



Szoftvertchnológia

4. Követelmények kezelése

BSc kurzus

Dr. Balla Katalin



- A 2018.10.19-i előadást Hargitai Zsolt tartja
 - Az eredeti diasort néhány helyen módosította
- Bemutatkozás
 - Lufthansa Systems Hungária Kft.
 - Tesztelői, tesztelési részleg vezetői és tesztelési folyamat fejlesztéséhez kapcsolódó tapasztalat
 - Magyar Szoftvertesztelői Tanács Egyesület
 - ISTQB
 - IREB
 - TMMI, TPI Next szakértő
 - CMMI



Tartalom

- A követelmények definíciója és fontossága
- Követelmények fejlesztése
- Követelmények felmérése, egyeztetése
- Követelmények típusai
- Követelmények dokumentálása, modellezése
- Követelmények kezelése, karbantartása
- 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



Követelményfejlesztés (RE)

- A követelmények meghatározásának és menedzselésének szisztematikus megközelítése, a következő célokkal:
 - A helytálló, megfelelő követelmények **megismerése**, az érdekelt felek **egyetértésének** elérése ezekre a követelményekre vonatkozóan, a követelmények szabványos **dokumentálása**, a követelmények folyamatos **menedzselése**.

Követelményfejlesztés (Requirements engineering, RE)



- A RE négy alaptevékenysége:
 - Meghatározás, egyeztetés, felmérés (elicitation)
 - Dokumentálás (documentation)
 - Validáció, egyeztetés (validation/negotiation)
 - Követelmények menedzsmentje (management)

- Ezen tevékenységek elvégzésére idő és erőforrások szükségesek!



Követelményekkel kapcsolatos problémák

- A felhasználók nem tudják, hogy mit szeretnének
- Kommunikációs problémák/félreértések merülnek fel
 - vevők
 - fejlesztők
 - menedzsment
- A szoftver működési elemeit „magától értetődőnek” tekintik
- Majdnem mindig elvárás, 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



Példa

- Követelmények fontossága <ház, ceruza>
- Milyen követelményeket támasztanak?



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!





Követelmények forrásai

- Felhasználó(i csoportok)
- Más (hasonló) rendszerek (HW, SW)
- „Általános” tudás, tapasztalat az üzleti környezettel és/vagy a szoftverfejlesztéssel kapcsolatban

- 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. IT infrastruktúra)
 - Örökölt termékek vagy komponensek (újrafelhasznált komponensek)
 - Törvényi szabályozások



Követelmények forrásai

- A követelmények forrására vonatkozóan – ha ők érdekelt felek – legalább a következő információkat szokás feljegyezni:
 - Név
 - Beosztás, szerepkör
 - 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 (glossary), amely tartalmazza a követelményekben megfogalmazott (alap)fogalmakat!



Követelmények felmérése

- 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

- 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 konfliktusmegoldás dokumentálása
 - ➔ a következő alkalomra lesznek tapasztalatok!



Követelmények típusai

- Funkcionális követelmények
- Nem-funkcionális követelmények
 - Minőségi követelmények
 - Korlátok (constraints)
- Követelmény-típusnak tekinthetjük még az alábbiakat is:
 - Üzleti követelmények
 - Felhasználói követelmények
 - Termék követelmények
 - Interfész követelmények



Funkcionális követelmények

- „**Amit** a rendszer csinál”
- **Funkcionális követelmény** (Functional Requirements Specification): olyan követelmény, amely a szoftverrel szemben támasztott funkcionális elvárást írja le.
- Ezt széles / általános hallgatóság számára fejlesztik. Az olvasónak meg kell értenie a rendszert, de semmilyen sajátos vagy technikai tudásra nincs szükség ehhez.
- A funkcionális követelményeknek tartalmazniuk kell:
 - ☐ A rendszer bemenő adatainak leírását
 - ☐ Az egyes képernyők által megvalósított műveletek leírását
 - ☐ A rendszer munkafolyamatainak 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

- „**Ahogy**an a rendszer csinálja”
- Expliciten dokumentálni kell őket. Különös tekintettel az alábbiakra:
 - ☐ Teljesítmény
 - ☐ Biztonság
 - ☐ Megbízhatóság
 - ☐ Használhatóság
 - ☐ Karbantarthatóság
 - ☐ Hordozhatóság
- Részletesebb leírás: ISO/IEC25010:2011

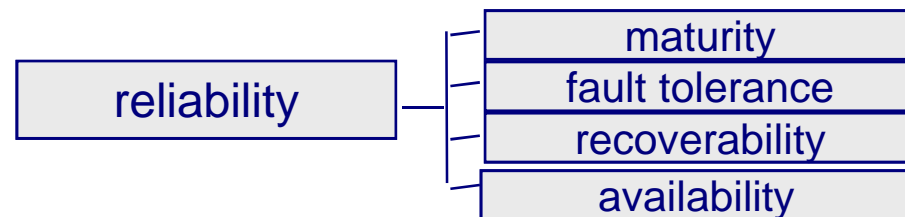


Nem-funkcionális követelmények (az ISO 9126 szerint)

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.

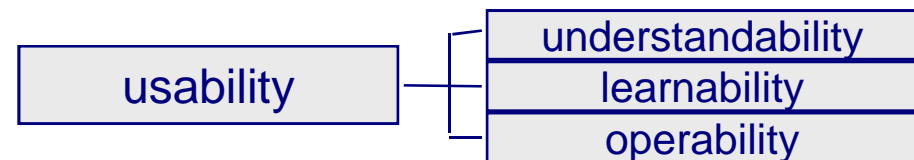
Példa nem-funkcionális követelményekre

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

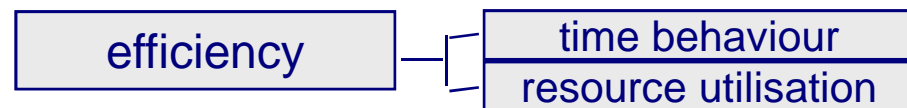


Példa nem-funkcionális követelményekre

- 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)





Korlátok (constraints)

- A korlátozás olyan követelmény, amely leszűkíti a megoldási lehetőségeket azon belül, ami ahhoz szükséges, hogy a rendszer a számára leírt 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 ...



Felhasználói követelmények

- Magas szintűek

- ☐ Kezdetben kicsit bizonytalanok.
- ☐ Segíteni kell a felhasználónak megfogalmazni őket.

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

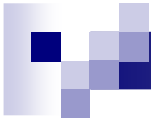
- ☐ Tehát, azt írjuk le, hogy **MIT** akarunk, és nem azt, **HOGYAN fogjuk ezt megoldani!**

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



Felhasználói követelmények

- 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 e-mail-ben megkapja a rendelése visszaigazolását



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.



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

- A modell a valóság leképezése
- Három szempontja:
 - - valóságból indulunk ki (representation)
 - - valóságot leképezzük/szűkítjük (reduction)
 - - egy bizonyos szempont dominál (pragmatic)
- *Példa: BP autós vs metró térkép*



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 három nézőpontból végezhetünk modellezést:
 - ☐ Adatok
 - ☐ Funkcionalitás
 - ☐ Viselkedés
- 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

- A követelményekben megjelenő hiba igen veszélyes, de sajnos gyakori! Ne sajnáljuk az időt a **követelmények ellenőrzésére!**
- Jó megközelítés, ha ellenőrizzük, hogy a különböző, követelményeket leíró modellek elemei konzisztensek-e!
 - Eszközök segíthetnek!



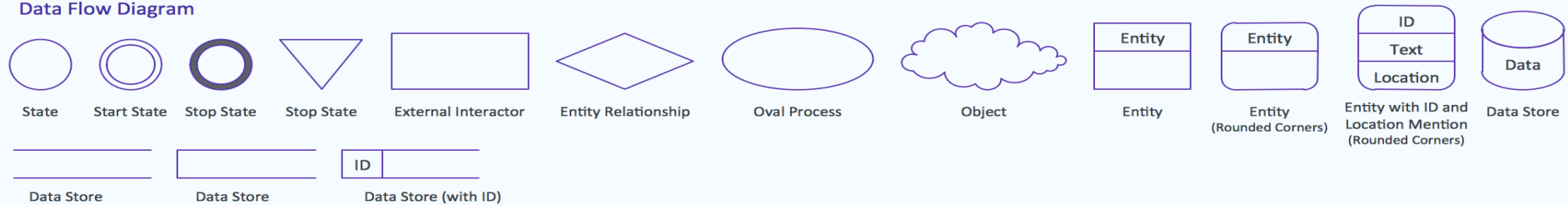
Követelmények modellezése

■ Felhasználható alapelemek:

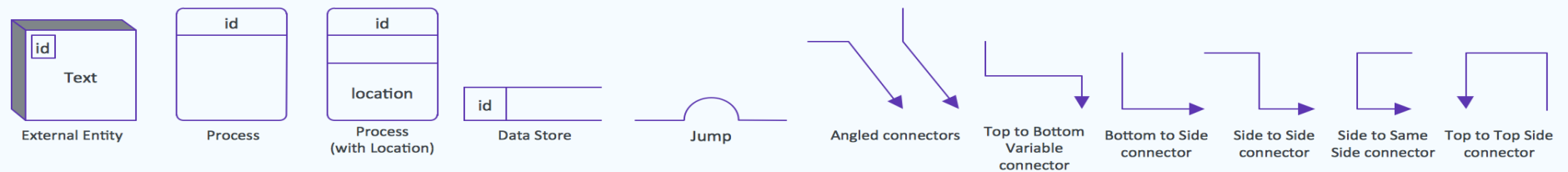
- ☐ Adatok
 - ☐ Akciók
 - ☐ Eszközök
 - ☐ Események
 - ☐ Vezérfonalak
- (Jorgenssen alapján)



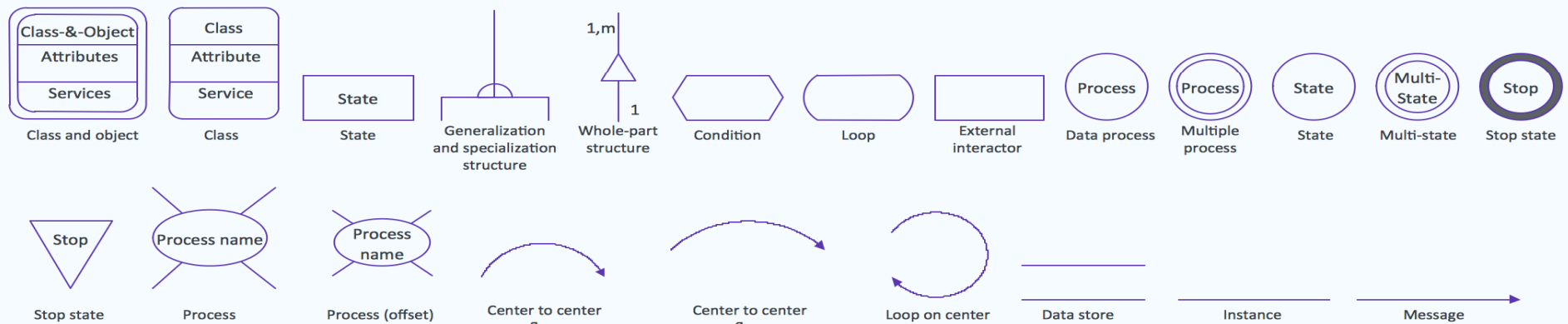
Data Flow Diagram



Gane-Sarson notation



Yourdon and Coad notation



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



Követelmények priorizálása

- Fontos: kihat az implementáció sorrendjére
- Előfordulhat egyet nem értés
 - ☐ Kezelni kell
 - ☐ Kell egy rendezési elv
- Prioritások változhatnak
 - ☐ Piaci körülmények miatt
 - ☐ Megrendelői igények változása miatt



Prioritization Methods

- Tradicionális
 - Kötelező, kívánatos, opcionális
- Kano
 - Alapvető, elvárt, lelkesedést kiváltó faktorok
- Rangsorolás és top 10 technika
 - Egy kritérium alapján a top 10 (utána másik kritérium)
- Nyelvi kifejezőeszközön keresztül
 - MoSCoW: must have, should have, could have, won't have



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

- 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
 - Verziókezelt (követhető a fejlődése, változtatások)
 - 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



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

Követelmények agilis környezetben

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

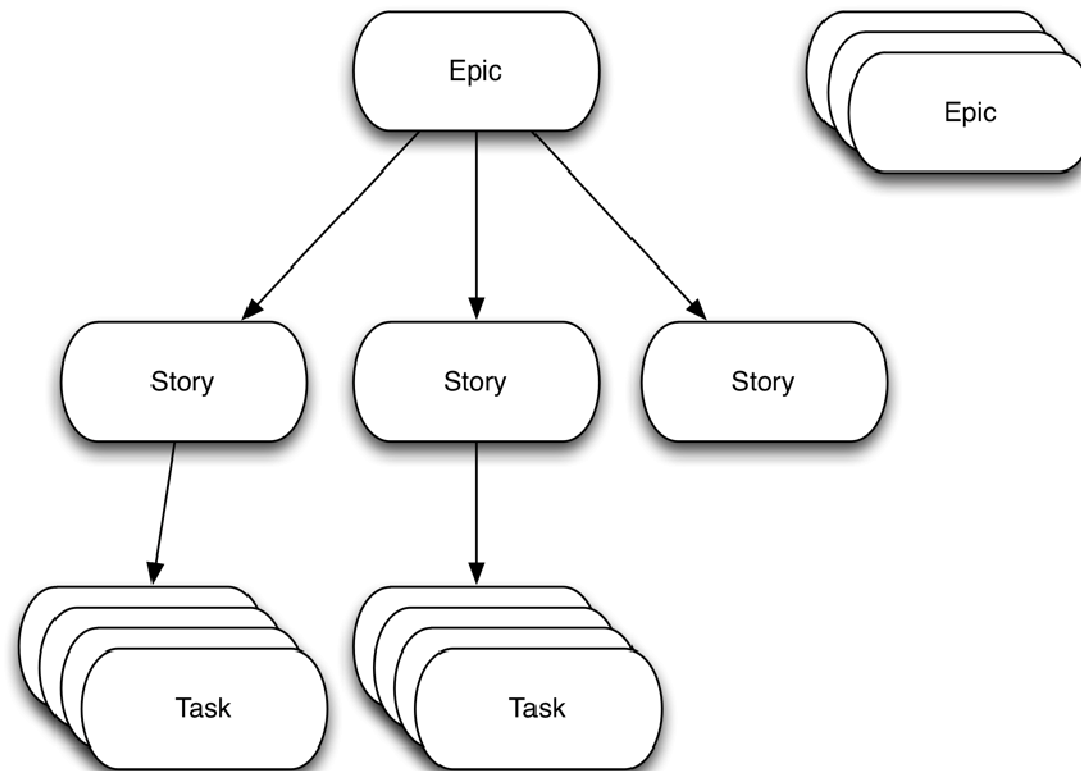
- Became smaller.
- Became less precise.
- Less time for elaborating them.
- Easy to change or extend.
- Now they encourage discussions instead of describing the final state.



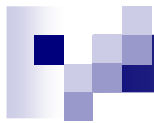
<https://www.slideshare.net/E5Trainings/requirements-in-agile-conference>



Követelmények agilis környezetben



<https://www.agileconnection.com/article/stories-epics-and-tasks-organizing-agile-requirements>



Követelmények agilis környezetben

- Epic: egy funkció/termék/szolgáltatás egy olyan része, amely annyira összetett és komplex, hogy részletekre kell bontani
- User story (felhasználói történet): jellemzően egy, vagy több, hétköznapi, vagy üzleti nyelven megfogalmazott mondatot tartalmaz, amely leírja, hogy a felhasználónak milyen funkcionalitásra van szüksége, vagy bármilyen egyéb nem-funkcionális követelményt fogalmaz meg, továbbá tartalmazza az **átvételi kritériumot** is.



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?

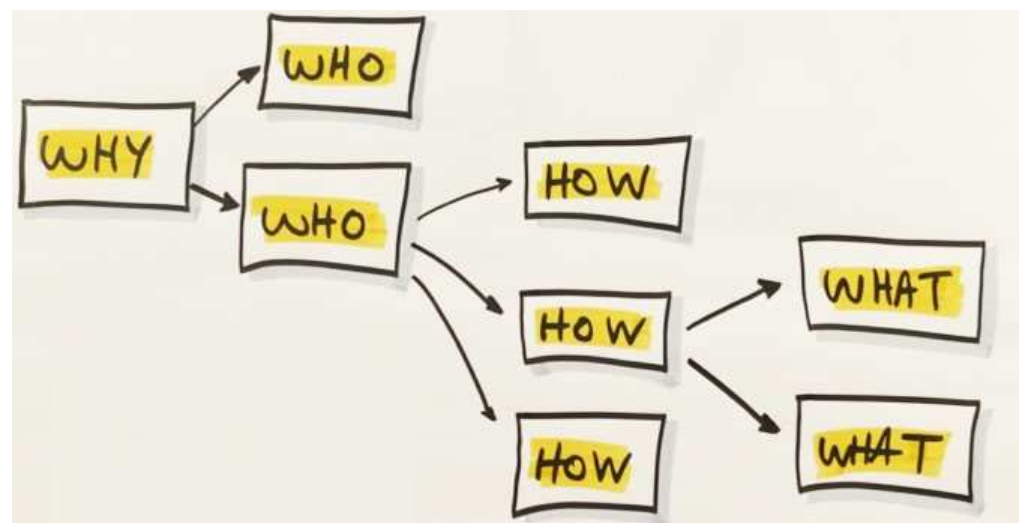
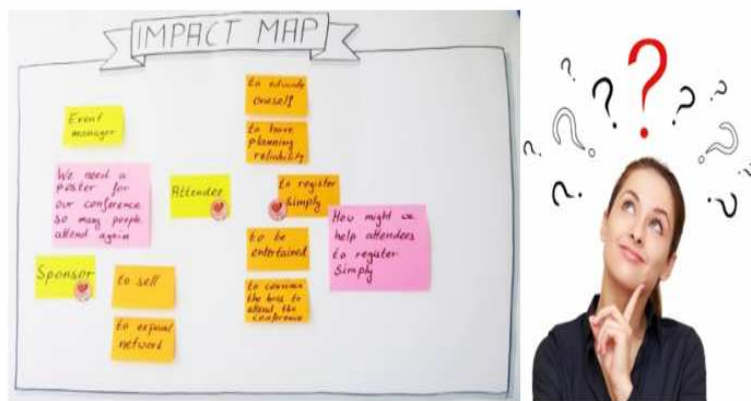


Követelmények agilis környezetben

■ 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 egyeztetése agilis környezetben



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

Követelmények priorizálása agilis környezetben

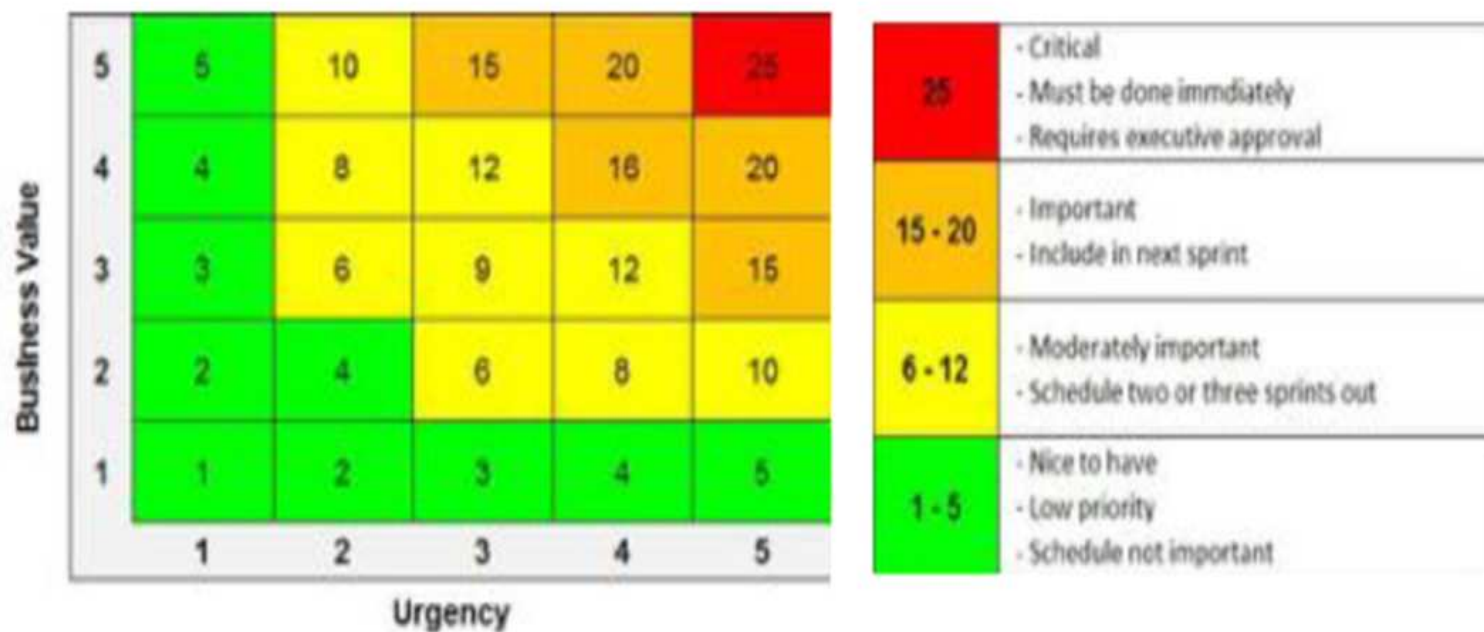
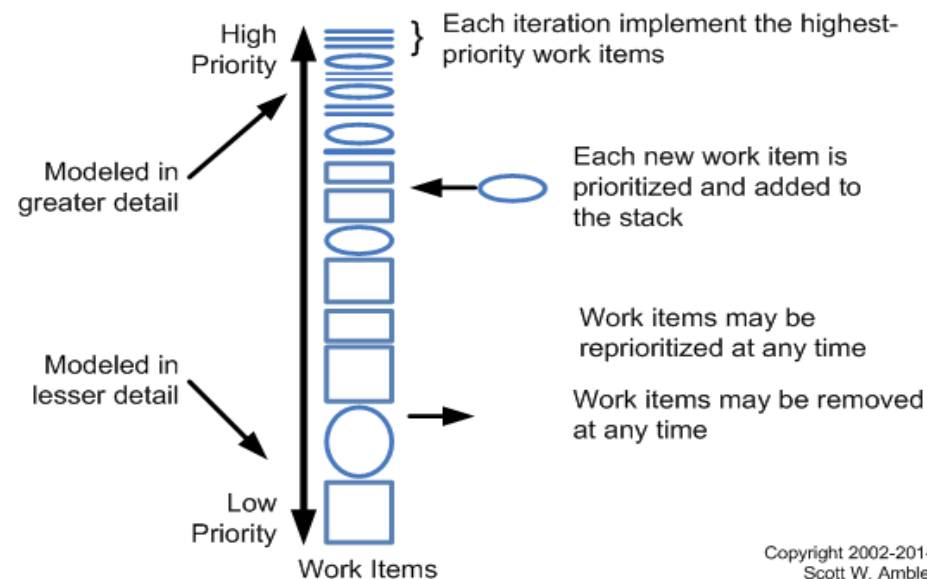


Figure 5-9. Putting the user stories in priority

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!



Miről volt szó...

- A követelmények definíciója és fontossága
- Követelmények fejlesztése
- Követelmények felmérése, egyeztetése
- Követelmények típusai
- Követelmények dokumentálása, modellezése
- Követelmények kezelése, karbantartása
- Példák különböző követelményekre
- Követelmények agilis környezetben

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/#cpre-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_cpre_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

