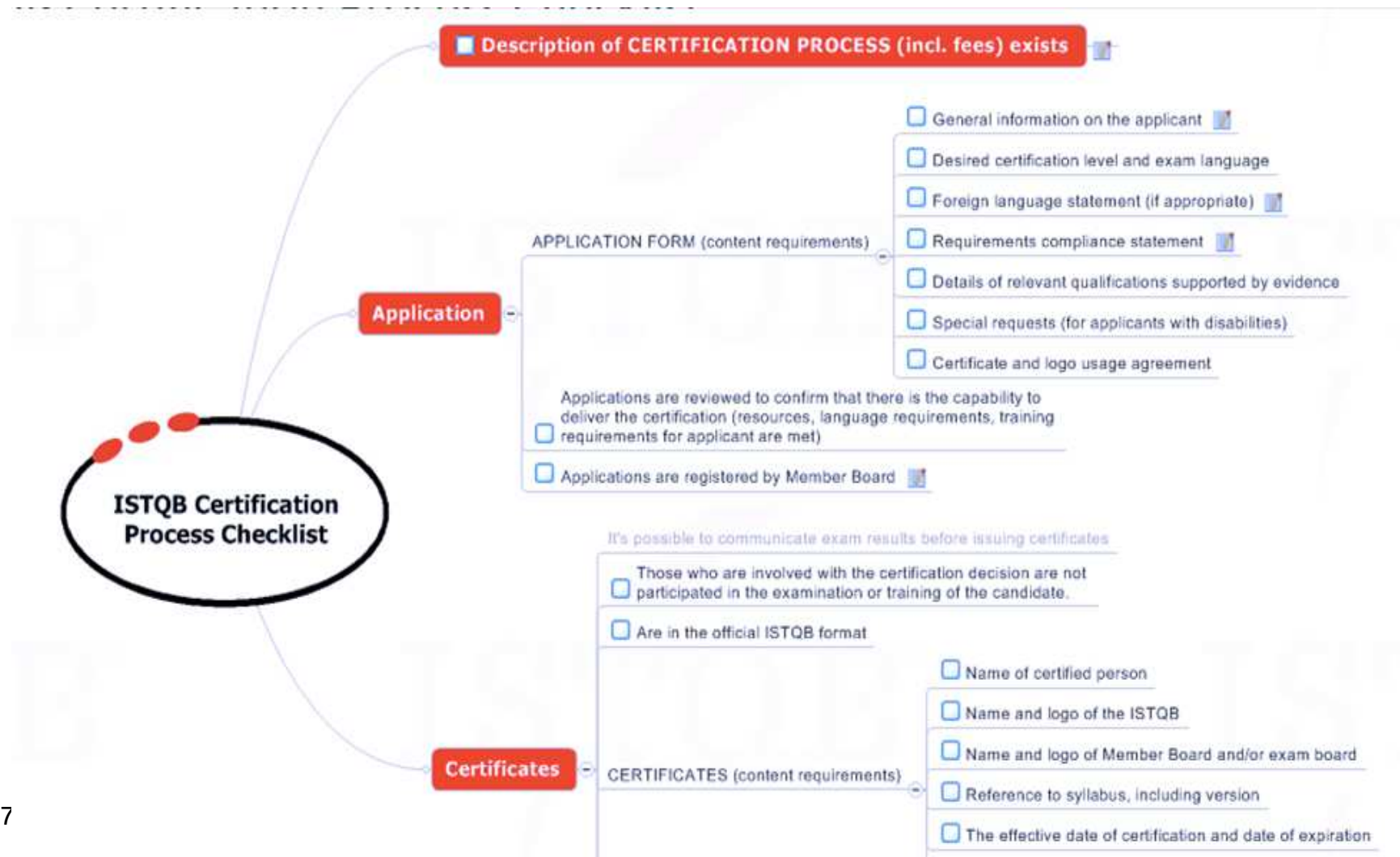
Példa követelmény leírására

■ Strukturált nyelven való leírás

<i>Insulin Pump/Control Software/SRS/3.3.2</i>	
Function	Compute insulin dose: Safe sugar level
Description	Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units
Inputs	Current sugar reading (r2), the previous two readings (r0 and r1)
Source	Current sugar reading from sensor. Other readings from memory.
Outputs	CompDose – the dose in insulin to be delivered
Destination	Main control loop
Action	CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.
Requires	Two previous readings so that the rate of change of sugar level can be computed.
Pre-condition	The insulin reservoir contains at least the maximum allowed single dose of insulin
Post-condition	r0 is replaced by r1 then r1 is replaced by r2
Side-effects	None

Követelményspecifikáció.

Példa: Mind Map





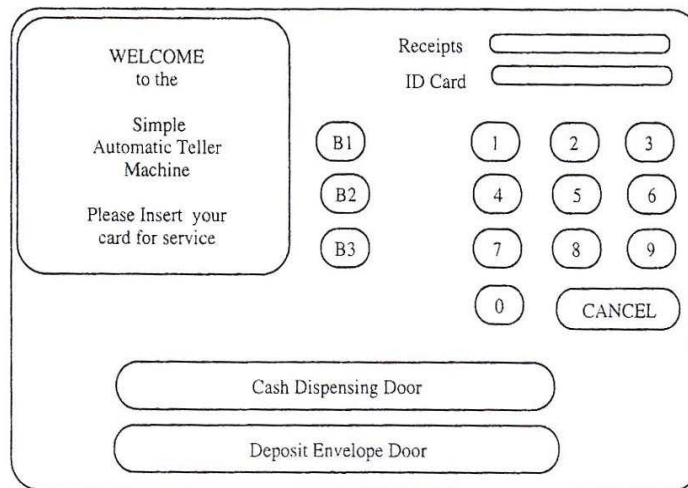
Követelményspecifikáció.

Példa: Mind Map

- **Gondolattérkép:** egy központi ötlet, vagy szó körül más szavak, ötletek, feladatok, vagy egyéb egységek elrendezését, illetve szerkesztését szolgáló ábra.
 - Gondolattérképeket ötletek létrehozására, megjelenítésére, rendezésére és osztályozására használnak, illetve segíthetnek tanulmányok, szervezetek, problémamegoldások és döntések meghozatalában és leírásában.
 - → mind map

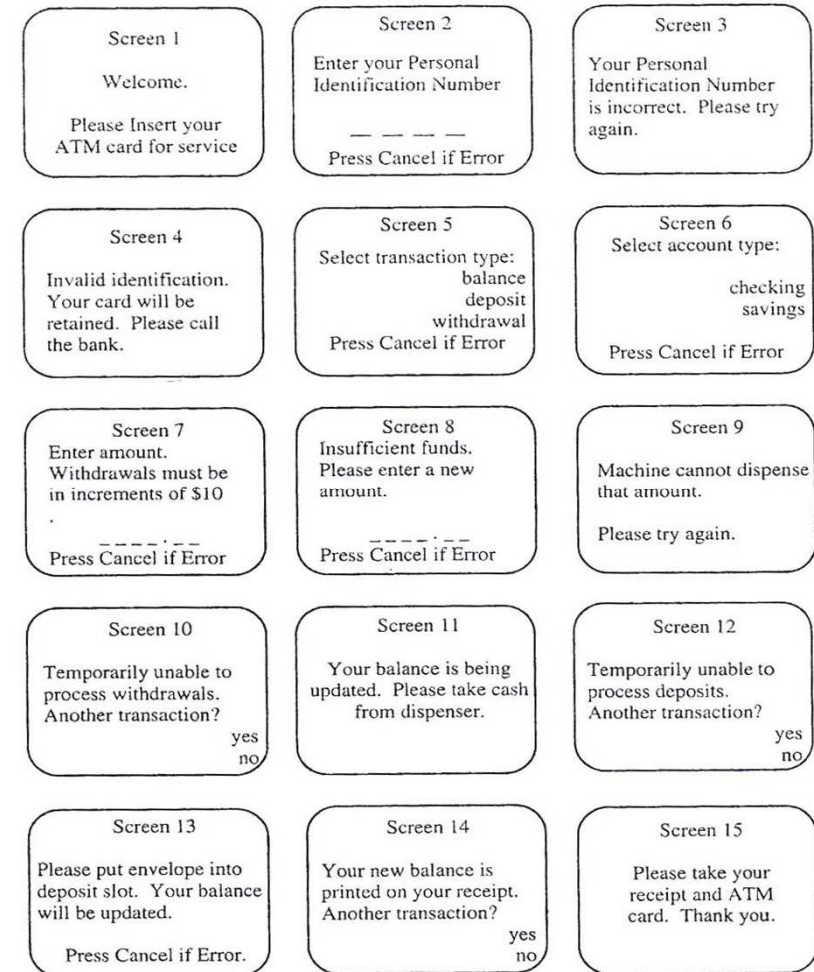


Követelmények modellezése



A rendszer megértése

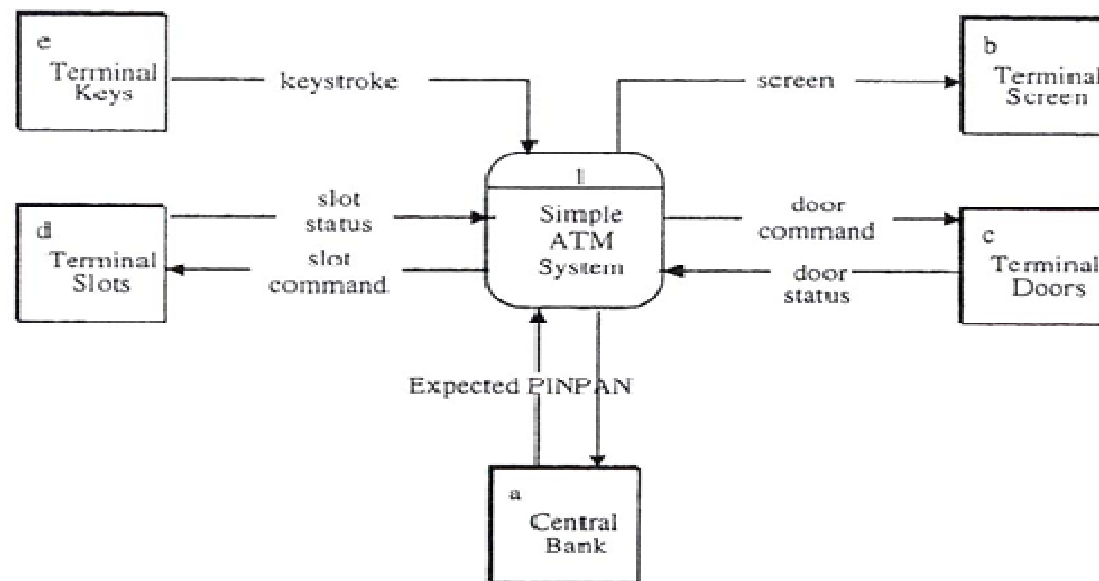
„Simple ATM” example from: Paul C. Jorgensen: Software testing. A Craftsman’s approach. CRC Press, 2013



A rendszer képernyőinek modellezése

Követelmények modellezése

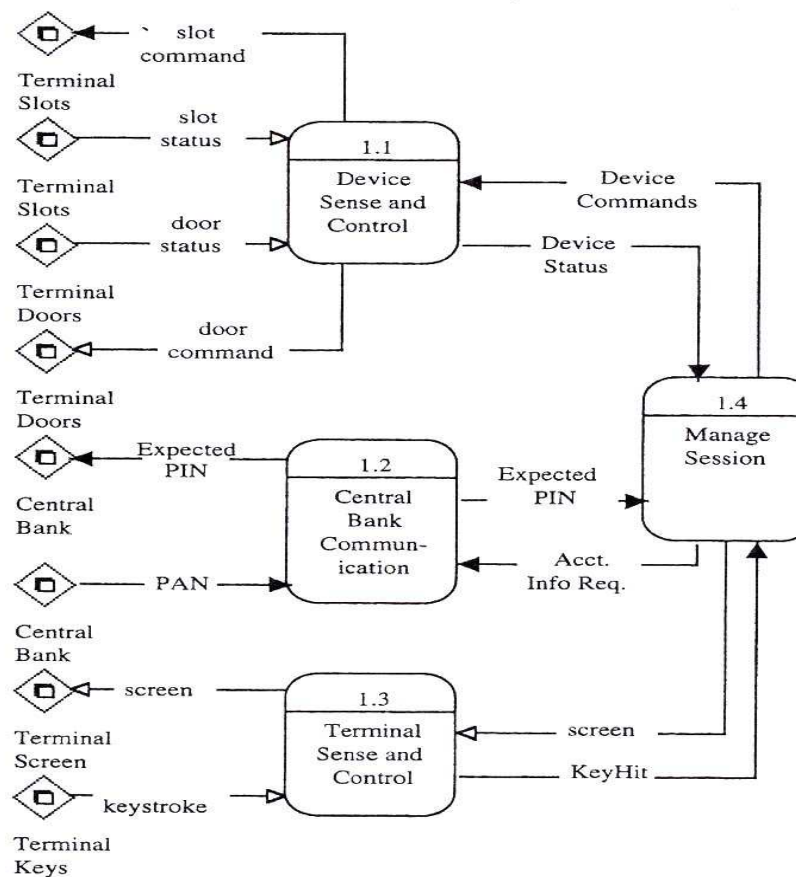
■ Kontextus diagram



Context diagram of the SATM system

Követelmények modellezése

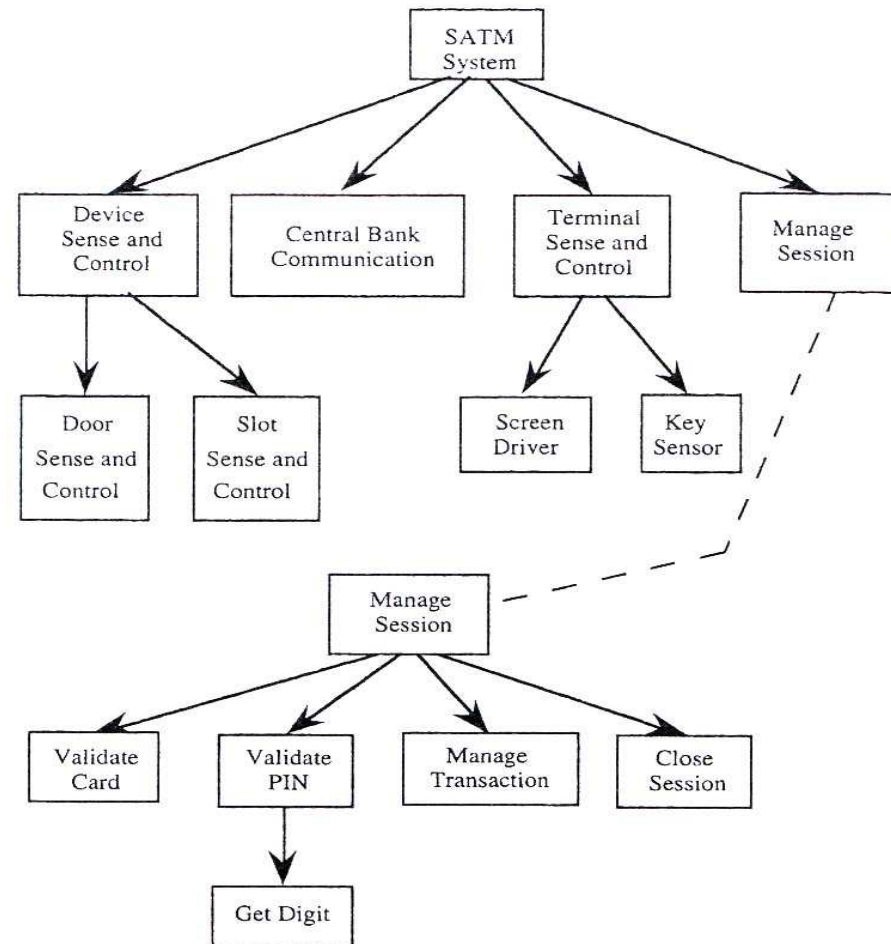
- **Adatfolyam diagram**



Level 1 data flow diagram of the SATM system

Követelmények modellezése

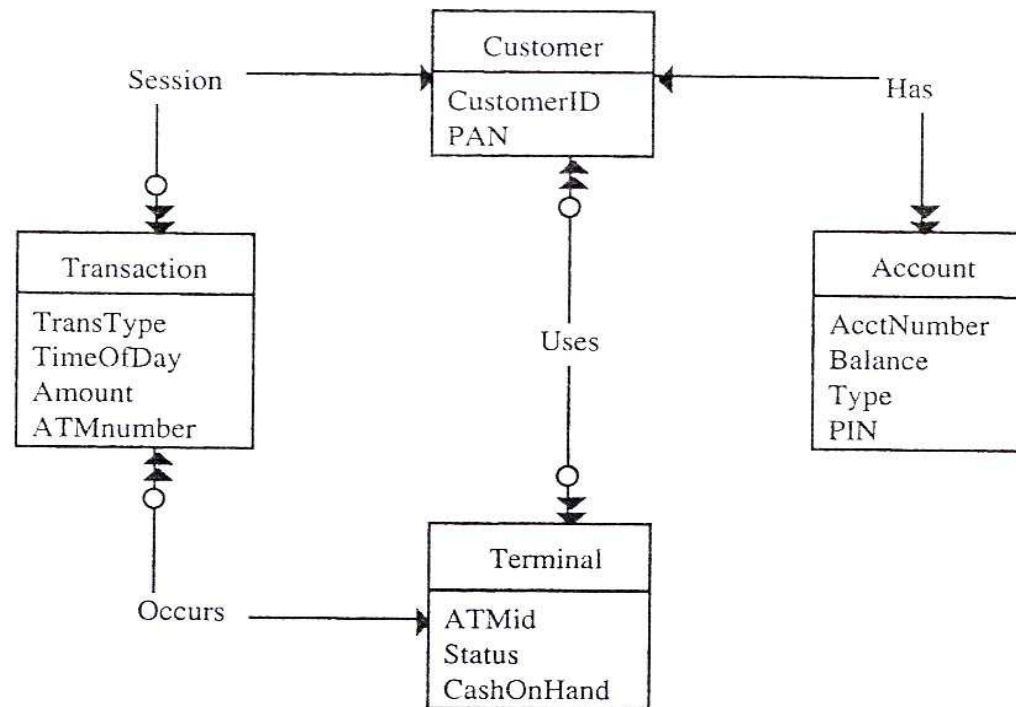
- Funkcionális dekompozíció



A decomposition tree for the SATM system

Követelmények modellezése

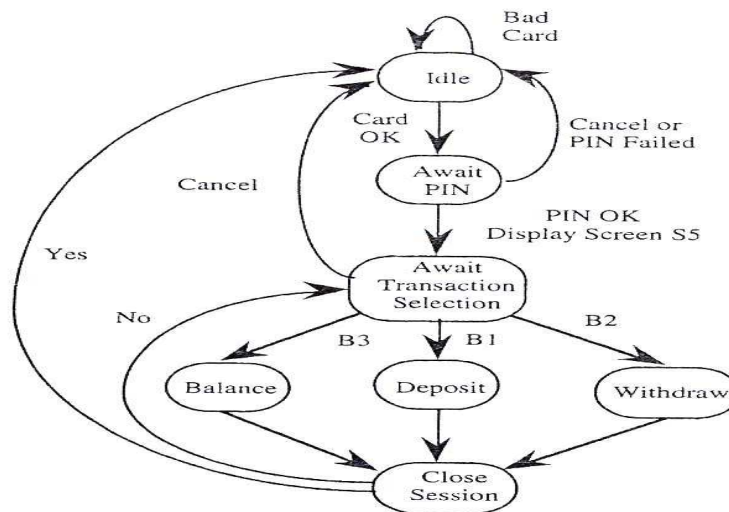
- Entity/ relationship diagram (ERD)



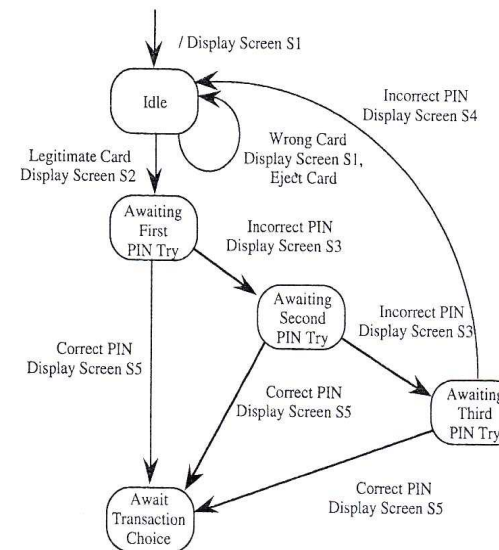
Entity/ relationship model of the SATM system

Követelmények modellezése

- Működési modell
 - Itt, véges automaták hierarchiája



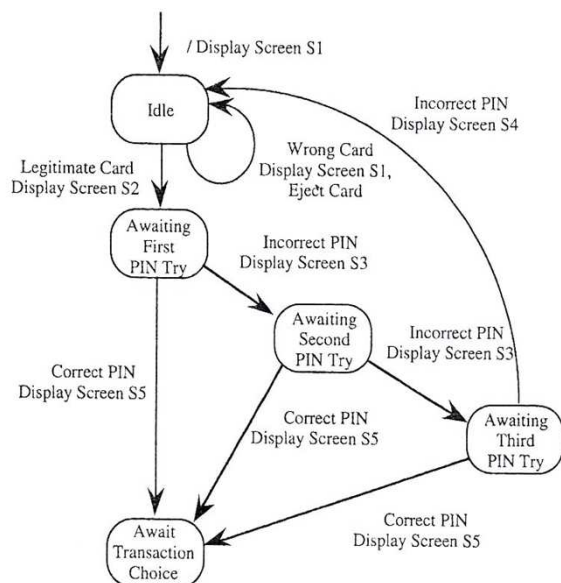
Upper-level SATM finite state machine



PIN entry FSM

Követelmények modellezése

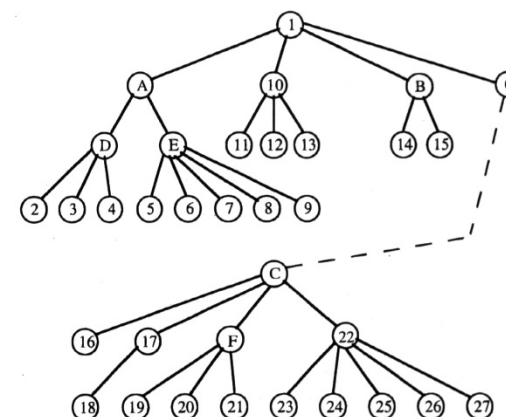
Példa: Követelmények felismerése és lebontása egy pénzkidő automata esetén



A komponensek szintje az ábra szerint számozva. Olvashatóság céljából új nevet kaptak.

Unit-ok és rövidített neveik

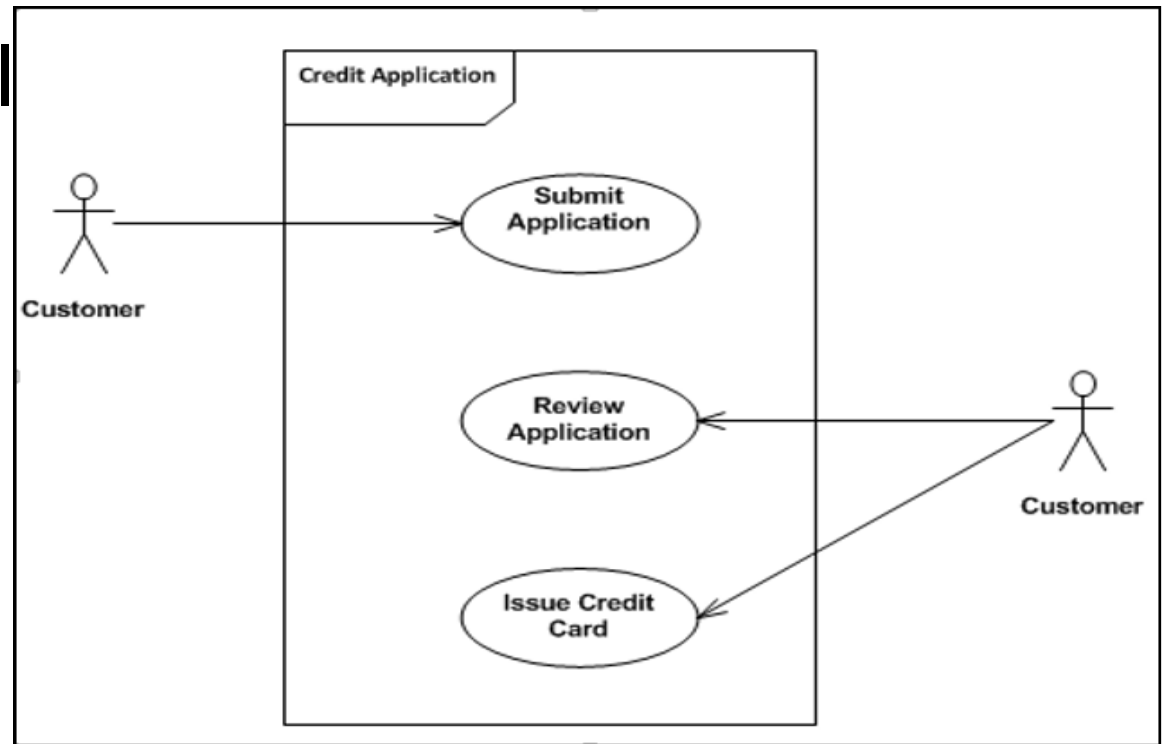
Unit száma	Szint száma	Unit neve
1	1	SATM System
A	1.1	Device Sense & Control
D	1.1.1	Door Sense & Control
2	1.1.1.1	Get Door Status
3	1.1.1.2	Control Door
4	1.1.1.3	Dispense Cash
E	1.1.2	Slot Sense & Control
5	1.1.2.1	WatchCardSlot
6	1.1.2.2	Get Deposit Slot Status
7	1.1.2.3	Control Card Roller
8	1.1.2.4	Control Envelope Roller
9	1.1.2.5	Read Card Strip
10	1.2	Central Bank Comm.
11	1.2.1	Get PIN for PAN
12	1.2.2	Get Account Status
13	1.2.3	Post Daily Transactions
B	1.3	Terminal Sense & Control
14	1.3.1	Screen Driver
15	1.3.2	Key Sensor
C	1.4	Manage Session
16	1.4.1	Validate Card
17	1.4.2	Validate PIN
18	1.4.2.1	GetPIN
F	1.4.3	Close Session
19	1.4.3.1	New Transaction Request
20	1.4.3.2	Print Receipt
21	1.4.3.3	Post Transaction Local
22	1.4.4	Manage Transaction
23	1.4.4.1	Get Transaction Type
24	1.4.4.2	Get Account Type
25	1.4.4.3	Report Balance
26	1.4.4.4	Process Deposit
27	1.4.4.5	Process Withdrawal



Az ATM funkcionális dekompozíciós fája

Követelmények modellezése

■ Use case modell



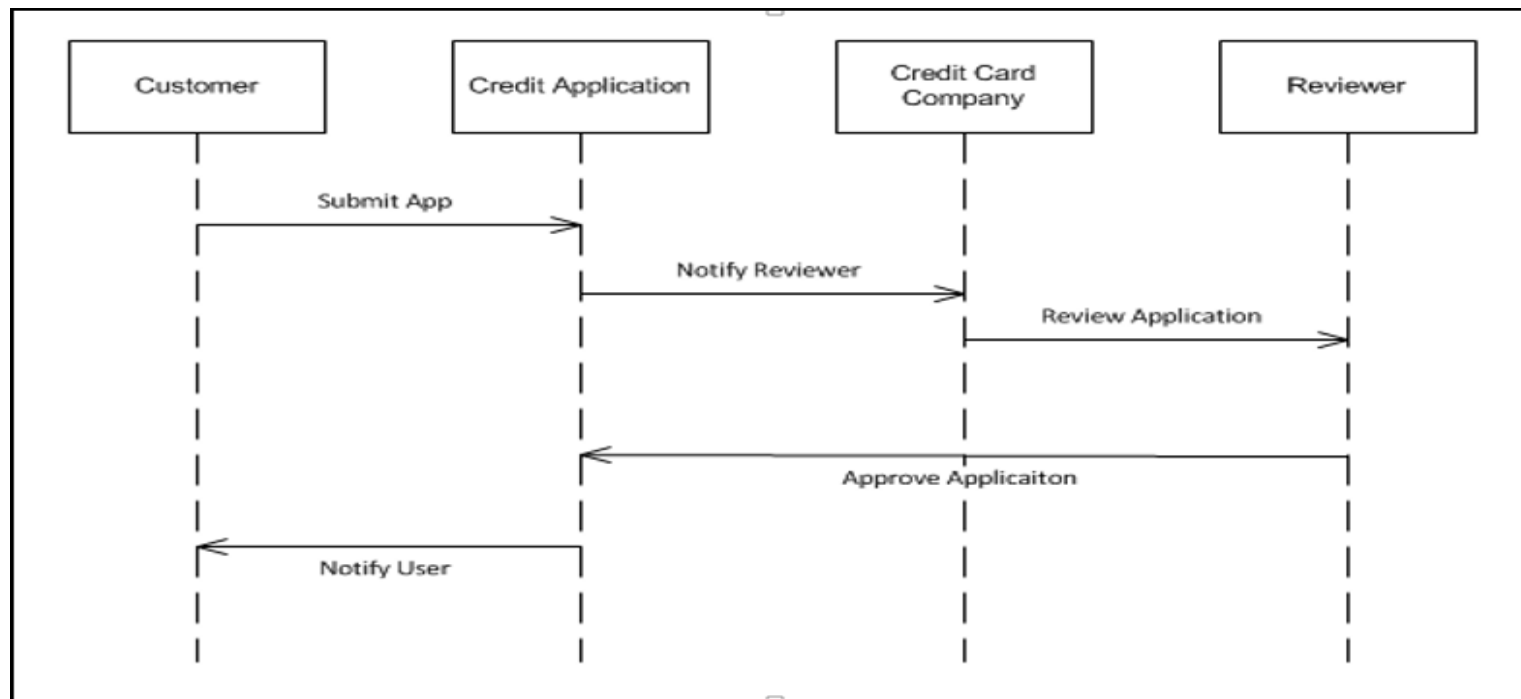
Describes the Credit Card application submission process

<https://www.liquidplanner.com/blog/7-tools-to-gather-better-software-requirements/>

Követelmények modellezése

■ Szekvencia diagram

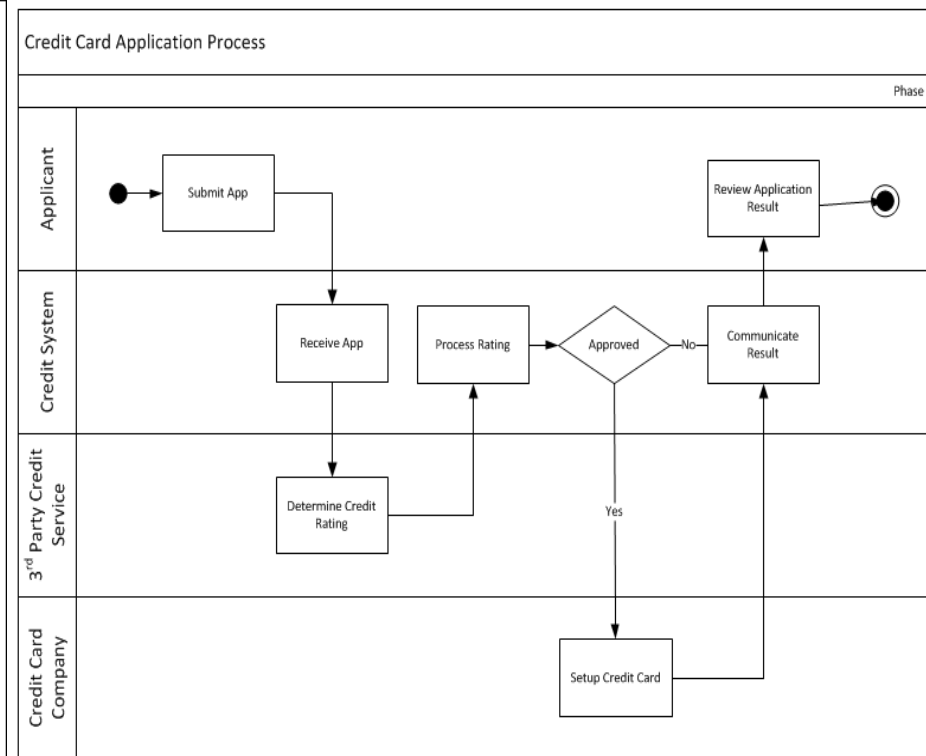
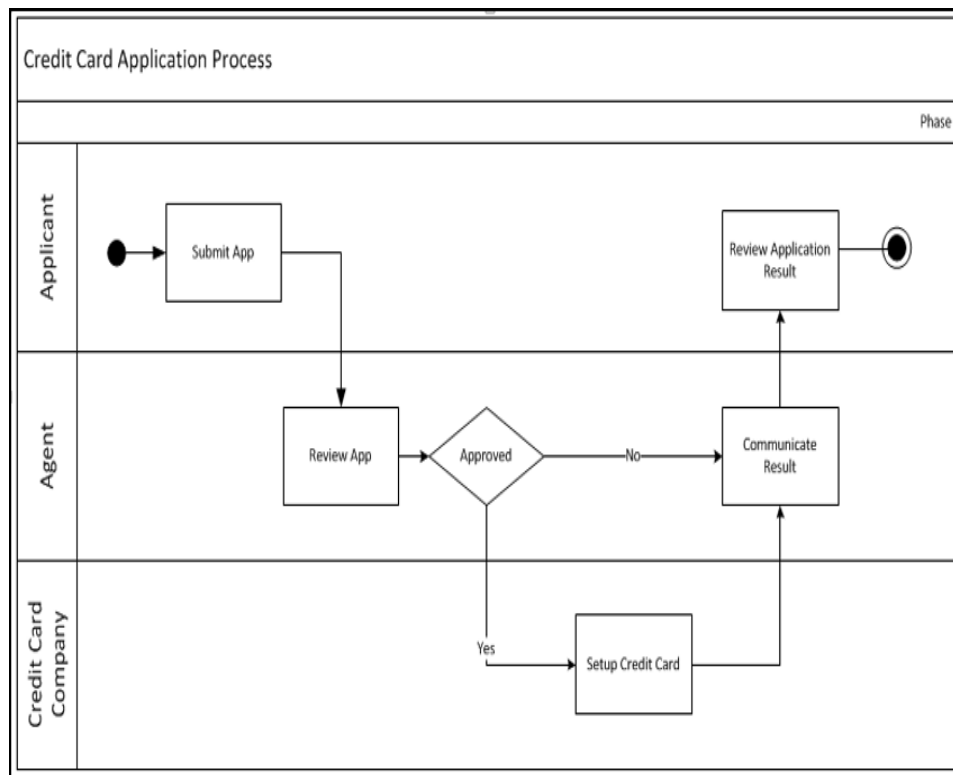
- shows the interactions between objects over time



Követelmények modellezése

■ As –Is és To-Be folyamatmodellek

□ A változások megértésére



Követelmények agilis környezetben

How are requirements changed in Agile compared with other methodologies (RUP, Waterfall)?

- Became smaller.
- Became less precise.
- Less time for elaborating them.
- Easy to change or extend.
- Now they encourage discussions instead of describing the final state.



<https://www.slideshare.net/E5Trainings/requirements-in-agile-conference>



Követelmények agilis környezetben

Epic	Story	Version	Sprint
Large body of work, contains stories	Smallest unit of work, also known as a task	The release of software to the customer	Iteration where team does the work



<https://www.atlassian.com/agile/delivery-vehicles>



Követelmények agilis környezetben

- **Felhasználói történet:** magas szintű felhasználói, vagy üzleti követelmény, amelyet általában az agilis szoftverfejlesztés során használnak. Jellemzően egy, vagy több, hétköznapi, vagy üzleti nyelven megfogalmazott mondatot tartalmaz, amely leírja, hogy a felhasználónak milyen funkcionalításra van szüksége, vagy bármilyen egyéb nem-funkcionális követelményt fogalmaz meg, továbbá tartalmazza az átvételi kritériumot is.

→ *user story*



Követelmények agilis környezetben

■ User story

- ☐ Az XP-ből eredeztethető
- ☐ Egységnyi munkát ír le
- ☐ A felhasználói követelményeket könnyű leírni vele
- ☐ Segít a csapaton belül megérteni a követelményt
- ☐ A követelményt a felhasználó szemszögéből írja le
- ☐ Ki , mit akar csinálni, és miért?



Követelmények agilis környezetben

■ Példák user story-kra:

- ☐ A diákok havi parkolási kártyát vásárolhatnak on-line módon.
- ☐ A parkolási kártyákat bankkártyával lehet kifizetni.
- ☐ A parkolási kártyákat PayPal rendszerben lehet kifizetni.
- ☐ A tanárok jegyeket írhatnak be a diákoknak.
- ☐ A diákok lekérdezhetik az aktuális laborbeosztásukat .
- ☐ A diákok hivatalos jegyzetet rendelhetnek.
- ☐ A diákok csak olyan előadásokra jelentkezhetnek, amelyekre megfelelő előfeltételekkel rendelkeznek.
- ☐ A jegyzeteket on-line el lehet érni, standard böngészővel.



Követelmények agilis környezetben

■ User story card

173. Students can purchase parking passes.

Priority: ~~High~~ 8
Estimate: 4

173

As a student I want to purchase a parking pass so that I can drive to school

Priority: ~~High~~ Should
Estimate: 4

<http://www.agilemodeling.com/artifacts/userStory.htm>



Követelmények agilis környezetben

Front of Card

173

As a student I want to purchase a parking pass so that I can drive to school

Priority: ~~High~~ Should

Estimate: 4

Back of Card

Confirmations:

~~The student must pay the card out~~

One pass for one month is issued at a time

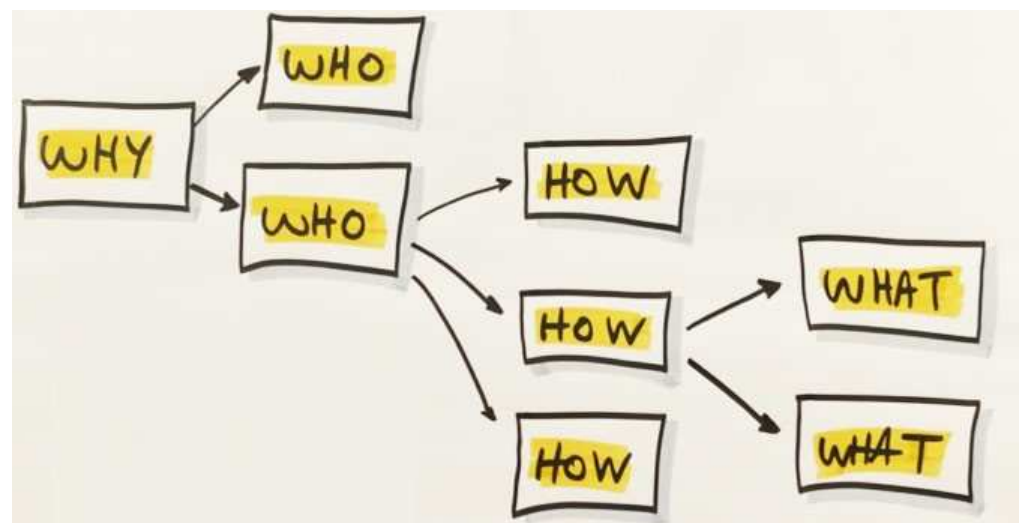
The student will not receive a pass if the payment isn't sufficient

The person buying the pass must be a currently enrolled student.

The student may only buy one pass per month.

Copyright 2005-2009 Scott W. Ambler

Követelmények egyeztetése agilis környezetben



<http://satalyst.com/requirements-elicitation-using-impact-mapping/>

Követelmények agilis környezetben

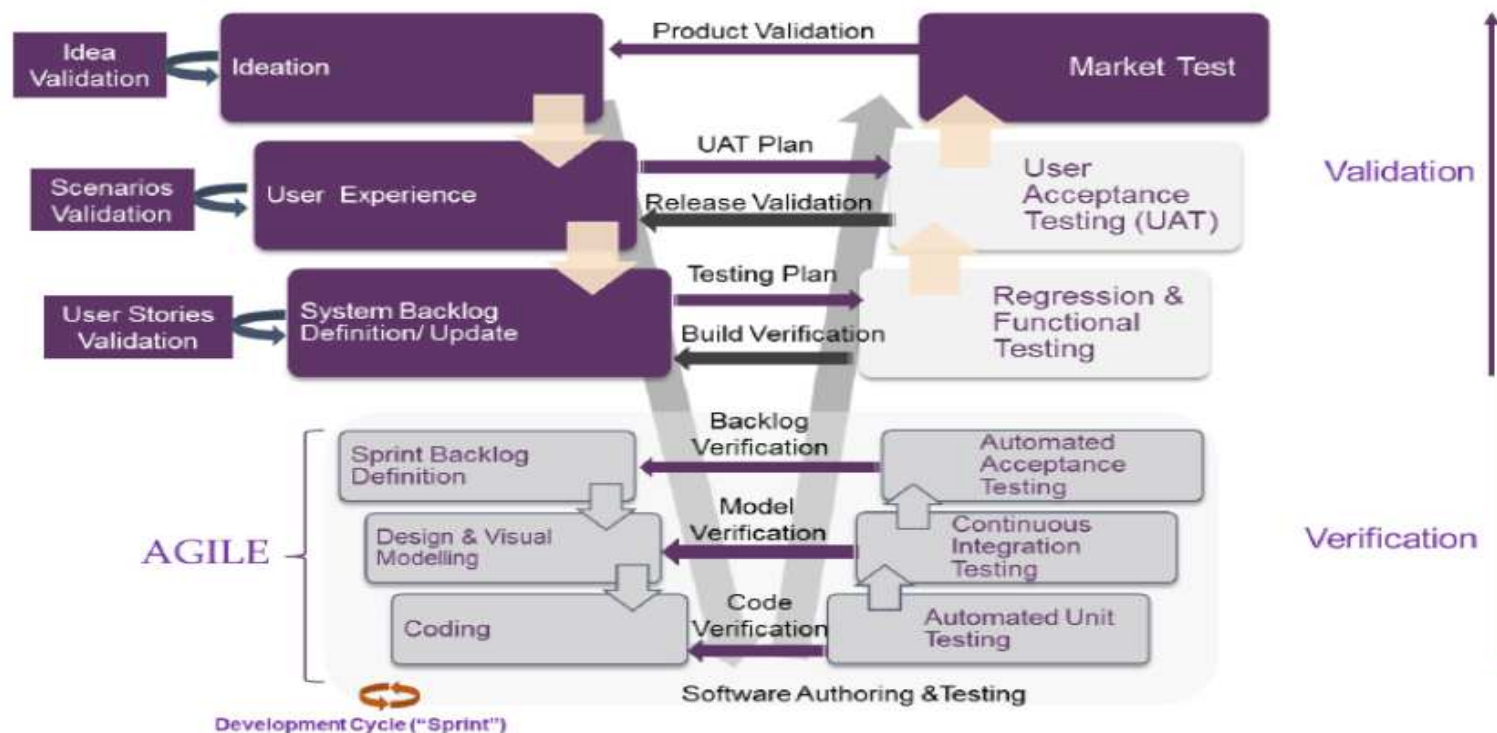


Figure 3-2. The CloudTeams methodology expanded with an Agile Software Development methodology

Követelmények agilis környezetben

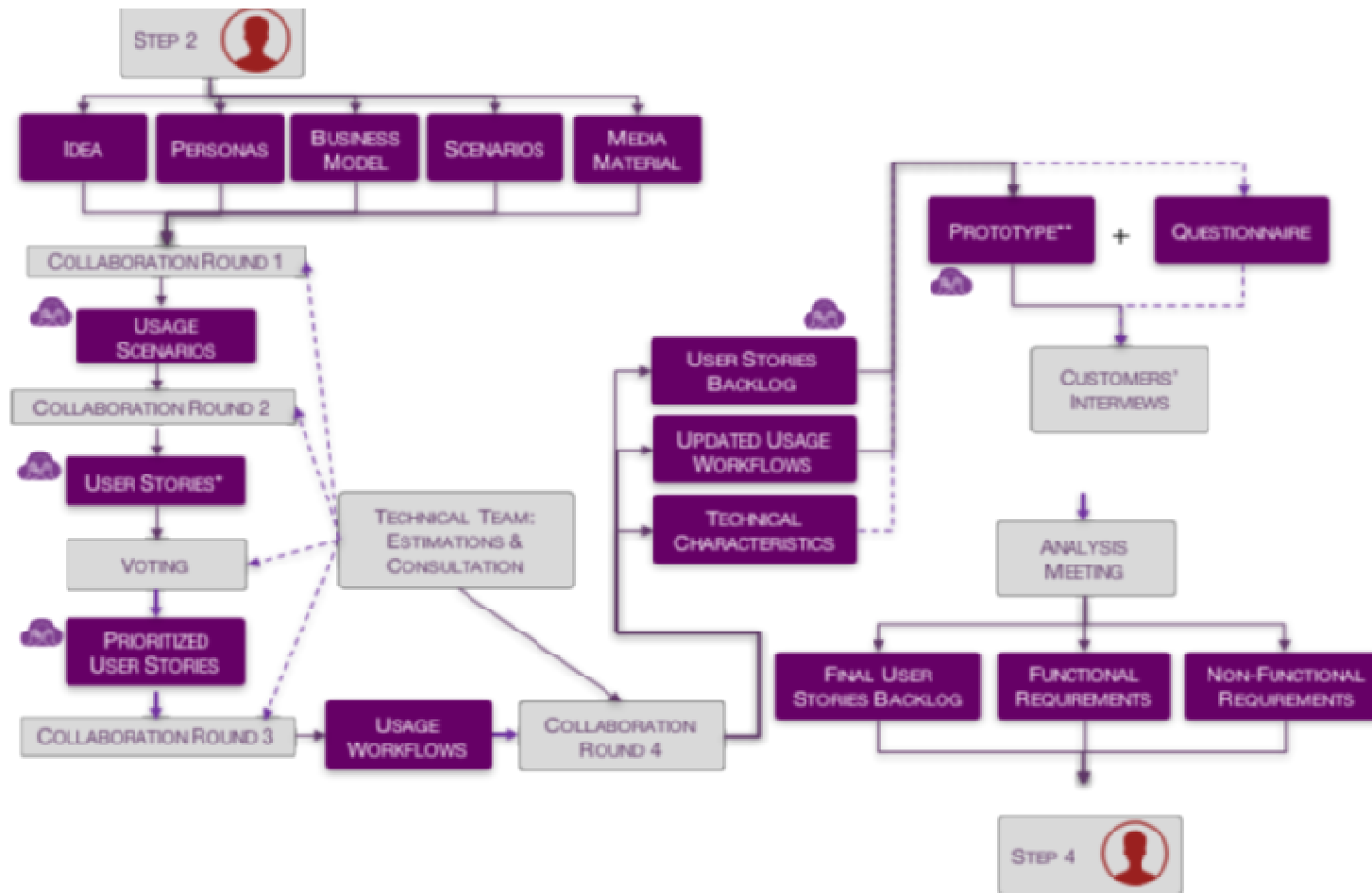


Figure 5-7. System backlog definition and User Stories Validation

Követelmények priorizálása agilis környezetben

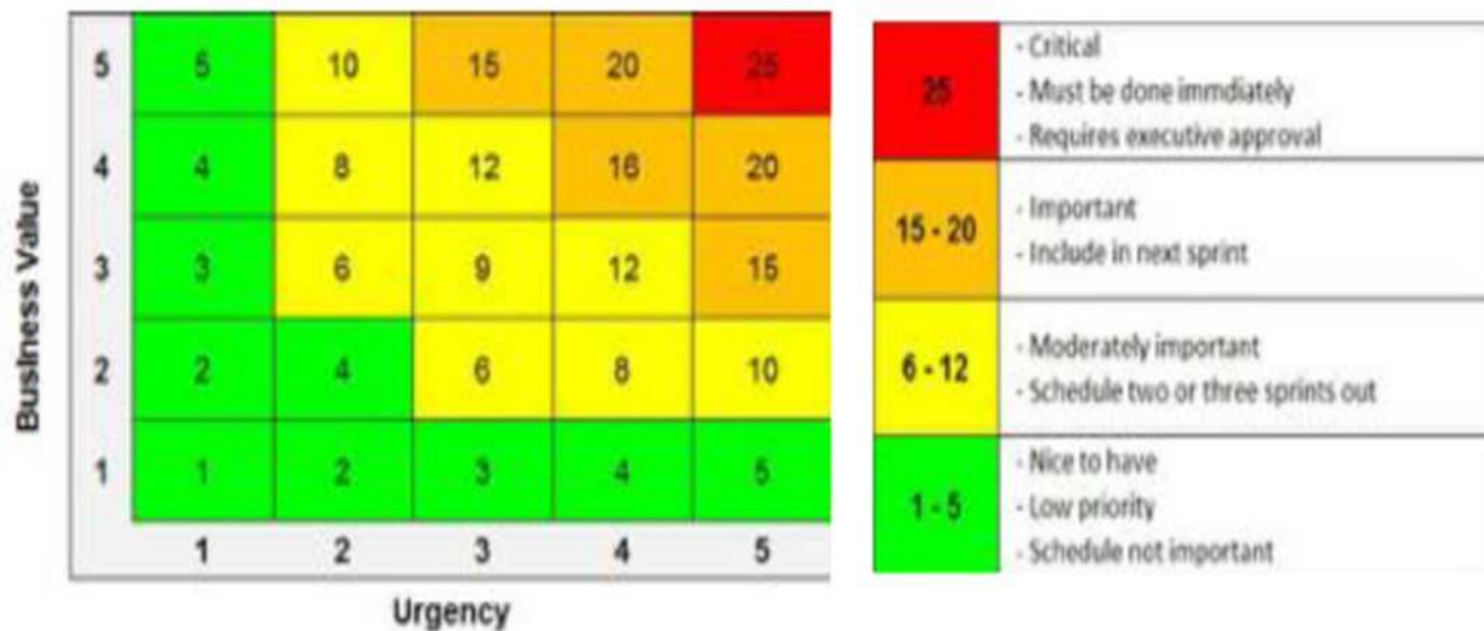


Figure 5-9. Putting the user stories in priority

User story-hierarchia- példa

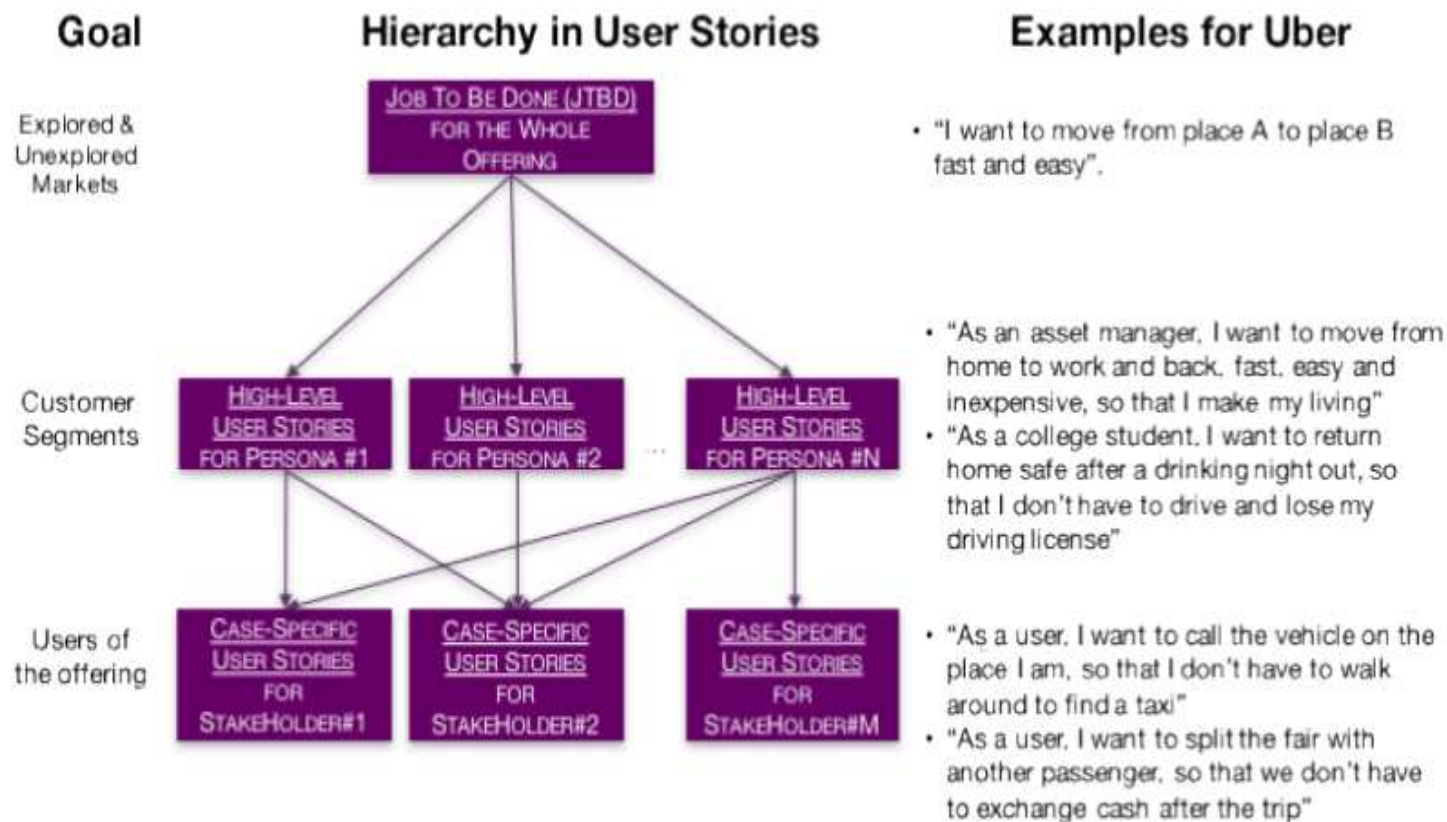
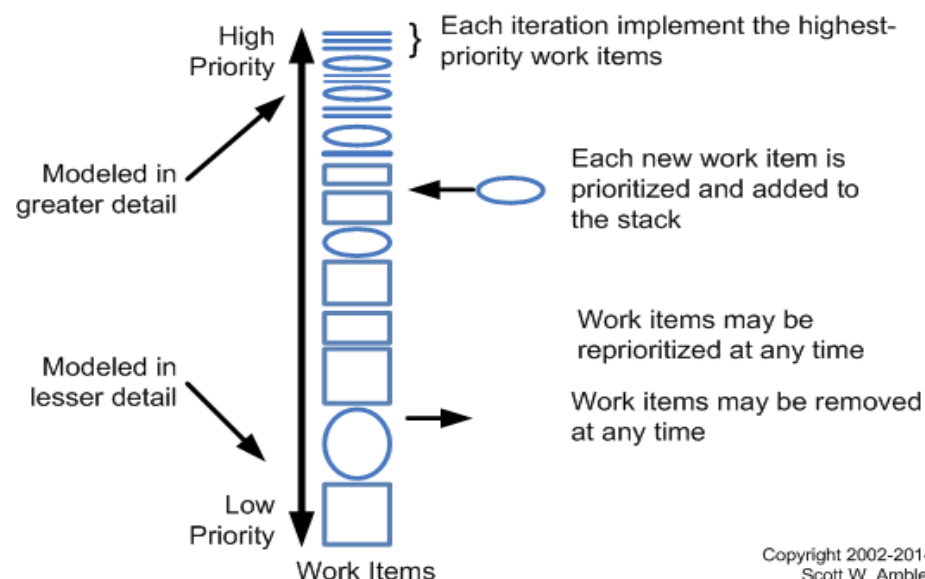


Figure 5-8. Hierarchy in User Stories, relation with the goal of each level, and relative examples

Követelmények változásának kezelése agilis környezetben

- Agilis környezetben is szükséges a fegyelmezett változáskezelés!



<http://agilemodeling.com/essays/changeManagement.htm>

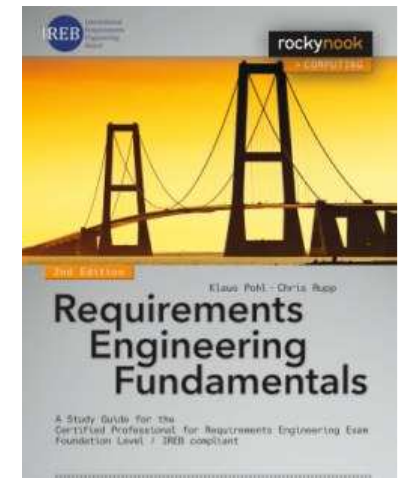


Következtetés (?)

- Jó követelményeket írni, melyek alapján jó szoftvert lehet tervezni és fejleszteni – **nem könnyű.**
- Fontos, hogy korábbi tapasztalatokból tanuljunk!

Források

- In this course we use :
 - Terms and definitions of IREB
(International Requirements Engineering Board), contained in the publicly available material
<https://www.ireb.org/en>
<https://www.ireb.org/en/downloads/#cpred-glossary>
 - A Glossary of Requirements Engineering Terminology Version 1.6 May 2014
 - Foundation Level Syllabus, Version 2.2.1 , July 24th 2017
https://www.ireb.org/content/downloads/2-syllabus-foundation-level/ireb_cppe_syllabus_fl_en_v221.pdf
 - Paul C. Jorgensen: Software testing. A Craftsman's approach. CRC Press, 2002.
-
- **Requirements Engineering Fundamentals**
Klaus Pohl, Chris Rupp
2nd edition
Rocky Nook Inc. (April 2015)
English, 184 Pages
Paperback
ISBN: 978-1-937538-77-4
Aligned to the syllabus of CPRE Foundation Level 2.2





Miről volt szó...

- A követelmények fontossága
- Requirement Engineering / Követelmények fejlesztése, követelmények kidolgozása
- Követelmények forrásai
- Követelmények felmérése, elemzése, priorizálása
- Követelmények típusai
- Követelmények modellezése, dokumentálása
- Követelménymenedzsment
 - Kétirányú követhetőség
 - Követelmények változásának kezelése
- Példák különböző követelményekre
- Követelmények agilis környezetben