

KORSZERŰ RENDSZERFEJLESZTÉSI TECHNOLÓGIÁK

VARJASI NORBERT

VARJASIN@SZE.HU

MODERN SOFTWARE DEVELOPMENT METHODOLOGIES



TÉMAKÖRÖK

- Történeti áttekintés
- Fejlesztési folyamatok szervezése, egyéni és csapatmunka, csoportdinamika (hagyományos, agilis)
- Források kezelése, verziókövetés (GIT, subversion) Fejlesztési folyamat, verziók követése és konfliktusok feloldása
- Projektet támogató eszközök, és automatizálás (Jira, Jenkins, TFS)
- Kód újratervezés, refactoring, migráció
- Tervezési minták használata, gyakorlati esetek. Tesztelés, tesztelési szintek (unit teszt, elfogadási teszt, stb)
- Éles rendszerek kezelése, és üzemeltetése. (adatmentés, helyreállítás, adatbiztonság)

TÉMAKÖRÖK

- Történeti áttekintés
- Fejlesztési folyamatok szervezése, egyéni és csapatmunka, csoportdinamika (hagyományos, agilis)
- Források kezelése, verziókövetés (GIT, subversion) Fejlesztési folyamat, verziók követése és konfliktusok feloldása
- Projektet támogató eszközök, és automatizálás (Jira, Jenkins, TFS)
- Kód újratervezés, refactoring, migráció
- Tervezési minták használata, gyakorlati esetek. Tesztelés, tesztelési szintek (unit teszt, elfogadási teszt, stb)
- Éles rendszerek kezelése, és üzemeltetése. (adatmentés, helyreállítás, adatbiztonság)



How the customer explained it



How the Project Leader understood it



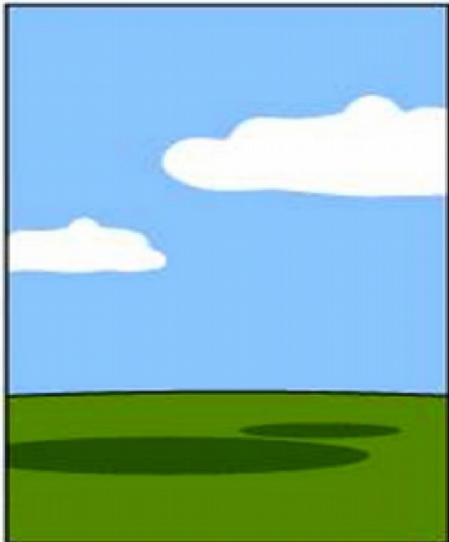
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



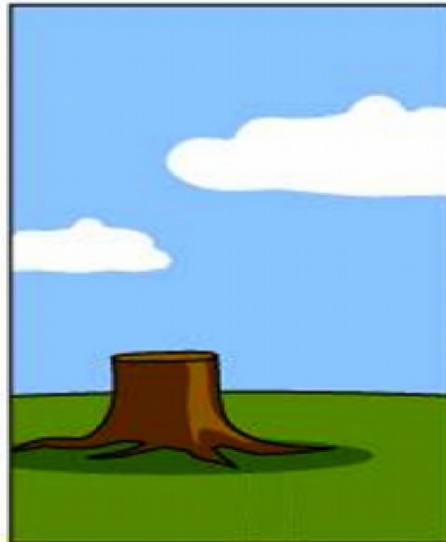
How the project was documented



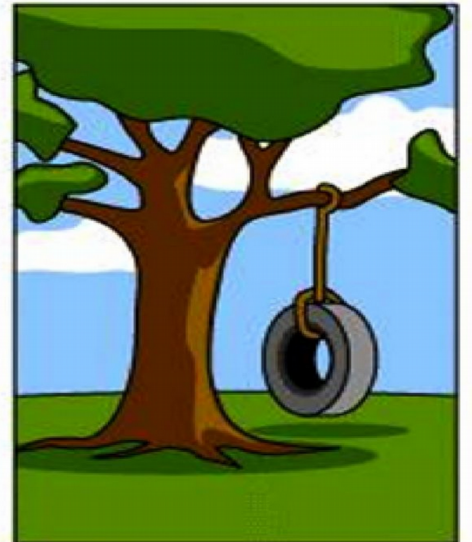
What operations installed



How the customer was billed



How it was supported

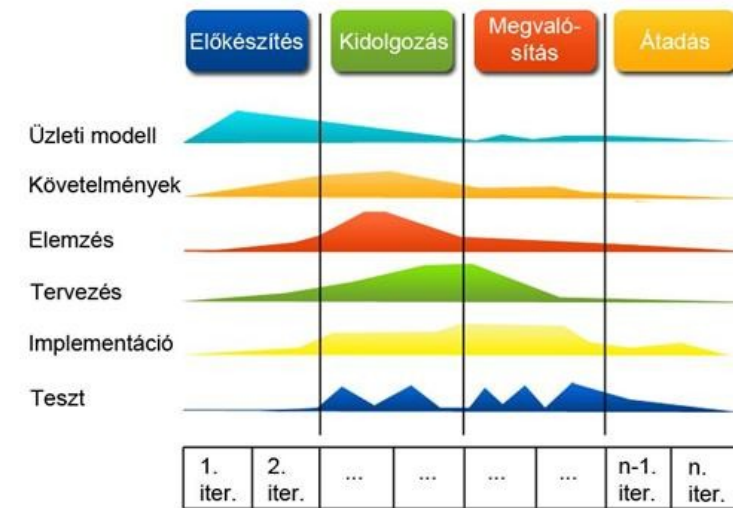
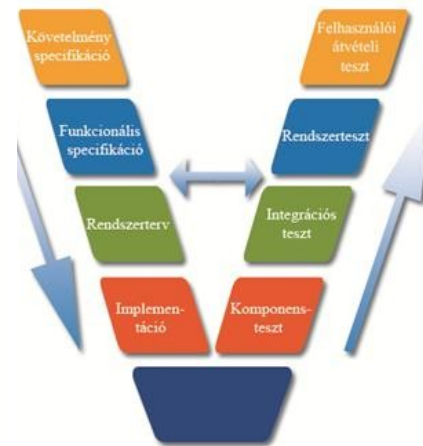
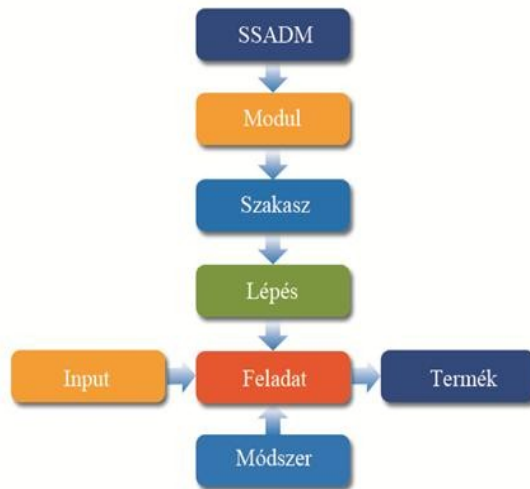
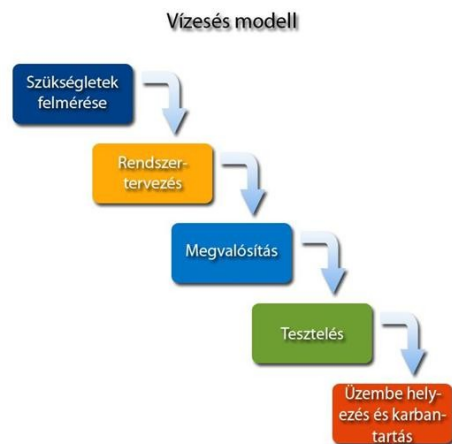


What the customer really needed

TÖRTÉNET

- Projektmenedzsment módszerek
- Problémák és eltérések a klasszikus mérnöki szemlélettől
 - (azaz miben különböznek az IT projektek?)
- Megoldások (évtizedek):
 - Vízésés (1960-70)
 - Iteratív modell (80) – spirál, RUP
 - Evolúciós modell (80-as évek második fele)
 - V modell (90)
 - Újrafelhasználható modellek (90)
 - Agilis (2000)
 - Scrum, Lean, Kanban,

HAGYOMÁNYOS MÓDSZERTANOK



FEJLESZÉSI FOLYAMATOK SZERVEZÉSE

- Fejlesztési folyamat lehetséges céljai:
 - Értékteremtés (pl eladható szoftver),
 - Ügyfél igények teljesítése (ami neki értékes azért fizet)
 - Projekt teljesítése (pályázat, szakdolgozat :)
 - Egyéb (hobbiból)
- Szereplők: *Megrendelő, Szakértő, Tervező, Fejlesztő, Tesztelő, Felhasználó,...*
- De mi ezzel a probléma?

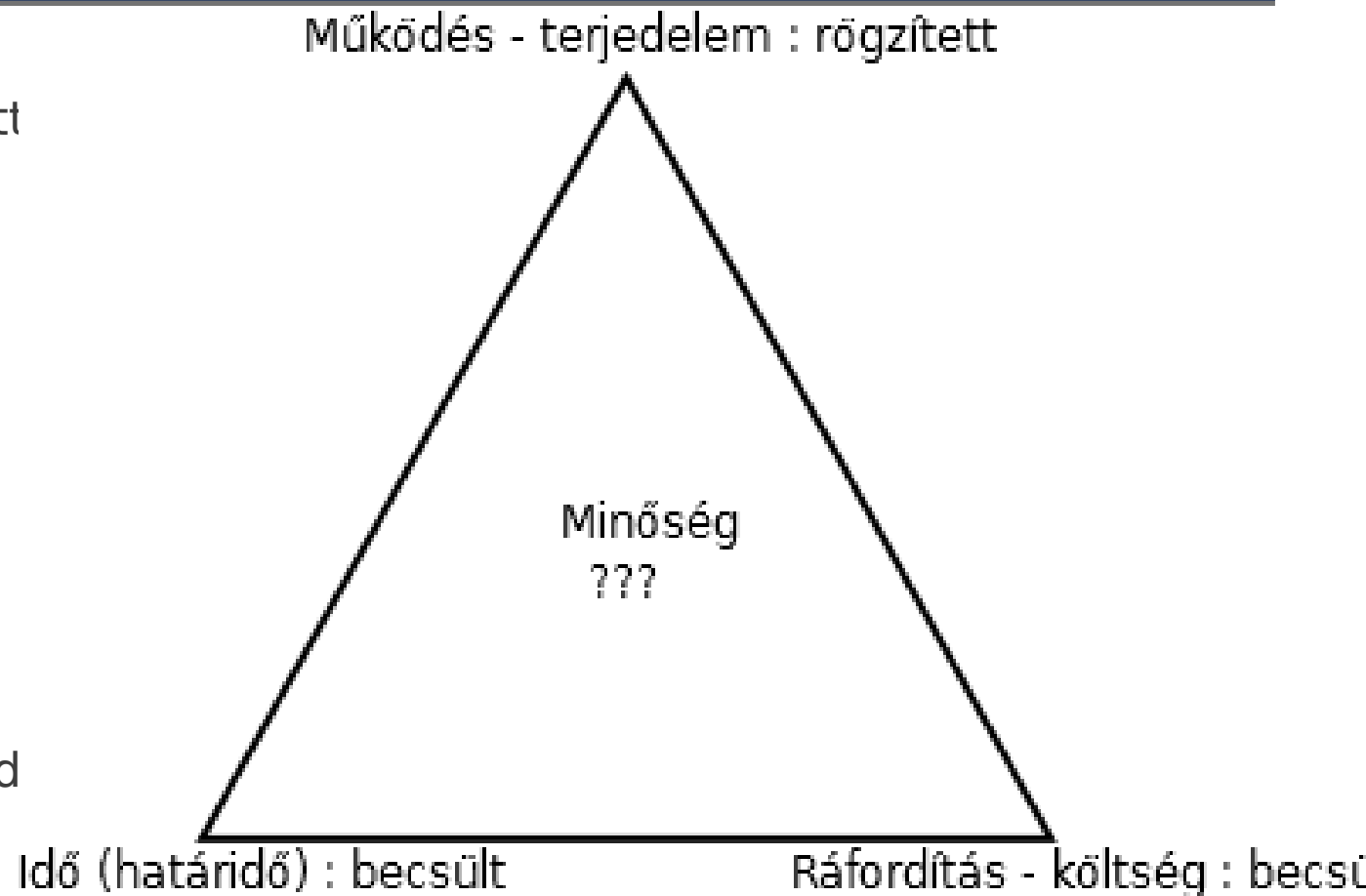
KLASSZIKUS PROJEKTEK

- Koncepció
- Tervezés, Szervezés
- Végrehajtás
- Lezárás

(Ötlet, megvalósíthatósági tanulmány, idő-erőforrás-költségterv, érdekcsoporthok, életképesség, cél-funkció, hatások, indikátorok, marketing,... fejlesztések, határidők)

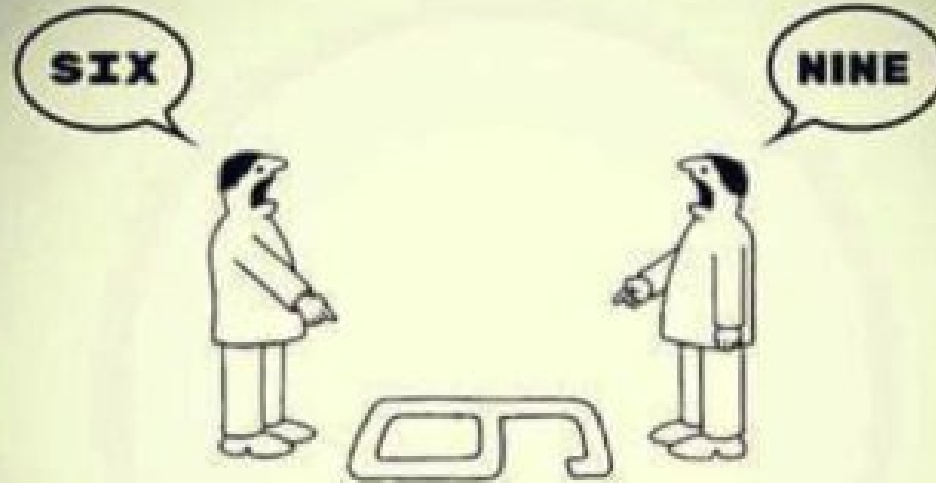
AZ ALAPPROBLÉMA

- Egyszerre minden alapfeltétel nem adott
- Hibás fejlesztések főbb jellemzői:
 - ismétlődő, hibák
 - quick and dirty megoldások
 - beégetett kódok (semmi rugalmasság)
 - nincs dokumentáció,
 - nincs tervezés,
 - nincs tesztelés,
 - lelassuló projektek, egyre lassabb haladás
 - rossz kommunikáció,
 - folyamatos határidő stressz, rossz menedzsment



SZOFTVERKRÍZIS

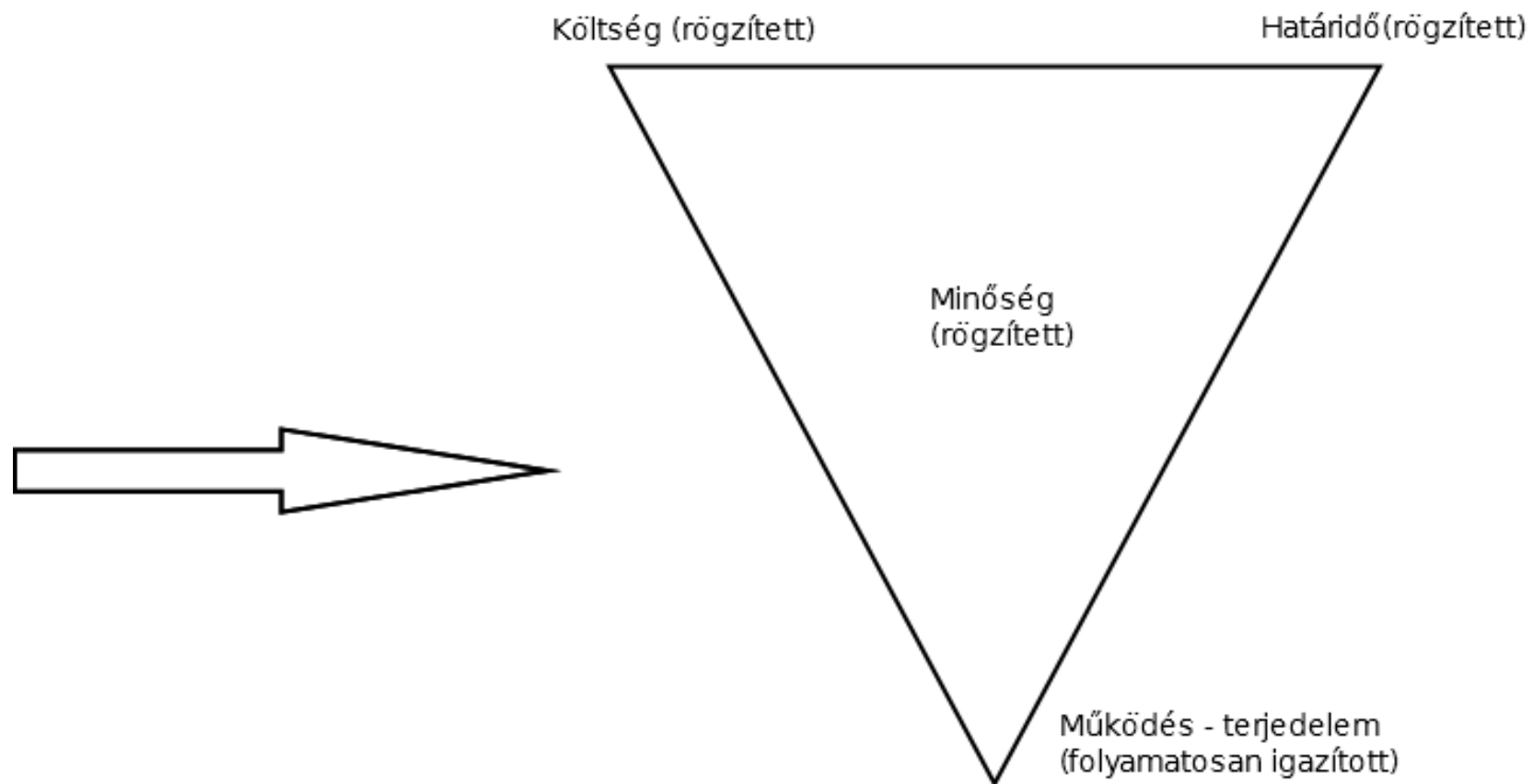
- Kifut az időből
- Kifut a költségkeretből, gyenge minőség,
- A szoftver nem hatékony (nem azt csinálja, amit rendeltek)
- Karbantarthatóság
- Bővíthetőség



Just because you are right,
does not mean, I am wrong.
You just haven't seen life
from my side.

Something to think about....

EGY HATÉKONYABB ALAPÁLLÁS



AGILIS FEJLESZTÉS

- AGILIS = alkalmazkodó, hatékony, mozgékony, életrevaló
- A csapat
- A tervezés
- A kivitelezés

Ken Schwaber, Jeff Sutherland 1995



SCRUM



CÉLOK

1. Felhasználók és a termék
partnereknek szólnak a fejlesztések, azaz az egyén és a kölcsönhatás fontos
2. Minőség
naprakészség, gyors reagálás, problémamegoldás, működőképes szoftver
3. Naprakészség
technológia, kihívások, igények
4. Folyamatosan javuló minőség
hibák csökkenése, visszajelzések, tanulás a hibákból

Product Owner



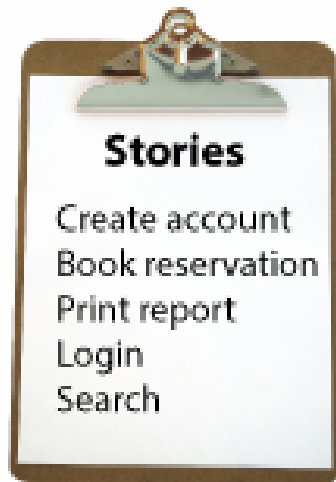
Scrum Master



Delivery Team



Product Backlog



Sprint Backlog



24 hrs

Daily Scrum

2 weeks

Sprint

Potentially Shippable
Product Increment



TERMÉKTULAJDONOS – PRODUCT OWNER

- Felelős a termék értékének maximalizálásáért és a Fejlesztőcsapat munkájáért
- Termék Teendőlista - Product Backlog kezelése
 - Tételek definiálása – a célok felbontása
 - Tételek egyértelmű leírása, priorizálása
 - Mérföldkövek, célok
 - Kapcsolattartás a megrendelővel

SCRUM MASTER

- Célja az agilis / scrum elvek alkalmazása és betartása a csapatban
- Módszerek a backlog kezeléshez
- Scrum eseményeket szervezi
- Terméktulajdonossal egyeztet, információt cserél
- Fejlesztőcsapatot támogatja, akadályokat elhárítja
- Növeli a csapat termelékenységét

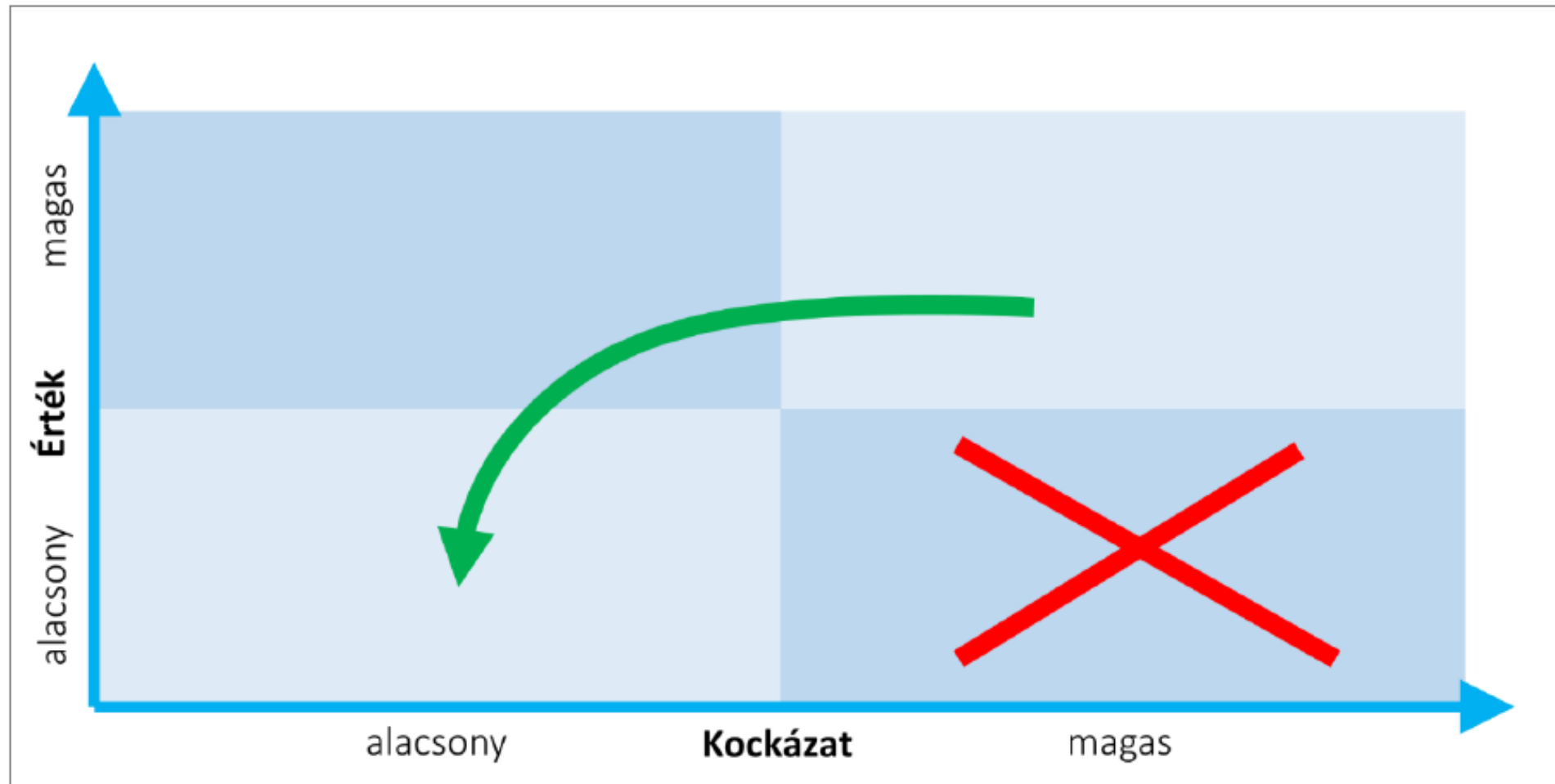
FEJLESZTŐCSAPAT

- **Szakemberek** vagy **fejlesztők** – (nem csak programozók!!!)
- Önszerveződő, önmenedzselő
- Ők határozzák meg, hogy hogyan halad egy fejlesztés
- Mindenki fejlesztőként vesz részt a folyamatban, nincsenek külön szerepek
- A tagok ismerete és tudása különböző (programozó, grafikus, közgazdász, gépész/eü/környezetmérnök..)

USER STORY

- **Felhasználóként [szerep] a következőket [a cél] szeretném [okok] ...**
- PL:
 - Felhasználóként – ha rendelkezik „archiválás” jogosultsággal, a Mentés gombra kattintva a tartalmat pdf formában menti.
 - Adminisztrátorként jogokat adok, vagy elveszek a felhasználóktól.
- Akkor „jó” ha megfogalmazható hozzá az elfogadási kritérium (AC)
- „Kész” fogalma (Definition of Done) !

FELADATOK ÉRTÉK / KOCKÁZAT ALAPÚ FELBONTÁSA



TEVÉKENYSÉGEK, FOGALMAK

- backlog (nyílt feladatlista)
- sprint (egy két hetes fejlesztési ciklus),
- sprint tervezés, fejlesztési lista összeállítás,
- sprint lezárása, (bemutató);
- napi rövid egyeztetés (daily scrum / stand-up)
- retrospektív megbeszélés,

Input from End-Users,
Customers, Team and
Other Stakeholders



Product Owner

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Product
Backlog



Team

Team Selects
How Much To
Commit To Do
By Sprint's End

Sprint Planning
Meeting
(Parts One and Two)



Sprint
Backlog



ScrumMaster

Product
Backlog
Refinement

Sprint

1-4 Weeks



Daily Scrum
Meeting and
Artifacts Update



Review



Potentially
Shippable Product
Increment



Retrospective

No Changes
in Duration or Goal

Világos célkitűzések

Egyes scrum guide-ok szerint az agilis fejlesztések itt buknak el leghamarabb:

„a cél az, hogy az összes backlog funkciót elkészítsük”

„azt csináljuk amit eddig...”

„szeptemberben valamit át kell adnunk...”

-> Nem ellenőrizhető, nehezen értelmezhető, nem váltható aprópénzre!

Mit tekintünk egy release céljának?

Mit tekintünk egy sprint céljának?

User story alapú megközelítés, megfogalmazás:

- a felhasználói tevékenységek szemszögéből, feladat leírás
- ki , mit és milyen célból vár el az alkalmazás adott képességétől

Kész fogalma (elfogadási kritérium – AC)

- Mikor van készen egy feladat?
 - 1) ha tényleg KÉSZEN van!
 - 2) ha a leírt feladat (user story) végigvihető hiba nélkül a rendszerben
 - 3) ha az a feladat azonnal átadható a megrendelőnek
- PO szempontjából: ha a feladat maradéktalanul teljesíti az AC-ban leírtakat
- Fejlesztők szempontjából:
 - 1)ha a kért feladat minden részfeladatot megvalósít,
 - 2)nem generál hibákat másutt,
 - 3)inputra ellenőrzött (minden tesztet teljesít),
 - 4)a verziókezelőbe feltöltött (git) és a tesztrendszeren is működik
 - 5)a feladatot megvalósítja az AC szerint.

PÉLDA 1

- Főzés (user story – recept, kész fogalma, elfogadási tesztek)
 - Zsíroskenyér
 - Rántotta
 - Pörkölt
 - Pizza
 - Medvehagymás fűszeres pizstráng
 - Wellington bélszín

PÉLDA 2

- Lakás berendezése
 - 50 m²,
 - Helyiségek száma és szerepe
 - Kialakítás költségei
 - Idő

