



Szoftvertesztelés

SCRUM keretrendszer

Történeti háttér

- 1986-ban publikálták elsőnek a Scrum kifejezést igaz, akkor ezt még kifejezetten a **kereskedelmi termékfejlesztésre** alkalmazták.
- Maga a scrum elnevezés eredetileg rögbis szakzsargon, és a labda körül fellépő **tolongást** jelöli.
- A csapattagok együtt vannak a kezdéskor, majd mindenki egyszerre indul el, és **mindenki pontosan azt csinálja, amit csinálnia kell**, persze a csapatban előre leosztott szerepe alapján.
- A csapat egy egységként halad a cél felé, de közben azért **nem mindig ugyanannál a személynél van a labda**, hiszen megy a passzolgatás annak megfelelően, ki van épp a játék egészének szempontjából a legmegfelelőbb pozícióban.

Történeti háttér

- Tehát ebben a játékban, ahogyan az agilis scrum keretrendszerben nem az egyéni teljesítmény a hangsúlyos, hanem a **csapatmunka**, a **világos célok**, a **bizalom**, valamint a szerepek közötti rugalmas munkavégzés, na meg a közösen vállalt szabályok fegyelmezett betartása.
- A vállalaton belüli **együttműködések minősége** egyáltalán nem mindegy, milyen. Gyakran tapasztalható, hogy a funkcionálisan szétagolt területek között **kommunikációs félreértések**, eltérő irányú célkitűzések, egyfajta bizalomhiány, vagy éppen az egymás munkájának nem megfelelő ismerethiányból fakadóan jelentős fennakadások, ebből kifolyólag pedig **anyagilagban mérhető károk** jelentkeznek a szervezetben.

Történeti háttér

- A csapatok fejlesztésének legfőbb célja, hogy a társterületek **megértsék, és pontos ismeretekkel rendelkezzenek** egymás működési folyamatairól, céljairól, nehézségeiről.
- Emiatt nélkülözhetetlenek a folyamatfejlesztések, és a munkatársak, tehát a humán tényező **pontos megismerése** a vezetők által, amelynek köszönhetően minden esetben **közelebb kerülhet egymáshoz** a szervezet minden részlege, speciális szakterülete.
- Ennek hatására pedig elősegíthető a **belső stabilitás**, és állandóság megteremtése.

Mit is jelent a SCRUM?

- Nagyon egyszerűen és tömören megfogalmazva, a scrum egy **agilis keretrendszer**, amely biztosítja a **komplex feladatok összetett kivitelezését**, a folyamatok pontos és behatárolható végrehajtását.



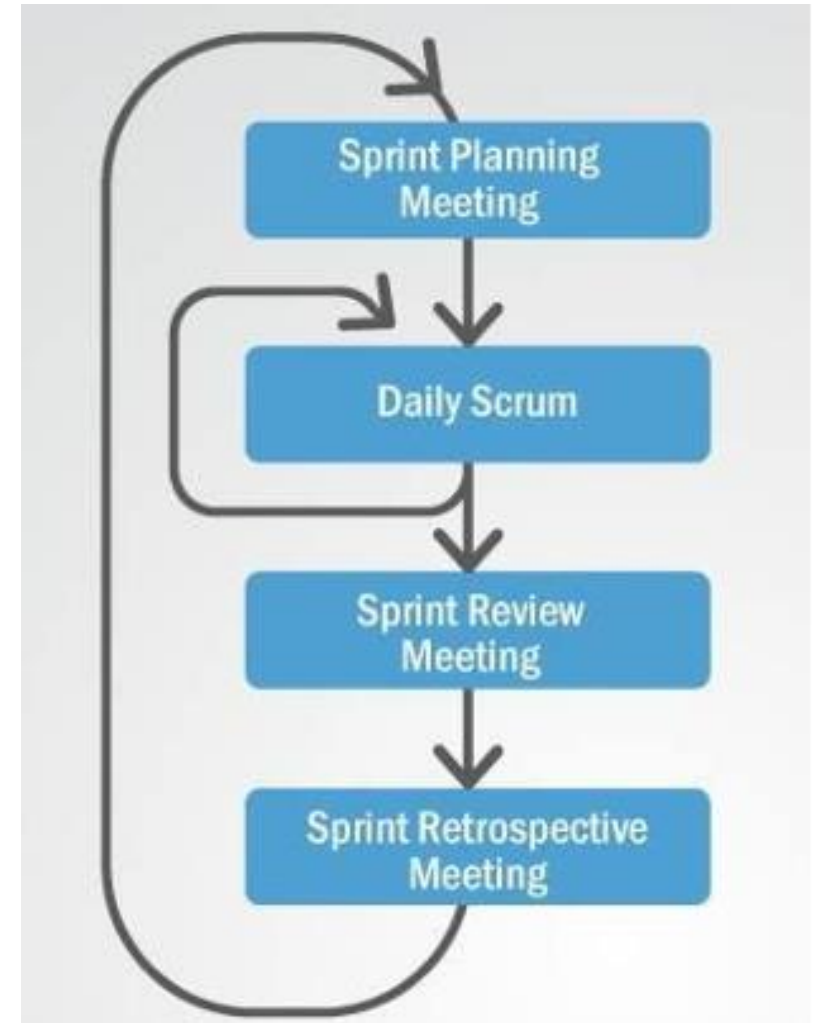
*„...NOHA A SCRUMOT EREDETILEG
SZOFTVERFEJLESZTÉSI PROJEKTEKHEZ
DOLGOZTÁK KI, DE BÁRMILYEN ÖSSZETETT,
INNOVATÍV TEVÉKENYSÉG ESETÉN JÓL
MŰKÖDIK. A LEHETŐSÉGEK VÉGTELENEK. A
SCRUM KERETE BECSAPÓAN EGYSZERŰ.”*

SCRUM keretrendszer



Scrum események

- **Sprint** (Futam)
- **Sprint planning** (Futam tervezés)
- **Daily Stand-up** (Napi SCRUM)
- **Sprint review** (Bemutató)
- **Sprint retrospective** (visszatekintés)



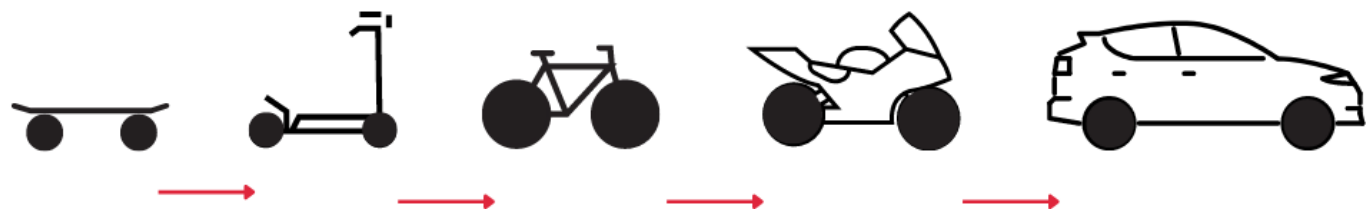
Futam

- Tulajdonképpen ez maga a scrum szíve-lelke, központi **fogaskereke**.
- A sprintek alatt lesz az **ötletekből érték**, amelyet értékesíteni lehet majd a jövőben, de minimum profitálni lehet belőle már a közeli jövőben.
- Időtartama **maximum 1 hónap**, de a lényeg, hogy mindenképp legyen egyfajta ritmusa a folyamatnak, azok időintervalluma pedig legyen rögzítve.
- Ezáltal biztosítva lesz a csapat munkájának, valamint az ügyféllel való együttműködésnek **az üteme** – amely kulcsfontosságú a scrum elsajátításának szempontjából.

Futam

- Tulajdonképpen ez maga a scrum szíve-lelke, központi **fogaskereke**.
- A sprintek alatt lesz az **ötletekből érték**, amelyet értékesíteni lehet majd a jövőben, de minimum profitálni lehet belőle már a közeli jövőben.
- Időtartama **maximum 1 hónap**, de a lényeg, hogy mindenképp legyen egyfajta ritmusa a folyamatnak, azok időintervalluma pedig legyen rögzítve.
- Ezáltal biztosítva lesz a csapat munkájának, valamint az ügyféllel való együttműködésnek **az üteme** – amely kulcsfontosságú a scrum elsajátításának szempontjából.

Sprint

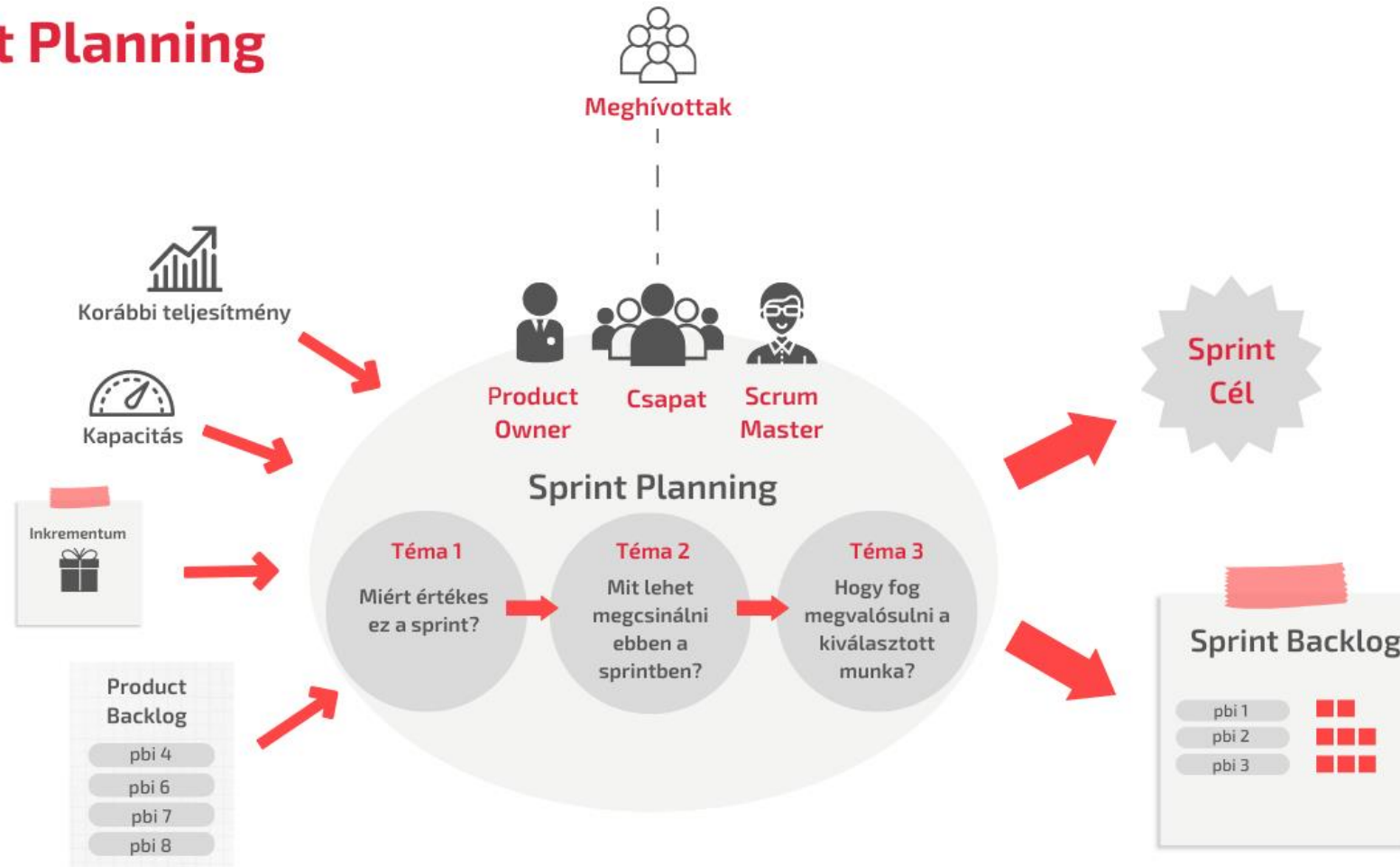


Futam tervezés (magas szintű tervezés)

- A csapat találkozik a termékgazdával, hogy megtervezzék a fentebb említett **sprint részleteit**, kivitelezését.
- A csapat saját gyorsaságát és teljesítőképességét valójában a **közelmúltbeli** események adta **eredményeire** alapozza, azok alapján számolja ki, jósolja meg kezdetben, ha úgy tetszik.
- A későbbiekben pedig már az **előző sprintek teljesítménye** fogja meghatározni, hogy hány feladatot képesek elvégezni a következő sprintek során.
- A csapat egyébként a sprint alatt **önállóan dolgozik** annak érdekében, hogy a **célt elérjék**, de közben végig szem előtt tartják, hogy sprintenként konkrét leszállítandók készüljenek el, amik az ügyfélnek átadhatók.

Futam tervezés

Sprint Planning



Becslésekhez - Planning Poker

- A csoport tagjai úgy készítenek becsléseket, hogy **a számozott kártyákat képpel lefelé az asztalra dobják**, ahelyett, hogy hangosan kimondanák azokat.
- A kártyákat felfedik, majd **megvitatják a becsléseket**
- Egy tipikus pakliban a **Fibonacci-sorozat** tartalmazó számozott kártyák találhatók, beleértve a nullát is: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89; más paklik hasonló sorrendet használnak, az egyes értékek között meghatározott arányban.
- A Fibonacci-sorozat használata azért indokolt ahelyett, hogy egyszerűen **megdupláznánk** minden egyes egymást követő értéket, mert egy feladatot pontosan kétszer akkora erőfeszítésre becsülni, mint egy másik feladatot, félrevezető pontosságú.

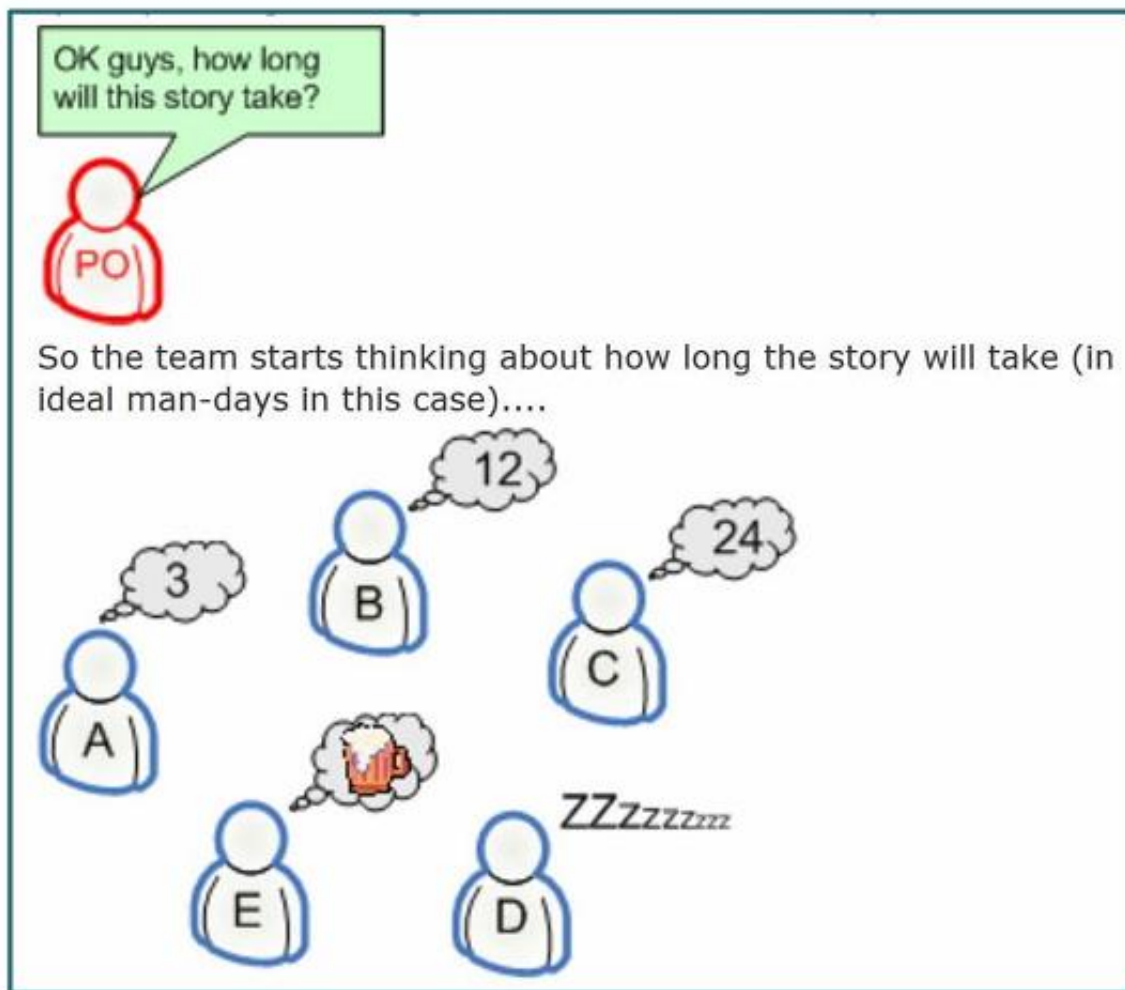
Planning Poker



Planning Poker

- 0 Triviális (pl. egyszerű átnevezés)
- 0,5 - 1 Technikailag nem komplex, egyszerű implementálni
- 2 Technikailag nem komplex, nehezebb implementálni
- 3 Valamelyest komplex vagy alapos átgondolást igényel az implementálás
- 5 Valamelyest komplex, ismeretlenekkel vagy külső függőséggel; kiterjedt tesztelést igényel
- 8 Komplex vagy szövevényes (a rendszer különböző részeit érinti); nagymértékű külső függelmek; különböző ismeretlenek; többszintű tesztelést igényel
- 13 Nagyon komplex és szövevényes (az egész rendszert érinti); rengeteg különböző mértékű függelem; határozottan sok ismeretlen; kiterjedt tesztelést igényel.
- 20 Eposz (Herkules)
- 40 Óriás eposz (Illiász és Odüsszeusz kombinálva)
- 100 Az összes Görög-római, Babiloni, Perzsa és Hetti hősi eposz kombinálva

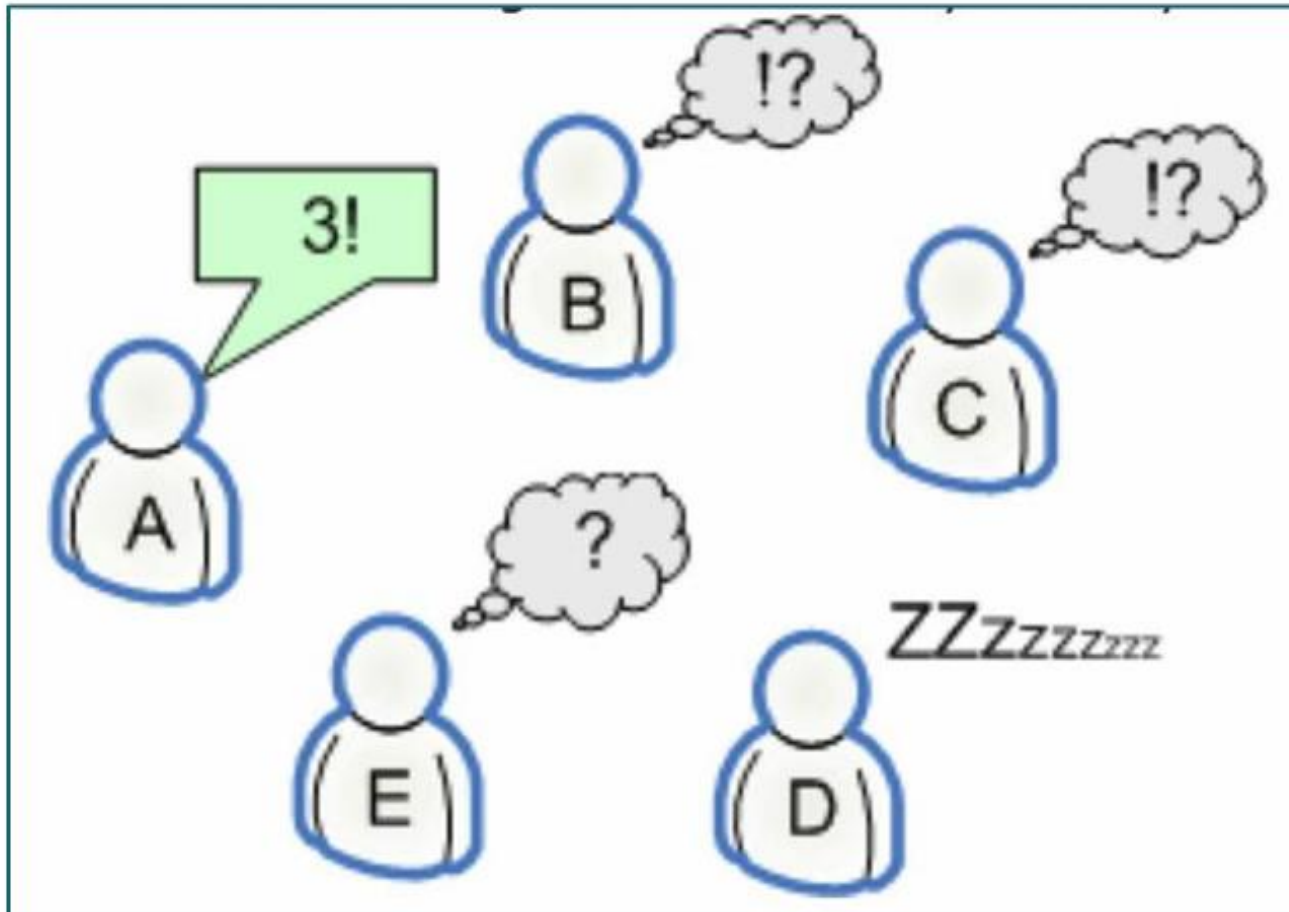
Becslés Planning Poker nélkül



A azt hiszi, hogy határozottan tudja, hogy mit kell csinálni, így azt gondolja, hogy 3 nap alatt meg tudja csinálni.

B és **C** sokkal pesszimistább. **D** és **E** gondolatban másutt jár.

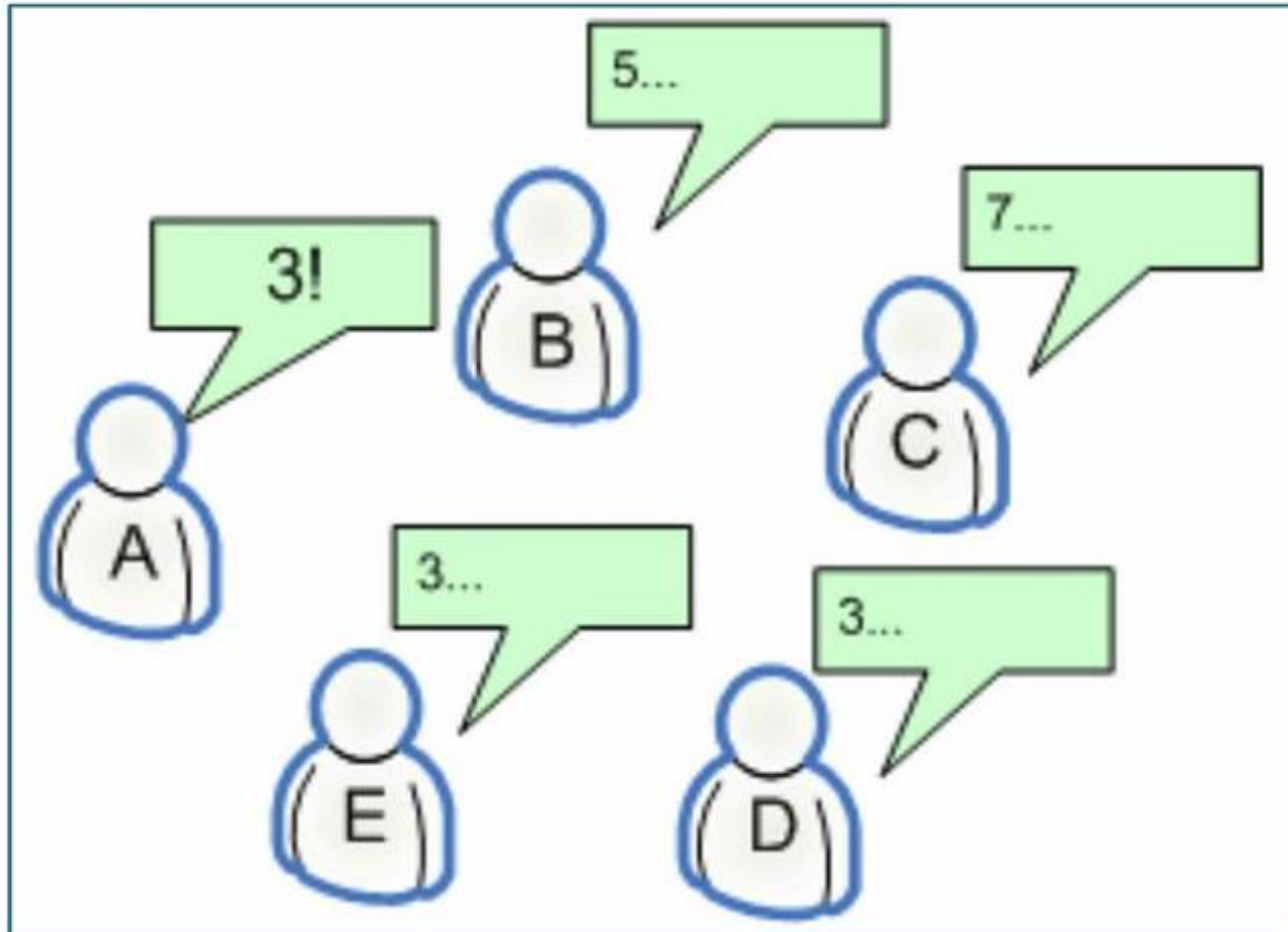
Becslés Planning Poker nélkül



Ez **B**-t és **C**-t
elbizonytalanítja,

E felébred, **D** még
továbbra is alszik.

Becslés Planning Poker nélkül



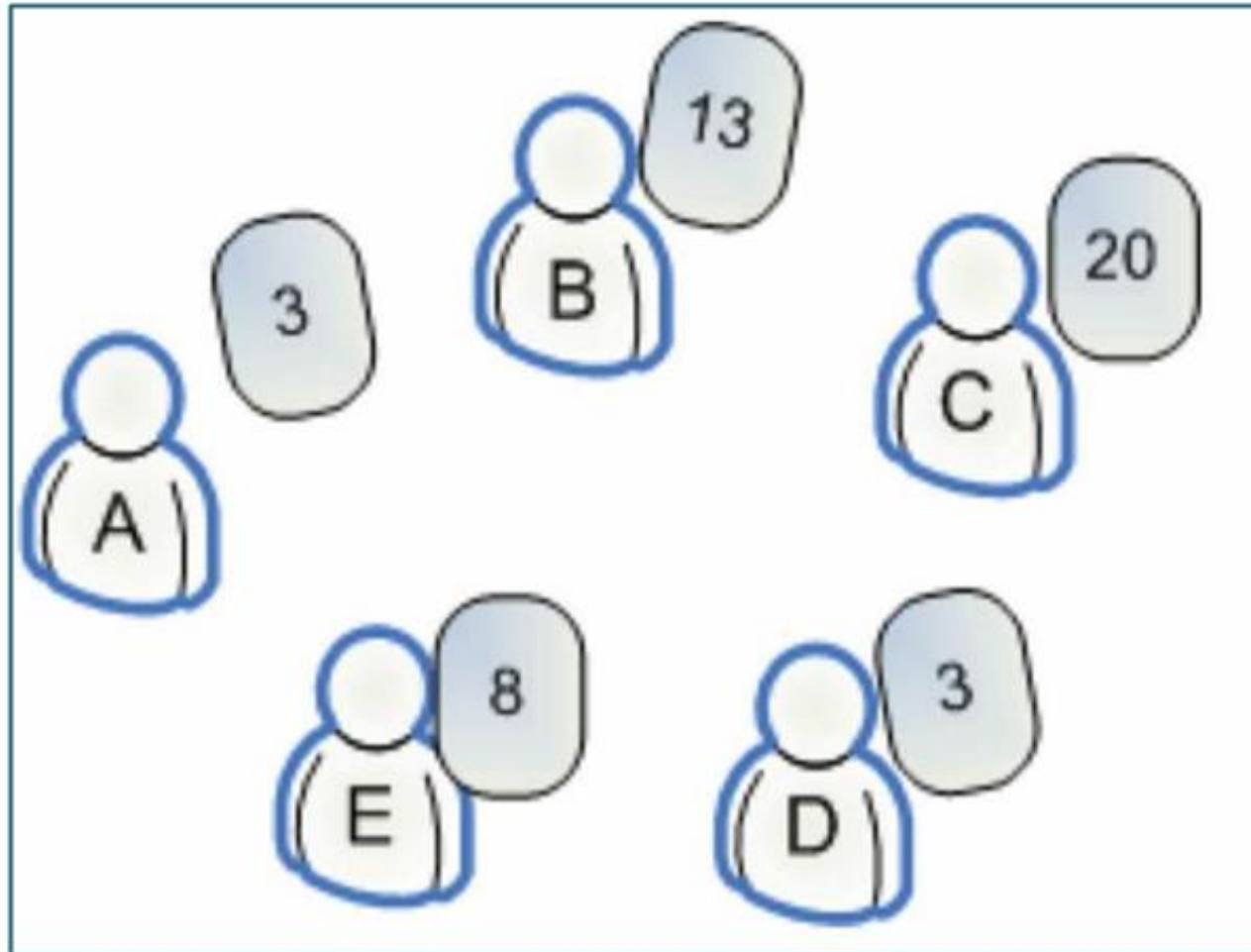
Az újra feltett kérdésre az **A** határozottsága érvényesült, bár **B** és **C** kezdetben sokkal többre gondolt.

Becslés Planning Pokerrel



Most senki nem alszik,
mert muszáj egy kártyát
felemelni.

Becslés Planning Pokerrel

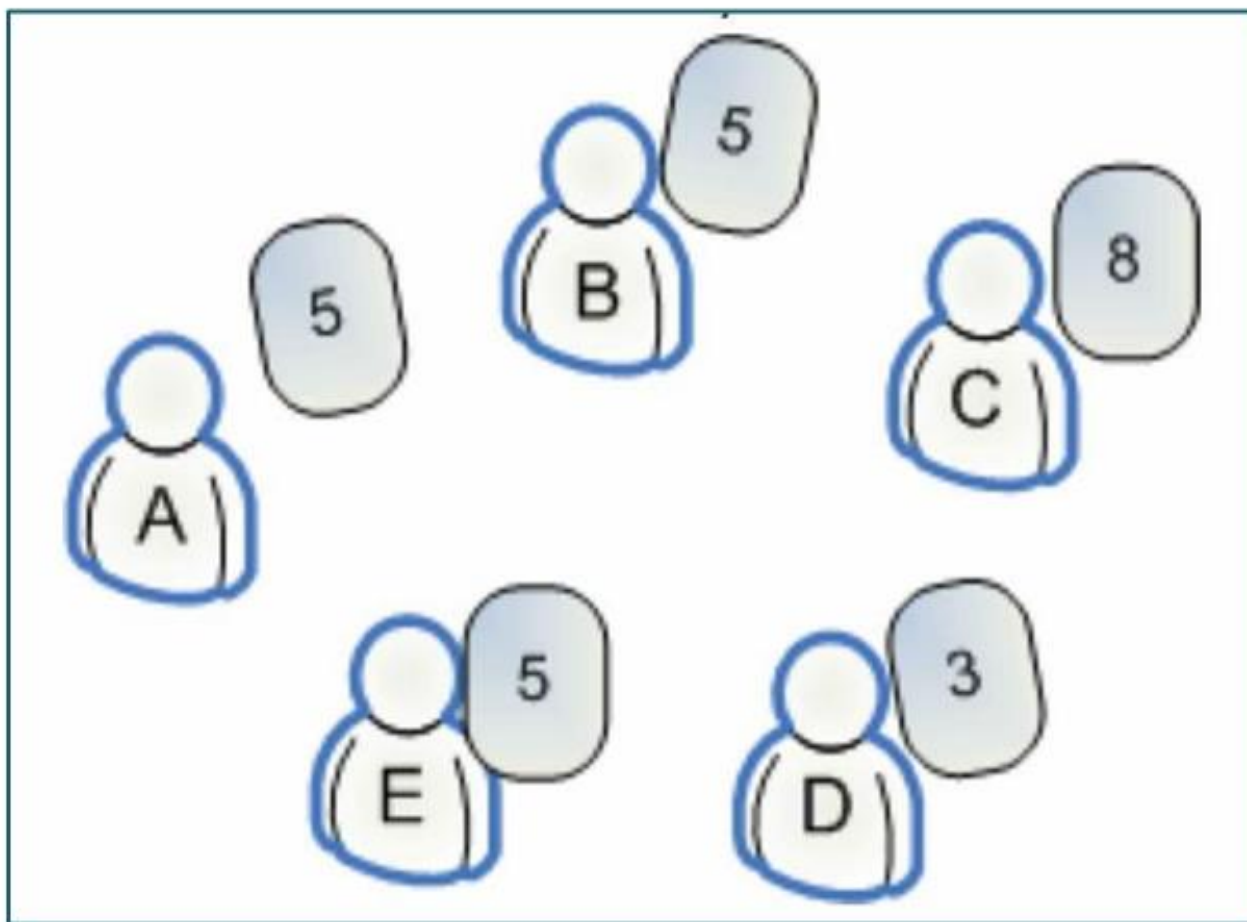


Nagy a különbség.
Megbeszélik.

A rájön, hogy egy
fontos feladatról
elfelejtkezett,

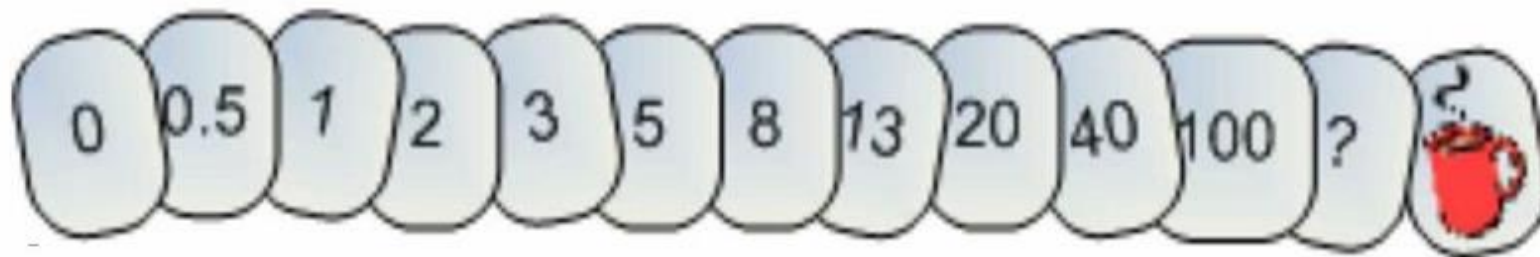
C pedig elismer,
hogy az A által
bemutatott
elképzeléssel a feladat
hamarabb
megoldható.

Becslés Planning Pokerrel



Eme következő licit
után
megegyeznek 5-
ben.

Becslés Planning Pokerrel



0

A feladat elkészült, vagy lényegileg semmi, pár perces munka

?

Fogalmam sincs. Ha túl gyakran használják,
akkor a user story-k nincsenek rendesen elemezve,
az információ nincs eléggé megosztva a csapatban



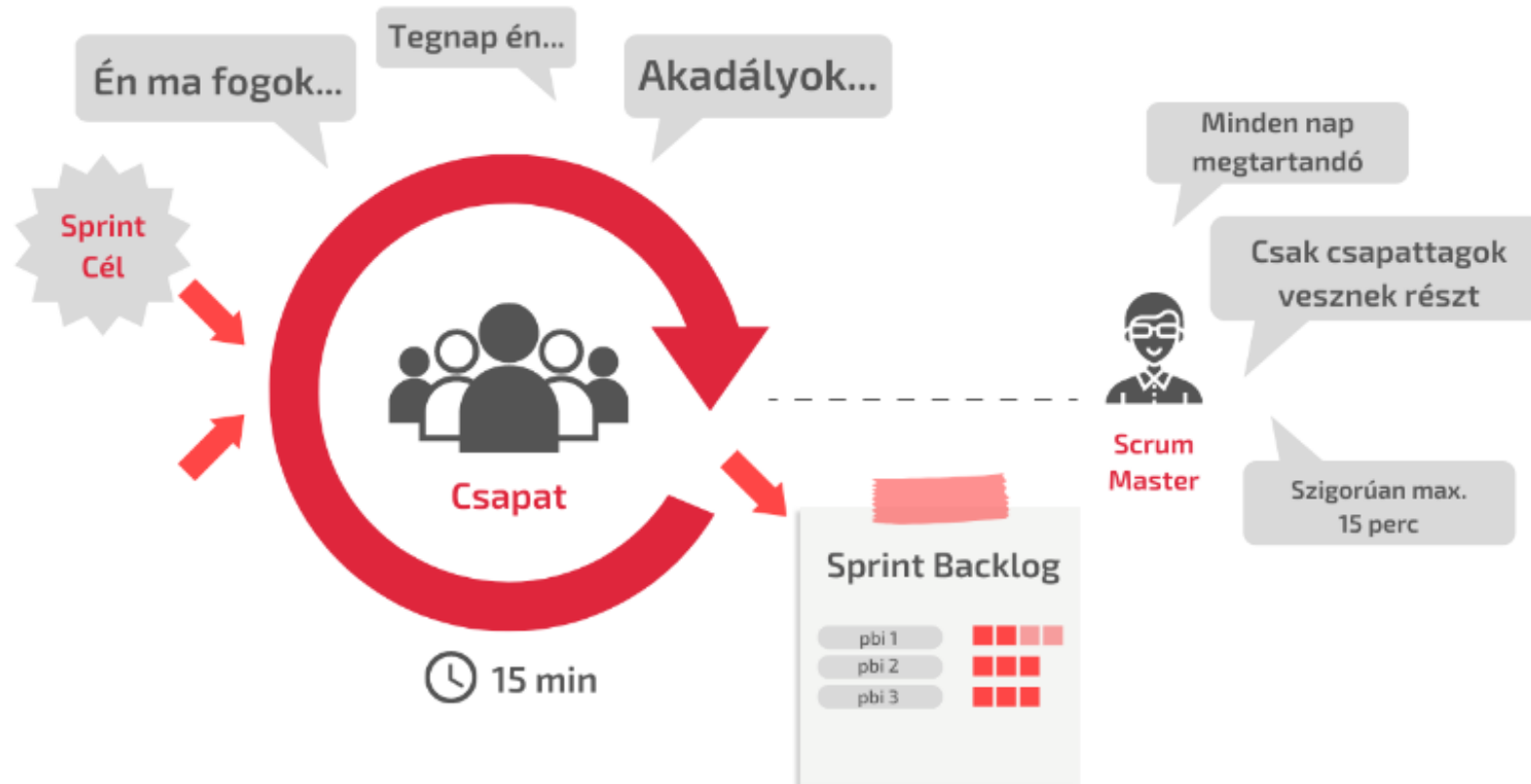
Fáradt vagyok, rövid szünetet kérek!

Napi SCRUM

- Azaz a **napi egyeztetés**.
- Mindennap ugyanabban az időben, **legfeljebb 15 percben** megvizsgálják, hogy teljesíthető-e a sprint cél, s azokat egyszerű kérdésekkel segítik, valamint szükség esetén újraterveznek annak érdekében, hogy a sprint célját a csapat el tudja érni.

Napi SCRUM

Daily Stand-up



Napi SCRUM – tipikus hibák

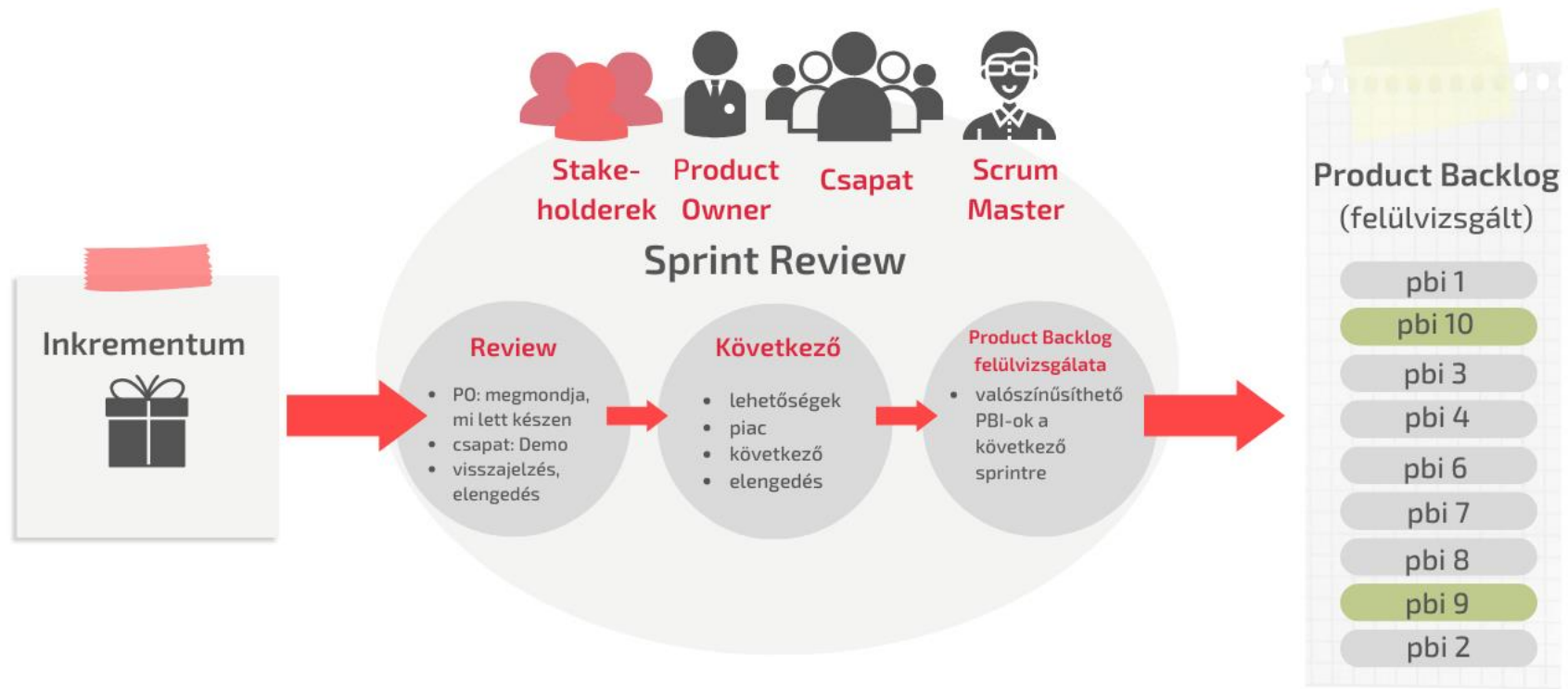
- Késve kezdődik
- Elmaradozik
- Elhúzódik
- A Scrum mesternek beszélnek
- Fecsegés
- Kísérlet a felmerülő problémák megoldására
- Nem a futamhoz tartozó problémákról beszélnek
- Belebeszél olyan valaki, aki nem tartozik a csapathoz
- Kétoldalú vagy kiscsoportos beszélgetések alakulnak ki
- Személyeskedés

Bemutató

- Ez a **futam felülvizsgálata**, egy olyan értekezlet, amely során a scrum csapat bemutatja munkájának eredményeit a kulcsfontosságú érdekelteknek (termékgazda mindenképp jelen van), és megvitatják a termékcél felé tett előrelépést.
- Ez egy mindenki számára **nyitott fórum**, és csak arról beszélnek a csapattagok, hogy mi az, amit késznek tekintenek, mi az ami már elvégzett feladatnak minősül.

Bemutató

Sprint Review



Bemutató – tipikus hibák

- **Nem történik** meg a futam lezárásaként
- Az ügyfél képviselője **nem vesz rajta részt**
- A **csapat valamely tagja** nem vesz rajta részt, mert például ő most nem készített bemutatható felhasználó történetet
- **Nem álltak rendelkezésre** kész kritériumok
- Olyan felhasználói történetet vesznek át, ami **nem teljesíti a kész kritériumokat**
- Az elfogadás nem a termékgazda **egyszemélyi döntése**
- Működő program helyett prezentációt tartanak arról, hogy **mit kellene a programnak tudnia**

Bemutató – tipikus hibák

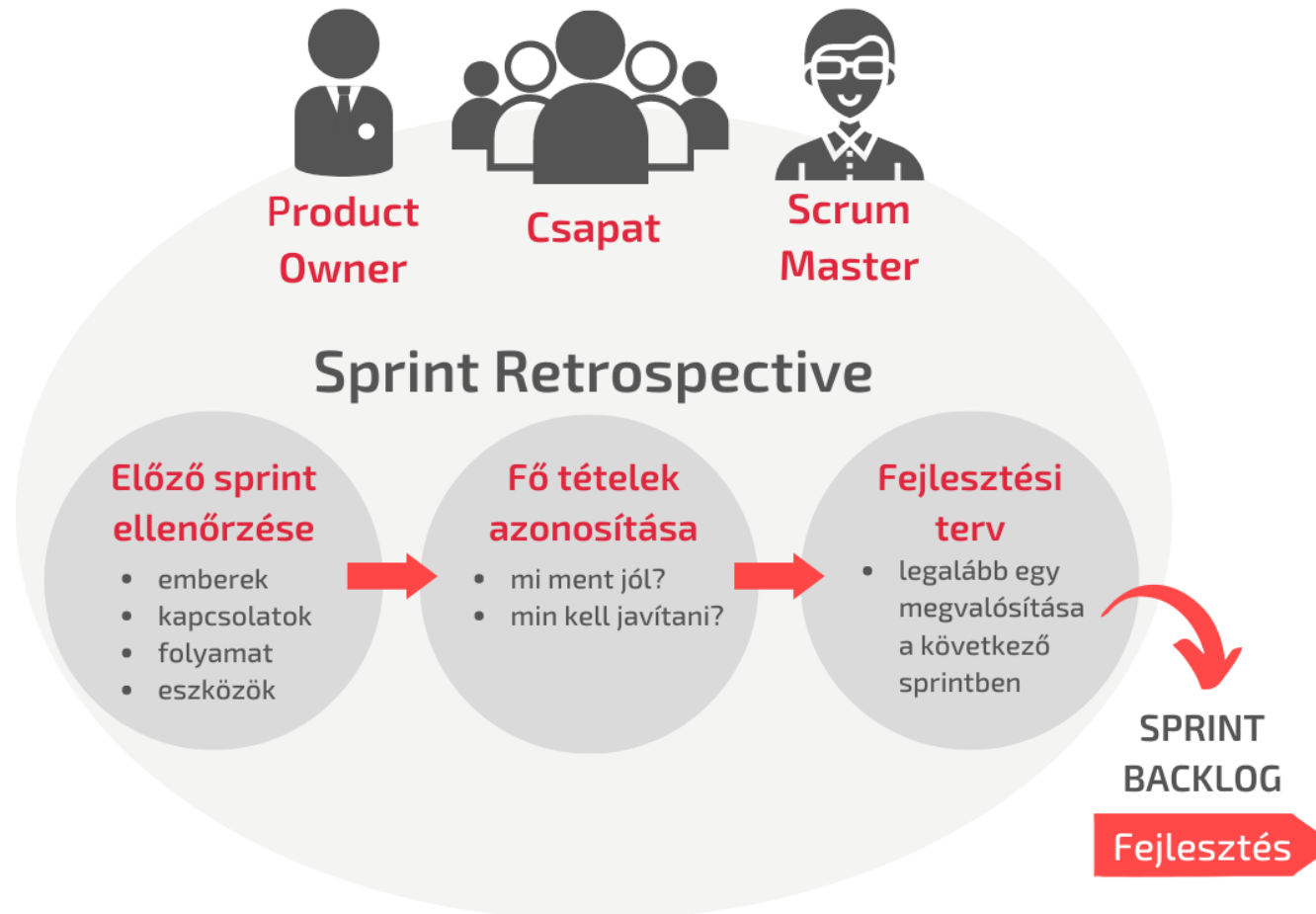
- **Nem a program(rész) készítője** tartja a bemutatót
- A bemutató **részként** próbálja a csapat összeállítani a működéshez szükséges környezetet
- Olyan felhasználói történetet próbál a csapat bemutatni, és így elfogadtatni, amelyről maga is tudja, hogy **nem teljesíti a kész kritériumokat** (hátha átmegy...)
- A bemutató során **felmerülő hibák elemzésével** töltik az időt

Visszatekintés

- A felülvizsgálatot követően a csapat leül egy hosszabb megbeszélésre, és áttekinti, **mi ment jól, mi mehetett volna jobban**, és min lehet egészen konkrétan javítani, hogy a következő sprint az előzőnél is sikeresebb, hatékonyabb legyen.
- Ehhez azonban fontos, hogy **a megbeszélés őszinte legyen**, és minden résztvevő releváns visszacsatolást adjon a munkájáról.
- Majd a csapat **megegyezik a javítás mikéntjéről**, kijelölnek egy (vagy több) javítandót, amelyet a következő sprint során végrehajtanak, és amely prioritást fog élvezni a fejlesztés produktivitásának szempontjából.

Visszatekintés

Sprint Retrospective



Visszatekintés – tipikus hibák

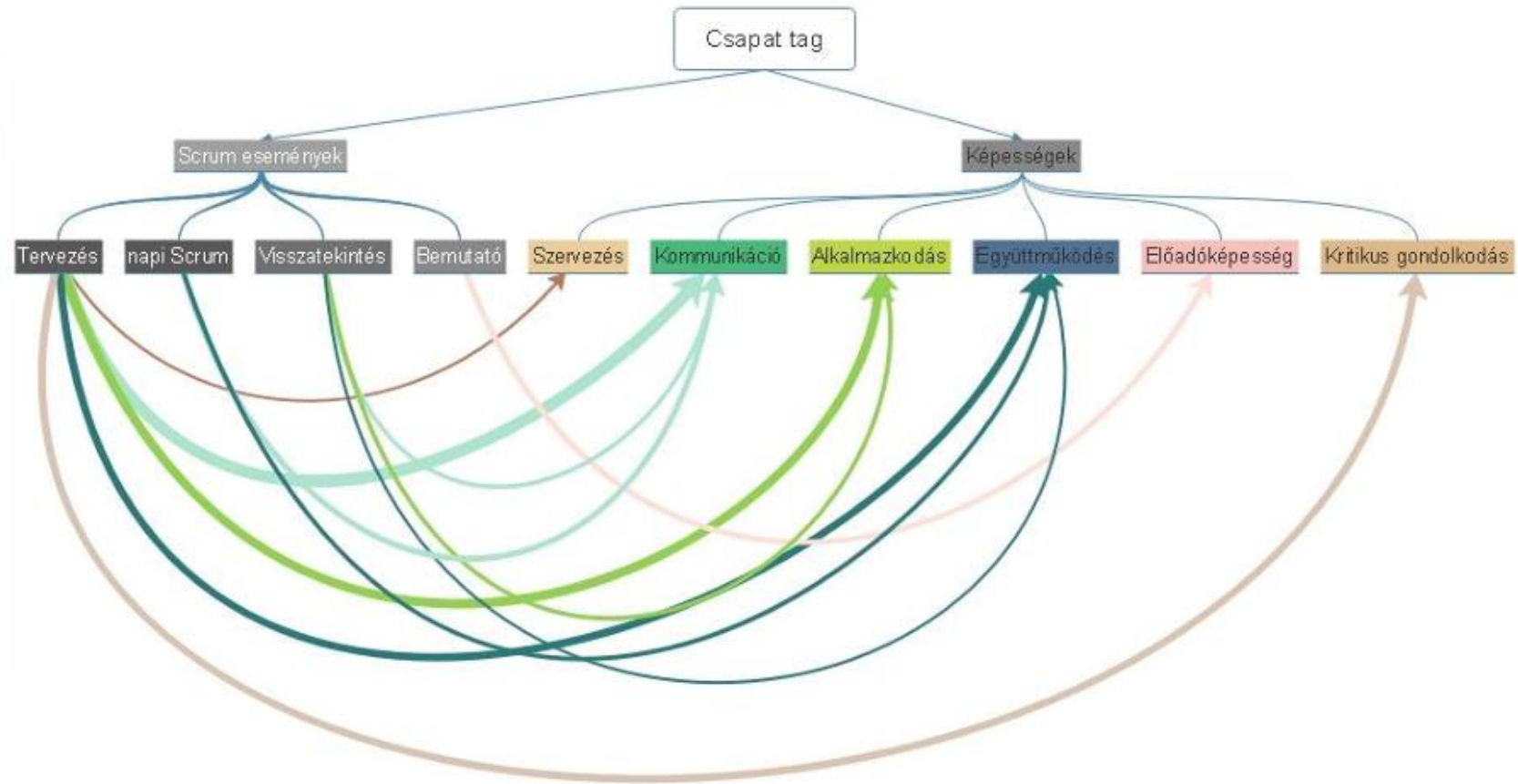
- Legdurvább hiba, ha **nincs visszatekintés**
- **Megelőzi** a bemutatót
- Formális: tudjuk le **minél gyorsabban**
- Külső partnerek **beleszólnak**
- **Személyeskedés**, személyeket értékelünk és nem a csapatot
- Végig vagy túlnyomóan a **Scrum mester beszél**
- A Scrum mester **kinyilatkoztatja** a szükséges folyamatjavító lépéseket
- A csapat valamely tagja folyamatosan **nem nyilvánít véleményt**

Maximális idő egy esemény kapcsán

| Események | Résztevői | Time-box (1 hónapos futam esetén) |
|----------------|---|--------------------------------------|
| Futam tervezés | Scrum mester, Termékgazda, Fejlesztő csapat | 8 óra |
| Napi Scrum | Fejlesztő csapat | 15 perc (mindig) |
| Bemutató | Scrum mester, Termékgazda, Fejlesztő csapat, Érintettek | 4 óra |
| Visszatekintés | Scrum mester, Termékgazda, Fejlesztő csapat | 3 óra |

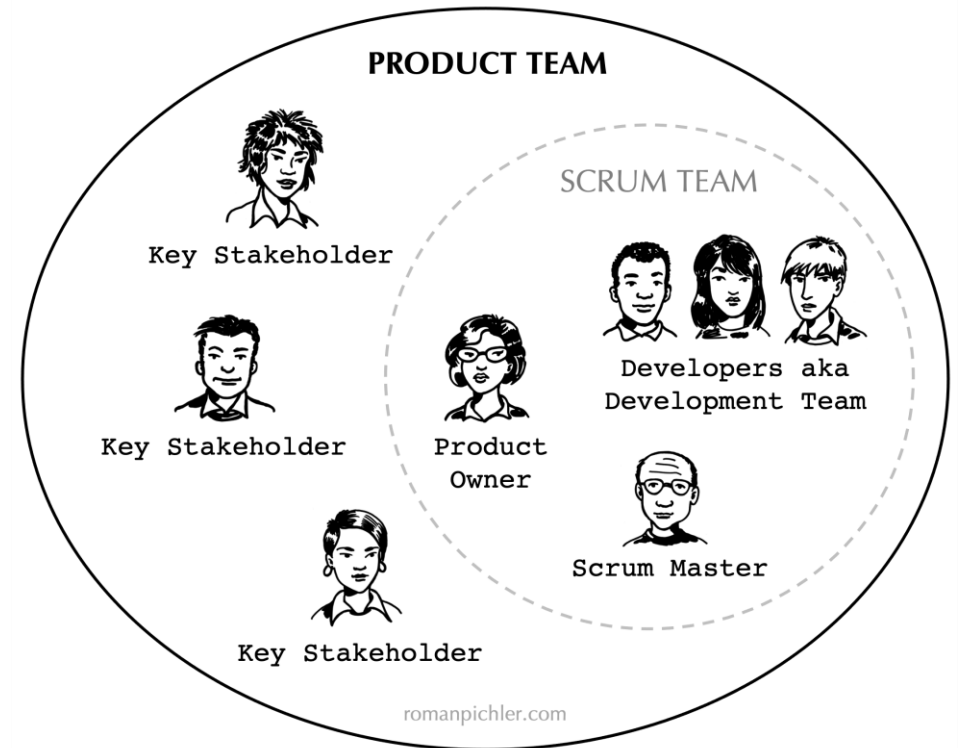
SCRUM csapattagok soft skilljeinek fontossága

| |
|--|
| Kommunikációs képesség |
| Együttműködési képesség |
| Szervező- képesség |
| Alkalmazkodó képesség |
| Analitikus és probléma megoldó képesség |
| Előadóképesség |



SCRUM csapat szerepkörei

- **Termékgazda** – PO (Product owner) – a vízionáló
- **Scrum master** (ceremónia mester) – a változásügynök
- **A fejlesztő csapat** (meghatározott létszámmal) – a szállítók
- **Ügyfél** (megrendelő) – a finanszírozó
- **A felhasználó** (User) – a haszonélvező



Termékgazda

- Az **ügyfél hangját** képviseli, átfogó műszaki, piaci és üzleti ismeretekkel rendelkezik
- A „**mit**” (a feladat) kérdésével foglalkozik, és nem a „**hogyan**”-nal
- Gondoskodik a termékvízióról, és **világossá teszi** a csapat számára
- Meghatározza a kibocsátási **dátumokat** és azok tartalmát
- Biztosítja, hogy a SCRUM csapat **üzleti szempontból hasznos** dolgokon dolgozik, azaz felelős a termék nyereségességéért (ROI)
- **Priorizálja az igényeket** üzleti értékük szerint, illetve szükség esetén futamonként módosítja azt
- **Elfogadja vagy visszautasítja** a futam eredményeit

A jó termékgazda jellemzői

- Mindig elérhető
- Üzleti tudás, bölcsesség
- Kommunikatív
- Döntésképes
- Felhatalmazással rendelkező
- Vezető típus, akit elfogad a csapat
- Jó kapcsolatteremtő (pl. az ügyféllel)

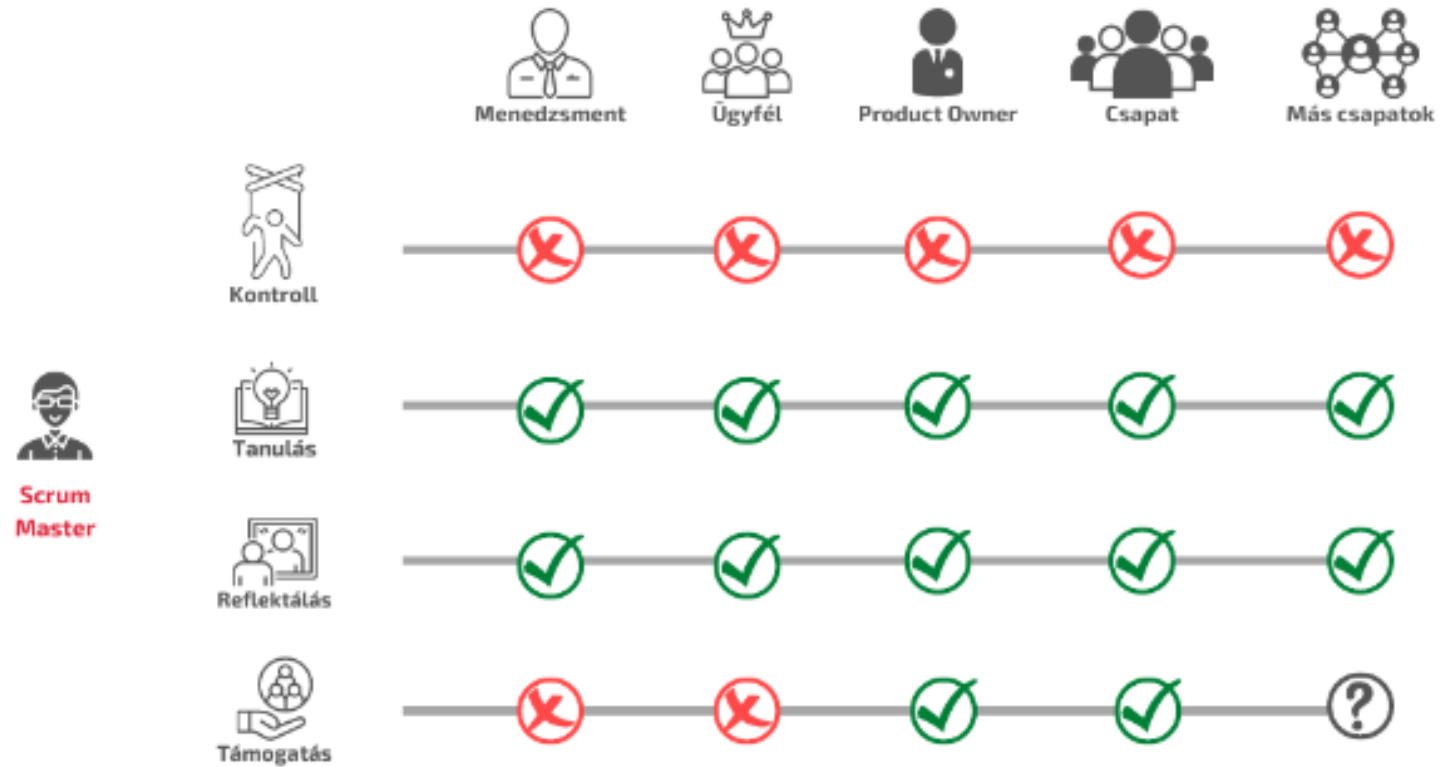
SCRUM Master

- A **folyamatokért felelős**
- **Tréner**, javító és „ajtónálló”, „team leader”
- Felelős a **SCRUM értékek** és gyakorlatok meghatározásáért
- A **napi SCRUM** résztvevője (de nem kötelezően!)
- Az akadályokat **elhárítja**
- Biztosítja a csapat **termelékenységét**
- Külső hatásoktól **védi** a SCRUM csapat munkáját

SCRUM Master

- Minden futam után **értékelő megbeszélést tart** a csapat tagjaival, ahol a tapasztalatokat és a következtetéseket levonják. Cél a tudás és a motiváció növelése a következő futamhoz
- **Nem** projektmenedzser (A projektmenedzseri funkciók a termékgazda, a Scrum mester és a csapat között megoszlanak)
- **Nincs hatalma** a csapattagok felett, de van hatalma a folyamatok felett

SCRUM Master szerepe



A jó SCRUM Master jellemzői

- Felelősségvállaló
- Szerény
- Együttműködő
- Elkötelezett
- Befolyásolási képességgel rendelkező
- Értelmes, jól informált

A fejlesztő csapat

- A **tényleges munkát** végzi
- Tipikusan **3-9 fős** csapat
- A csapattagok határozzák meg a **munka menetét és kiosztását**, a csapat önszerveződő. Ideális esetben nincsenek címek, rangok.
- Univerzális csapatok (tervező, programozó, tesztelő, stb.). Legjobb, ha **mindenki képes átvenni** másik munkáját.
- A csapat összetétele csak **futamok között** változik.
- A SCRUM mester és a termékgazda **nem tagja**.

A nagyobb csapat előnyei

- Szélesebb lehetséges **szakértelem**
- Kulcs személy kilépésének **kisebb a kockázati hatása**
- Nagyobb a **specializálódás** lehetősége

A kisebb csapat előnyei

- **Kevesebb** szociális lötyögés (3 fős csapat teljesítménye csak 2.5× az egyéni teljesítmények átlaga, 8 fős esetben ez a szám csak 4×)
- **Konstruktív interakció** gyakoribb kis csapatoknál
- Koordinációra **kevesebb idő** szükséges
- **Senki** nem bújhat meg háttérben
- Kisebb csapatban **jobban érzik magukat** az emberek
- Ártalmas túlspecializálódás **kisebb valószínűséggel** fordul elő

Ügyfél

- Általában ő **felel a költségért**
- **Követelőző lehet** érthető üzleti okokból, de ha belső folyamatokról van szó
- Ez a szerepkör tartozhat az ügyvezetőhöz is (azaz a nagyfőnökhöz), mindenképp elsőbbséget élvez, tekintve hogy az **egész folyamat gyökere**, a fejlesztés létjogosultsága hozzá, hozzájuk köthető.

A felhasználó

- A folyamat során a **visszajelzésért felelős**
- Kulcskérdés az együttműködés és a folyamatos információáramlás biztosítása, hiszen a végcél az ő **elégedettségének** elérése (legyen az külsős megrendelő, vagy belső szakterület).

SCRUM alapelvek

- A scrum esetében arról beszélhetünk, hogy minden egyes alapelv (bátorság, fókusz, nyitottság, elkötelezettség, tisztelet) **biztonságot nyújt**, komplex, összetett fejlesztési elakadások feloldását teszi lehetővé.
- Specialitása, hogy nemcsak ismert, de egyúttal **váratlan helyzeteket teremtő problémák megoldásában** is segít, útmutatást ad, gondolkodásra ösztönöz.

SCRUM alapelvek



SCRUM alapelvek

- Manapság még mindig inkább az a jellemző a **hagyományos módszerekkel** dolgozó vállalatoknál, hogy **funkcionális egységekben** gondolkodik a vezetés.
- Ez általában **lassítja a munkavégzést**, felesleges és számtalan megbeszéléseket indukál, a vezetők gyakran motiválatlanok, a munkatársaknak pedig nincs nagyobb rálátása a folyamatokra, mindenki csak a saját területét látja – ezáltal a kockázatok nőnek, a hatékonyság csökken.

Érdemes tehát átgondolni

- Adminisztráció **vs** keretfeltételek és alapelvek
- Hosszú és értelmetlen megbeszélések **vs** vezetett együttgondolkodás
- Nagy és bürokratikus részlegek **vs** önszerveződő csapatok
- Bonyolult, érthetetlen **vs** egyszerű, kevesebb lépésből álló döntési folyamat
- Centralizált tudás **vs** tudásmegosztás, információáramoltatás
- Átláthatatlanság **vs** transzparencia
- Hibáztatás **vs** egészséges “hibakultúra”
- A kudarcok figyelmen kívül hagyása **vs** állandó visszatekintés

Érdemes tehát átgondolni

- A scrum alapvetően tehát **egyetlen csapat működését** támogatja. Szervezeti működés esetén több csapat összehangolása, a skálázott agilitás megvalósítása minden cégnél egyedi megközelítést igényel.
- A sport példájával zárva – egy csapatot lehetséges egy idényre jól összepakolni, amennyiben van megfelelő edző, azonban bejutni a Bajnokok Ligájába, és ott több éven keresztül is potenciális esélyekkel labdába rúgni, már sokkal inkább **múlik a csapatmunkán**.

Termékek

- Termék kívánságlista (Product backlog)
- Felhasználói történet (User story)
- Futam feladatlista (Sprint backlog)

Termék kívánságlista

- Követelmények
- A projekt elvárt munkáinak listája
- Ideális esetben olyan elemek, amelyek az ügyfél számára üzleti értékkel bírnak
- A termékgazda priorizálja
- Minden egyes futam megkezdése előtt újra priorizálásra kerül

Felhasználói történet

- A termék kívánságlista ebből áll.
- A felhasználói történet egy funkció vagy jellemző rövid, egyszerű leírása azon személy (általában a felhasználó vagy megrendelő) által megfogalmazva, aki ezt a új képességet szeretné.
- Tipikusan leírható az alábbi mintával:
As a <type of user>, I want <some goal> so that <some reason>
vagy
In order to <achive value>, as a <type of user>, I want <some goal>
- A felhasználói történetnek van neve (azonosítója), leírása, elfogadási teszt forgatókönyve (kész/Done kritériuma)

Kritikus szempontok (3C)

- A felhasználói történetek megírása, illetőleg az üzleti követelmények felhasználói történetekre való felbontása során az alábbi pontokban megfogalmazott szempontokat kell figyelembe venni:
- **Kártya (Card)** - a felhasználói történeteket egy tényleges kártyára (fizikai kártya vagy bejegyzés egy elektronikus táblán) írják fel.
- **Beszélgetés (Conversation)** - a felhasználói történet készítése során a szoftver várt funkcionalitását be kell mutatni (lehet szóban vagy dokumentáltan).
- **Megerősítés (Confirmation)** - az elfogadási feltételekben kerülnek rögzítésre.

Kritikus szempontok (3C)

- Egy felhasználói történet elfogadási feltételei azok a feltételek, amelyeknek a felhasználói történet megvalósítása során teljesülniük kell ahhoz, hogy a teljesítést az érdekelt felek elfogadják.
Az **elfogadási feltételeket tekinthetjük tesztfeltételeknek**, amelyeket a teszteknek ki kell elégíteniük, vagyis a teszteseteket az elfogadási feltételekből származtatják.
- A teszteket manuálisan és automatizált módon is végrehajthatják.
- Az elfogadási kritériumok **általában funkcionálisak**, azonban figyelni kell arra is, hogy a nemfunkcionális követelményekről és az azokhoz kapcsolódó tesztesetekről se feledkezzünk meg.

Kritikus szempontok (3C)

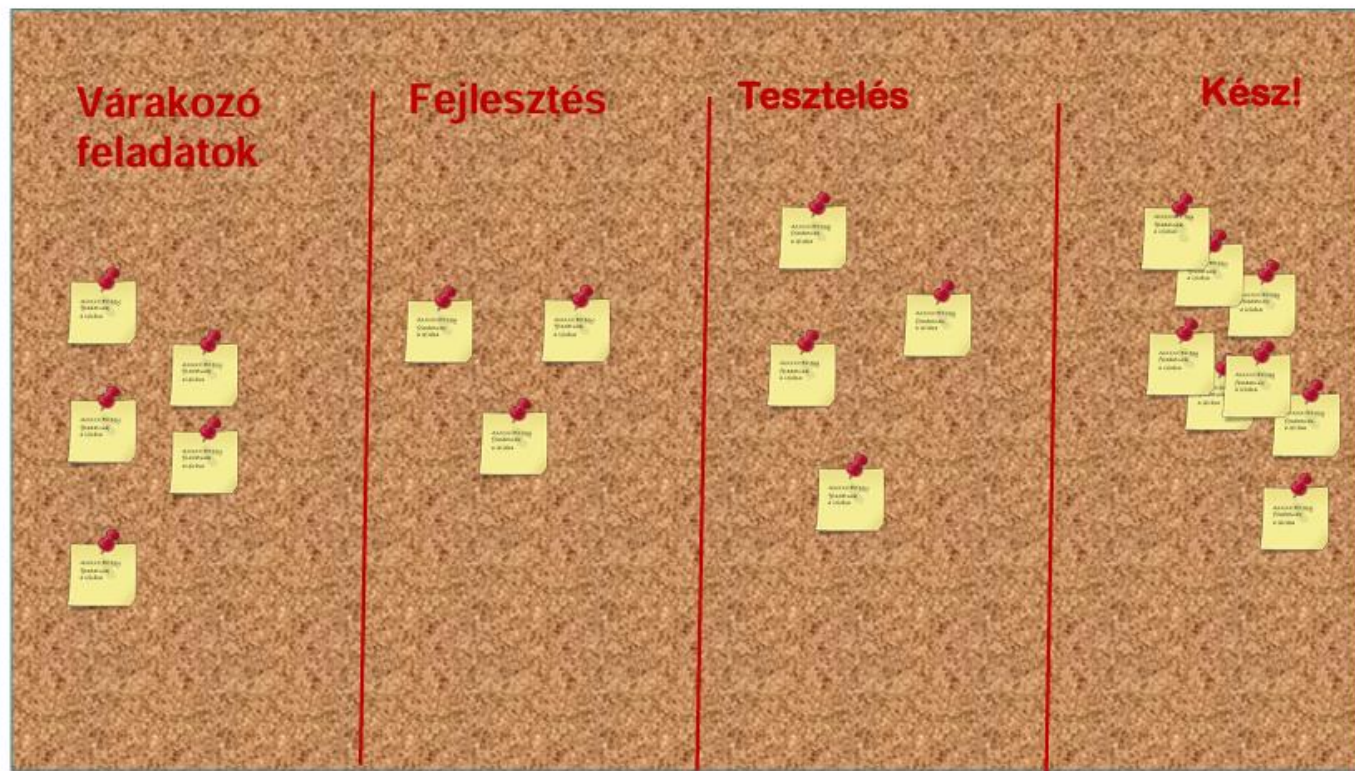
- A felhasználói történetek a projekthez kapcsolódó epicekből (*az epic olyan komplex projektfeladat, amely az üzleti követelmények összességét tartalmazza*) származnak, és az egy fejlesztési ciklusban (például sprint során a SCRUM-ban) megoldandó feladatokat tartalmazzák.
- Mikor a feladattervezés történik, a csapatnak **becslést kell adni** a feladat komplexitására, illetőleg a fejlesztéshez szükséges ráfordításokra.
- A becslés eredménye egy olyan mérőszám, amely tükrözi a feladat megoldásához szükséges erőfeszítés mértékét és a csapattagok számára ugyanazt jelenti. (Planning Poker)

Felhasználói történet példák

- Felhasználóként regisztrálni kell magamat felhasználói névvel és nem triviális (legalább 8 karaktert, nagybetűt, kisbetűt és számot egyaránt tartalmazó) jelszóval azért, hogy később biztonságosan be tudjak jelentkezni a rendszerbe.
- Webes könyvtárház felhasználójaként szeretném látni a legnépszerűbb száz könyv listáját azért, hogy egyet vagy többet ki tudjak választani belőle megvásárlás céljára.
- Webes könyvtárház felhasználójaként szeretném rendezni a legnépszerűbb száz könyv listáját áruk szerint úgy, hogy a legolcsóbb legyen legelöl.

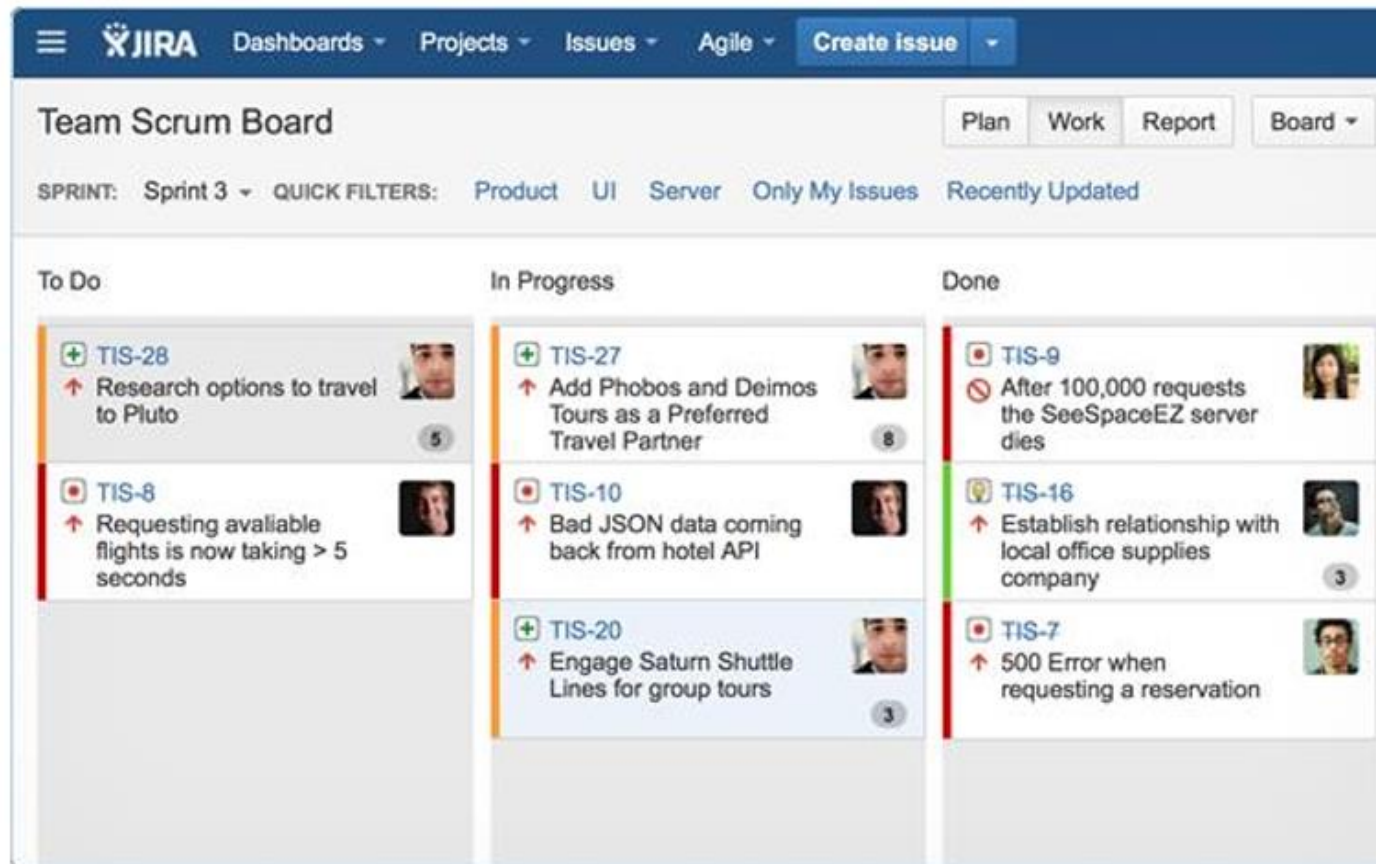
Scrum kapcsán gyakran használjuk

Scrum tábla részlet



Scrum kapcsán gyakran használjuk

Jira – Scrum tábla részlet

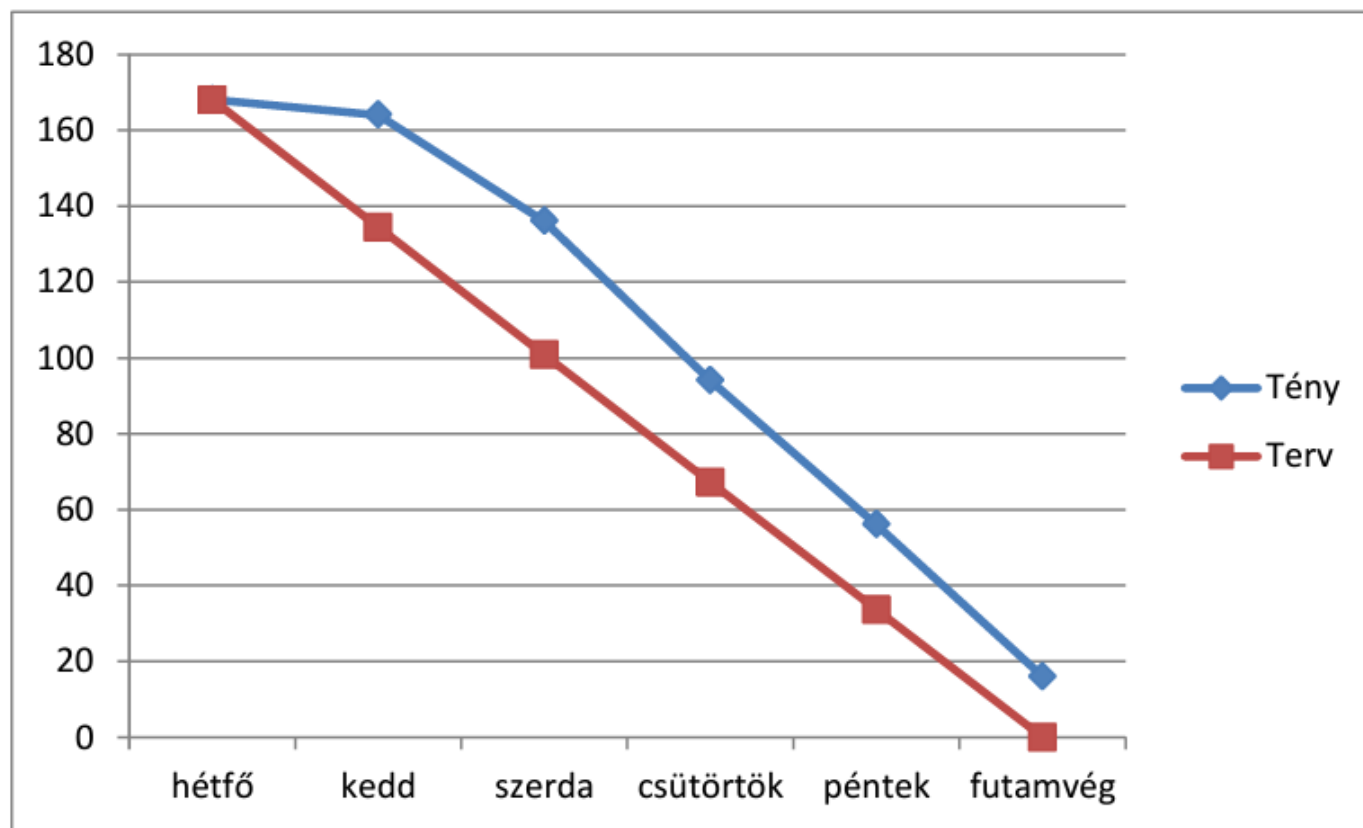


The screenshot displays a Jira Team Scrum Board for 'Sprint 3'. The board is organized into three columns: 'To Do', 'In Progress', and 'Done'. Each column contains task cards with IDs, descriptions, and assignee avatars. The 'To Do' column has two cards, 'In Progress' has three, and 'Done' has three.

| Column | Task ID | Description | Assignee | Count |
|-------------|---------|---|----------|-------|
| To Do | TIS-28 | Research options to travel to Pluto | [Avatar] | 5 |
| | TIS-8 | Requesting available flights is now taking > 5 seconds | [Avatar] | |
| In Progress | TIS-27 | Add Phobos and Deimos Tours as a Preferred Travel Partner | [Avatar] | 8 |
| | TIS-10 | Bad JSON data coming back from hotel API | [Avatar] | |
| | TIS-20 | Engage Saturn Shuttle Lines for group tours | [Avatar] | 3 |
| Done | TIS-9 | After 100,000 requests the SeeSpaceEZ server dies | [Avatar] | |
| | TIS-16 | Establish relationship with local office supplies company | [Avatar] | 3 |
| | TIS-7 | 500 Error when requesting a reservation | [Avatar] | |

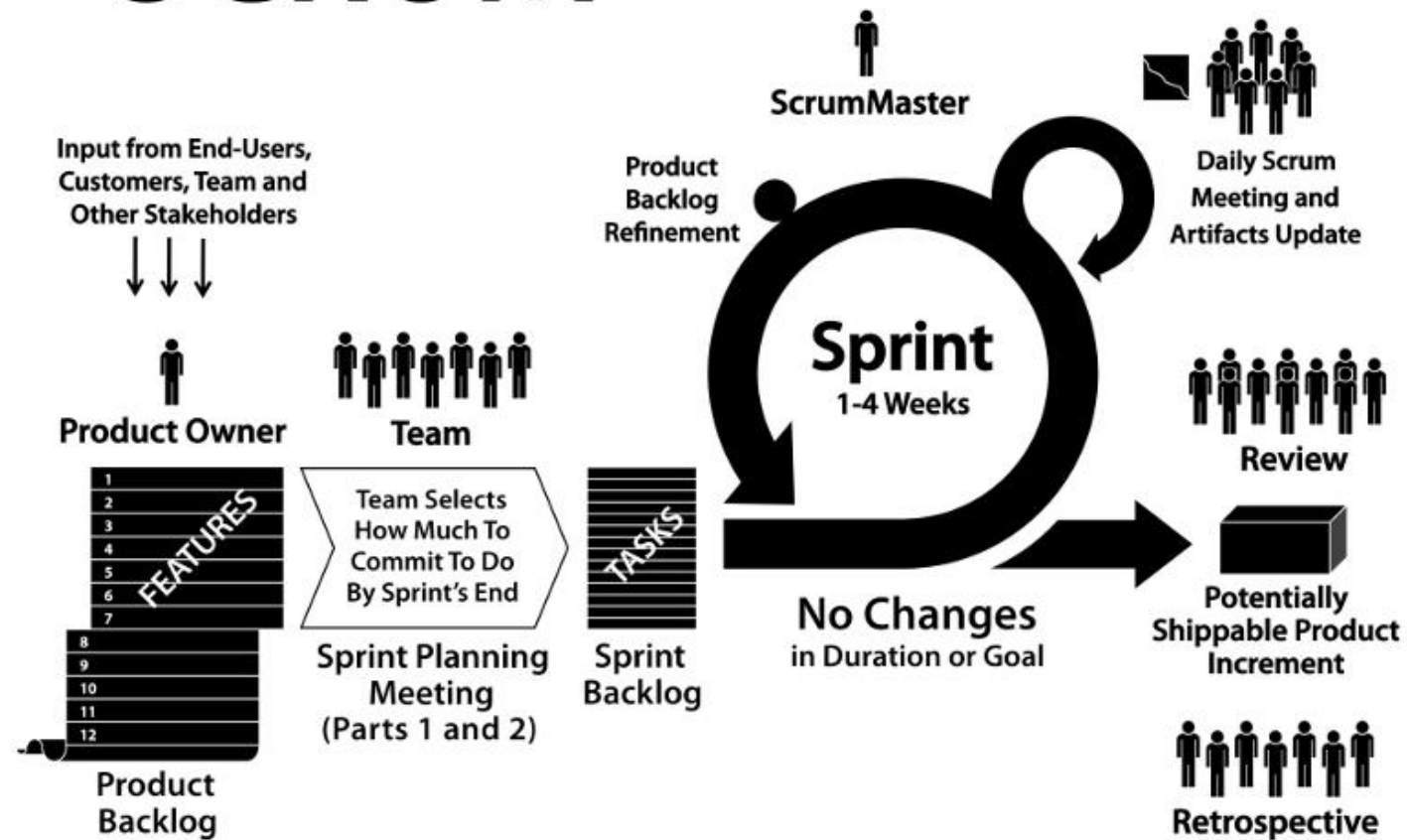
Scrum kapcsán gyakran használjuk

Haladási Diagram



Összefoglalás

SCRUM





Köszönöm a figyelmet!