

Agilis fejlesztési megközelítés

Waterfall project management

A vízesés modell, szakaszokból épül fel. Ez a stílus mindent egyetlen, nagy kockázatú lapra tesz fel. Ha egy projekt áthalad egy fázison, fájdalmas annak újbóli meglátogatása, mert a csapatok mindig előre lépnek a következő szakaszba.



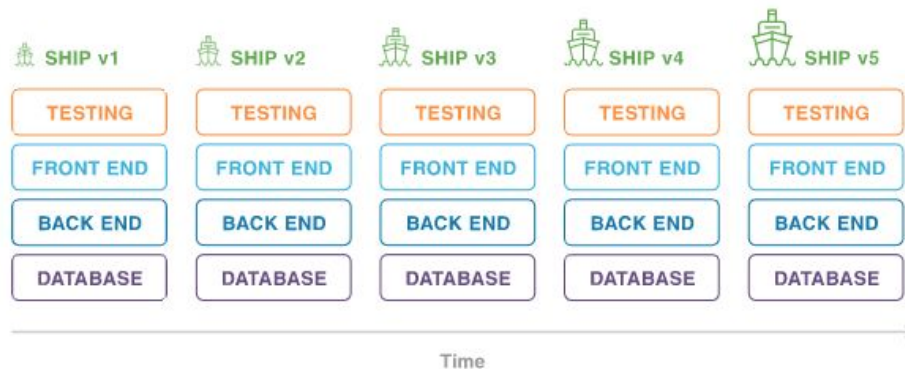
A hagyományos projektmenedzsment stílusok gyakran "kritikus útvonalakat" hoznak létre, ahol a projekt nem haladhat tovább, amíg egy blokkoló probléma nem oldódik meg. A végfelhasználó csak akkor léphet kapcsolatba a termékkel, amikor az **teljesen elkészült**. Így a termék kialakításában és kódjában a fontos problémákat a kiadásig nem fedezik fel.





Agilis project management

Rendszeres visszacsatolási időközönként **iteratív megközelítést** alkalmaz a fejlesztéshez. Ezek az iterációk lehetővé teszik, hogy a csapat átirányításra kerüljön (és eredményes legyen) a projekt egy másik területén, miközben a blokkoló kérdés megoldódik.



A kritikus útvonalak eltávolításán túl az iterációk lehetővé teszik a termékkel való interakciót a fejlesztés során.

Ez viszont állandó lehetőségeket kínál a csapat számára az építéshez (build), szállításhoz (release), tanuláshoz és alkalmazkodáshoz. A piaci (váratlan) változások nem okoznak akkora károkat, és a csapatok készen állnak arra, hogy gyorsan alkalmazkodjanak az új követelményekhez.



Agilis szoftverfejlesztés

Az agilis projektmenedzsment egy iteratív megközelítés a szoftverfejlesztési projektek irányításához, amely a **folyamatos kiadásokra** összpontosít, és az **ügyfelek visszajelzéseit** belefoglalja minden iterációba.

A hagyományos agilis projektmenedzsment két keretbe sorolható: **scrum** és **kanban**. Míg a scrum a rögzített hosszúságú projekt iterációkra összpontosít, addig a kanban a folyamatos kiadásokra összpontosít. A befejezés után a csapat azonnal tovább megy a következőre.



Agilis szoftverfejlesztés

Előnyben részesíti:

- Egyének és folyamatok
- Működő szoftver
- Együttműködés a megrendelővel
- Reagálás a változásra (úgy igény)

Kevésbé összpontosít:

- Eszközökre
- Átfogó dokumentációkra
- Folyamatos szerződésre hivatkozásra
- Egy leszögezett terv követésére

Ezek az elvek (részben) léteztek az agilis fejlesztések előtt is, innentől viszont már rögzített formában volt, ami után az emberek az egész elvét átvették és elkezdték alkalmazni a saját munkájukra



Agilis kifejezések

- Roadmap (ütemterv)
- Requirements (követelmények)
- Backlog (várólista..etc)
- Epics
- User Stories
- Estimation

Roadmap (ütemterv)

Az ütemterv bemutatja, hogyan fejlődik egy termék vagy megoldás az idő múlásával. Az ütemtervek olyan részletekből (initiative) állnak, amelyek egy adott funkcionalitást lefednek, és hozzávetőlegesen jelzik, hogy az adott funkcionalitás mikorra készül el.

Ahogy a program fejlődik elfogadottnak számít, hogy az ütemterv megváltozik - néha finoman, néha nagymértékben. A cél az, hogy az ütemtervet a jelenlegi piaci körülményekre és a hosszú távú célokra összpontosítson.



Requirements (követelmények)

Az ütemterv minden kezdeményezése (initiative) követelményekre osztható. Az agilis követelmények a szükséges funkcióknak a könnyű leírása, nem pedig a hagyományos projektekhez kapcsolódó 100 oldalas dokumentumok.

Idővel fejlődnek, és kihasználják a csapat megértését az ügyféllel és a kívánt termékkel kapcsolatban.

Az agilis követelmények továbbra is karcsúak, miközben a csapat tagjai a folyamatos beszélgetés és együttműködés révén közös megértést alakítanak ki.

Backlog

Az **backlog** határozza meg az agilis program prioritásait.

A csapat magába foglalja az összes munkadarabot a backlogban: új funkciók, hibák, fejlesztések, műszaki vagy építészeti feladatok stb.

A terméktulajdonos a mérnöki csapat számára priorizálja a hátralévő munkát.

A fejlesztői csapat ezután a prioritási sorrendben lévő backlogot használja az elvégzendő munkához.

User Story

A legkisebb munkaegység egy agilis keretrendszerben. Ez egy végcél, nem funkció, a szoftver felhasználó szempontjából kifejezve.

A felhasználói történet célja annak megfogalmazása, hogy az elvégzett story egy adott munkát végrehajtson. Az "ügyfeleknek" nem kell, hogy a hagyományos értelemben vett külső végfelhasználók legyenek.

A felhasználói történetek néhány mondat egyszerű nyelven, amelyek körvonalazzák a kívánt eredményt. Nem mélyülnek a részletekbe. A követelmények később kerülnek beillesztésre, miután a csapat megállapodott.

Epic

Az **epic** egy olyan egység, amelyet fel lehet bontani konkrét feladatokra (úgynevezett „**user story**-ra”) az ügyfelek vagy a végfelhasználók igényei alapján.

Az **epic-ek** hasznos keretet adnak a munka megszervezésére és hierarchia létrehozására. Az ötlet az, hogy a munkát kiadható darabokra bontja. Az epic-ek segítik a csapatokat a feladatok lebontásában, miközben folytatják a munkát egy nagyobb cél elérése érdekében.



User Story

- A felhasználókra összpontosítanak
- Lehetővé teszik az együttműködést
- Ösztönzik a csapatot, hogy kreatívan gondolkodjanak azon, hogyan lehet a végső célt legjobban megoldani.
- lendületet adnak. A fejlesztő csapat minden egyes történetet követően kis kihívásokkal és kis nyereményekkel jár, lendületet adva.



Estimation (becslés)

Figyelembe kell vennie számos olyan tényezőt, amelyek segítenek a terméktulajdonosoknak (PO) az egész csapatot és az üzletet érintő döntések meghozatalában.

Amikor a mérnöki csapat megkezdí a becslési folyamatot, általában kérdések merülnek fel a követelményekkel és a felhasználói történetekkel kapcsolatban. És ez jó: ezek a kérdések segítik az egész csapatot, hogy jobban megértse a munkát. Kifejezetten a terméktulajdonosok számára a feladatok darabra bontása és a becslések összessége segít nekik a munka összes (és potenciálisan rejtett!) területének a prioritizálásában. És amint megkapják a fejlesztő csapat becsléseit, nem ritka, hogy egy terméktulajdonos átrendezi a backlog elemeit.

A becslés egy magas szintű tevékenység így ha sokáig tart újabb kérdéseket kell felvetni. nem megbecsülhető storyt (hiányzó specifikáció, vagy könnyedén változhat a jövőben) becslés nélkül kell hagyni.

A pontozásos alapú becslés révén a csapat tagjai az adott story nehézségeit veszik számításba. Így a fejlesztés komplexitására tudnak összpontosítani, és nem az eltöltött időre. (fibonacci pontosítás)

A pontozás csapatra vonatkoztat (10 pont seniornak vs 10 pont juniornak VS ha ugyanez órában megadva) A múlt becsléseiből is lehet tanulni.

PRODUCT BACKLOG REFINEMENT

PRODUCT BACKLOG



PRIORITISED PRODUCT BACKLOG



Hosszútávú agilis projekttervezés

1. Start with the big picture
2. Nagyobb feladatok azonosítása
3. Feladatok kisebb egységekre való osztása
4. Becslés
5. Kis release-ek tervezése
6. Roadmap
7. Roadmap validálása a csapattal
8. Folyamatos javulás

- Agilis módszertan: **gondolkodásmód**.
Kiindulópont, hogy a tervek változhatnak. A cél világos, a hozzá vezető út változhat.
- Agilis módszertan egyik célja: bürokrácia csökkentése
- Agile alapgondolata: egy olyan cégnek, amely „túl akar élni” rugalmasnak kell lennie, és alkalmazkodnia kell a változásokhoz.
- Agilis munkavállaló:
 - tudásmegosztás
 - kreativitás
 - kezdeményezés
 - jó probléma megoldás
- SCRUM: **keretrendszer**

Agile & SCRUM



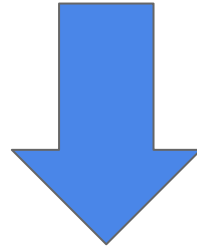
Scrum születése

- **1986**

[Ikujiro Nonaka és Hirotaka Takeuchi](#) *The New New Product Development Game* című publikációja → legjobb eredmény kis csapatban

- **1966**

Jeff Sutherland és Ken Schwaber által létrehozott fejlesztési módszert



SCRUM

Scrum meghatározása

- **Scrum (főnév):** Egy olyan keretrendszer, melynek segítségével emberek komplex problémákat tudnak adaptív módon kezelni úgy, hogy közben termelékenyen és kreatívan szállítják le a lehető legértékesebb terméket.
 - Egyszerű
 - Könnyen érthető
 - Rendkívül nehezen művelhető mesteri szinten
- Miből áll?
 - Scrum Csapat és hozzájuk rendelt szerepekből, eseményekből, munkaanyagokból és szabályokból
- A Scrum szabályai kapcsolják össze az eseményeket, szerepköröket és a munkaanyagokat, meghatározva a köztük lévő viszonyokat és kölcsönhatásokat

SCRUM elmélet

- Tapasztaláson alapuló folyamatellenőrzési elméleten alapul.
- A tudás a tapasztalatokból és az adott ismeretleneken alapuló döntésekből ered.
- Iteratív, inkrementális megközelítést alkalmaz a kiszámíthatóság optimalizálása és a kockázat kézben tartása érdekében.
- **Transzparencia (Átláthatóság)**
A folyamat lényeges nézőpontjainak láthatónak kell lenni azok számára, akik felelősek az eredményért. A nézőpontokat egy közös szabvány szerint kell meghatározni, hogy mindenki számára érthetőek legyenek.
- **Ellenőrzés**
A felhasználóknak gyakorta kell ellenőrizni a Scrum munka anyagait, a cél felé történő haladást, hogy időben érzékelhető legyen a nem kívánatos eltérés, de ez nem lehet olyan gyakori, hogy akadályozza a munkát.
- **Korrekción**
Amennyiben megállapításra kerül, hogy egy folyamat egy vagy több szempontból nem megfelelő, vagy, hogy a végtermék nem lesz megfelelő, akkor módosítani kell a folyamaton.

Scrum szereplők

- *Project manager*
- PO - Product Owner
- BA - Business Analyst
- SM - Scrum Master
- *Rendszerszervező*
- UX/UI
- Fejlesztő: FE, BE, adatbázis szakértő, DevOps
- Tesztelő



Scrum szereplők - Scrum Master

- Felelős a Scrum folyamat irányításáért
- Napi scrum, sprint tervezés, sprint demo és retrospektív találkozók megszervezése és megkönnyítése
- Felmerült akadályok elhárítása
- Megpróbálja javítani a fejlesztői csapat hatékonyságát
- Segít a terméktulajdonosnak, hogy a termék backlog jó állapotban legyen és készen álljon a következő sprintre
- Óvja a csapatát



Scrum szereplők - Product Owner

- Felelőssége, hogy a fejlesztő csapat munkájának eredményeként előálló termék értéke maximális legyen.
- Nem írhatja elő a csapat tagjainak, hogy ki, mit mennyi idő alatt és hogyan valósítson meg. —> PO nem irányítja a csapat tagjait csak a megfelelő irányba tereli a figyelmet.
- Priorizálás:
 - A csapat figyelmét az adott pillanatban legtöbb értéket teremtő célkitűzés felé irányítsa.
 - Képesnek kell lennie azonosítani azokat a problémákat és célokat, amikkel adott pillanatban leginkább érdemes foglalkoznia a csapatnak.
 - Ezek a Product Backlogban kerülnek vezetésre
- Tervezés és ütemezés:
 - Rendelkezik az elképzeléssel, hogy minek mikorra kell elkészülnie —> ez az elképzelést nevezhetjük magának a tervnek is
 - Folyamatosan gyűjtenie kell a kiadott termékverziókra érkező visszajelzéseket és beépíteni a projekt tervbe



Scrum szereplők - Business analyst

- Üzleti elemző
- Többségében olyan személyek, akik üzleti jellegű számítástechnikai rendszerek elemzésével, fejlesztésével, meghatározásával kapcsolatos munkát végeznek
- Fő feladata a megbízó részére készítendő szoftverrel szemben támasztott elvárások precíz dokumentálása
- További feladata, hogy közvetítsen a megrendelő és az informatikai szállító (Scrum csapat) között.



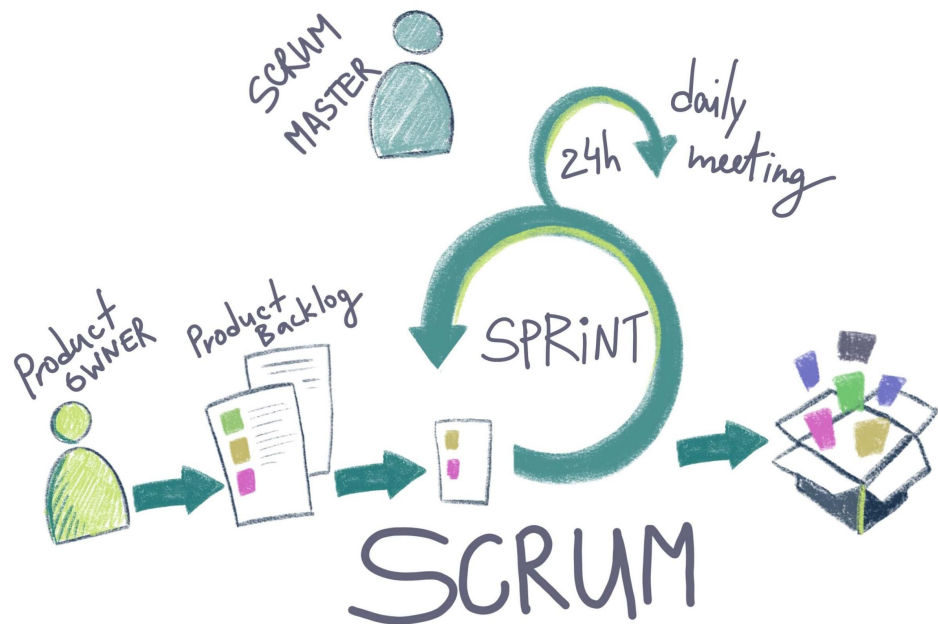
Scrum szereplők - Rendszerszervező

- Egyes módszertanokban nincs is ilyen személy —> SCRUM
- Nincs egységes megnevezés erre a munkakörre és a valós feladatok is eltérőek munkakörnyezetenként (néhol nagyon specifikus, néhol viszont projektvezetői feladatokig terjed)
- Köztes réteg az üzleti terület(ek) és a fejlesztők/IT között
- Kis csapatnál nem életszerű, de egy 10 fős fejlesztő csapatnál már hasznos lehet, ez fölött pedig nélkülözhetetlen
- Legfontosabb feladata az információ analízis, jellemzően nem kell hatalmas újításokat kitalálni, sokkal inkább a meglévő lehetőségek ismeretében hatékony megoldásokat szállítani/kommunikálni az érintett felekkel



SCRUM események

- **Célja**, hogy rendszerességet, szabályszerűséget teremtsenek, valamint minimalizálják az egyéb, Scrumban nem meghatározott megbeszélések szükségességét.
- Az események **időkorlátosak**
- Minden Scrum-esemény egy **formális lehetőség** valaminek az ellenőrzésére és korrekciójára.
- Bármely esemény elhagyása csökkenti az átláthatóságot, és elvesz az ellenőrzés és a korrekció lehetősége.



© /luminousmen.com






Roles

-  Scrum Master
-  Scrum Team
-  Product Owner

Events

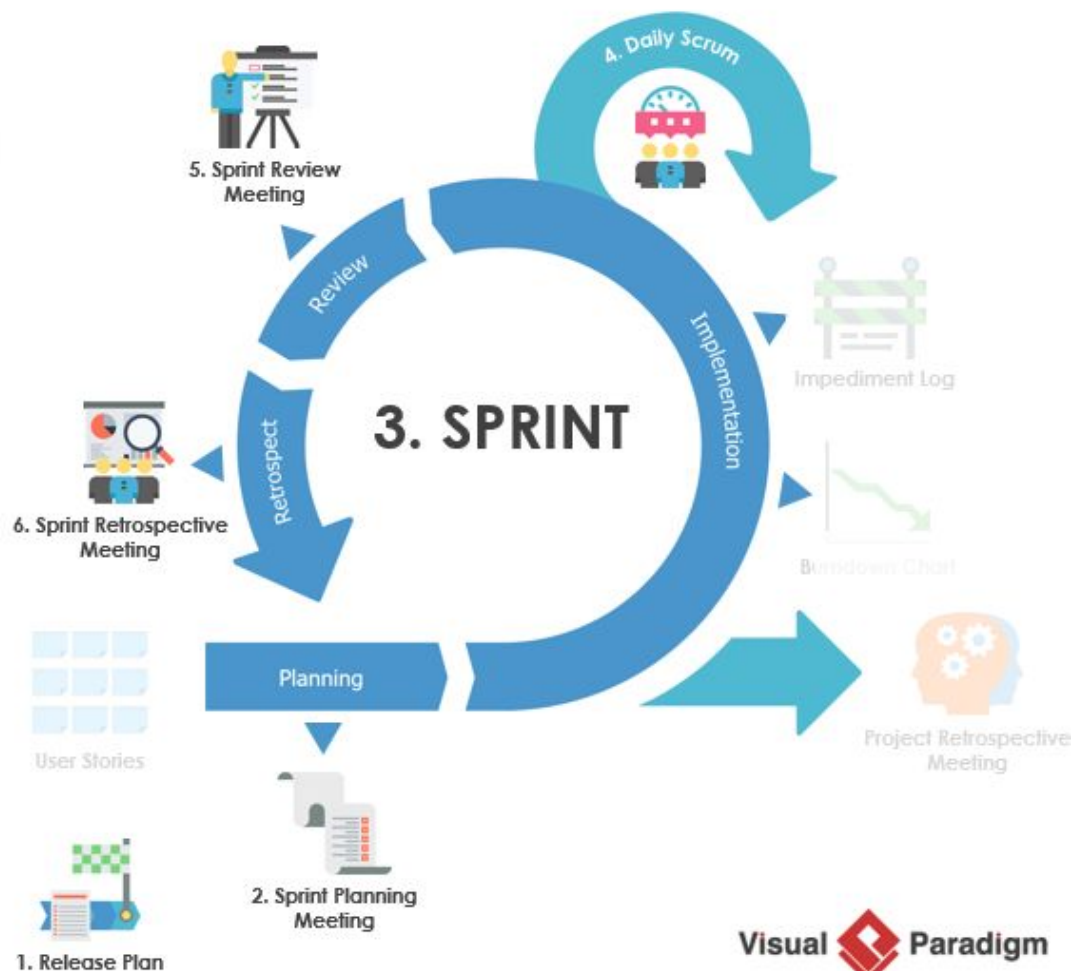
-  Release Planning Meeting
-  Sprint Planning Meeting
-  Sprint
-  Daily Scrum
-  Sprint Review
-  Sprint Retrospective

Artifacts

-  Product Backlog
-  Sprint Backlog
-  Sprint Burndown
-  Release Burndown

Rules

(bind the model together)



SCRUM események - Sprint

- Scrum lelke
- Általában 2-4 hét
- Tartalma: Sprint tervezés, napi scrum, fejlesztési munka, sprint áttekintés és sprint visszatekintés
- Sprint során:
 - Nem történhetnek olyan változások, amik veszélyeztetik a sprint célját
 - A minőségi célok nem csökkennek
 - A PO és a fejlesztő csapat újratárgyalhatja és tisztázhatja a feladatokat az időközben szerzett ismeretek alapján
- Minden sprint valamilyen cél elérését szolgálja
- A sprint részének tekintjük az elvégzett munkát és az eredményül kapott terméket is.
- A sprintet lefűjni kizárólag a PO-nak van jogosultsága az időkorlát lejárta előtt.

SCRUM események - Sprint tervezés

- A sprintben végzendő munka megtervezése.
- Ez a terv a teljes Scrum csapat munkájának közös eredménye.
- Scrum Master biztosítja, hogy az esemény megtörténjen és a résztvevők megértsék annak célját.
- A Sprint Tervezés a következő kérdésekre válaszol:
 - Mit fog tartalmazni a következő sprint eredményeképpen szállítandó inkrementum?
 - PO bemutatja a sprint során elérendő célt és azokat a Termék backlog tételeket, amelyek megvalósításával a Sprint eléri a Sprint célt.
 - Bemeneti elemei: Termék backlog, legutóbbi termék inkrementum, fejlesztői csapat tervezett kapacitása a sprint ideje alatt
 - Hogyan lehet elvégezni az inkrementum előállításához szükséges munkát?
 - A Fejlesztő csapat általában a feladatok meghatározásával és rendszer megtervezésével kezdi meg a munkát. Másokat is meghívhat a megbeszélésre, hogy technikai vagy szakterületi tanácsokat adjanak.
 - Sprint cél
 - Sprinthez rendelt célkitűzés, irányt ad a fejlesztőcsapatnak.

SCRUM események - Napi Scrum

- Daily Standup
- Maximum 15 perc időtartamú megbeszélés
- Fejlesztőcsapat összehangolja a tevékenységeket és megtervezi a következő 24 órát
- Minden nap ugyanabban az időben tartandó
- A megbeszélés során minden csapattag az alábbiakat fejti ki:
 - Mit sikerült elvégeznek tegnap, ami a Fejlesztő csapatot segítette a Sprint Cél elérésében?
 - Mit fogok tenni ma, ami a Fejlesztő csapatot segíti a Sprint Cél elérésében?
 - Van-e akadályozó tényező, ami gátol engem vagy a Fejlesztő csapatot a Sprint Cél elérésében?
- Scrum Master biztosítja, hogy a Fejlesztő csapat minden nap megtartsa a megbeszélést és ügyel arra, hogy kizárólag csak a Fejlesztő Csapat tagjai vegyenek részt a megbeszélésen.
- A napi scrum javítja a kommunikációt, szükségtelessé teszi egyéb megbeszéléseket, azonosítja és így elháríthatóvá teszi a fejlesztés útjába kerülő akadályokat.

SCRUM események - Sprint Áttekintés

- Sprint review
- Sprint végén tartják azzal a céllal, hogy ellenőrizzék az inkrementumot
- Scrum Csapat és Megrendelő oldal egyeztet, hogy mi történik a Sprint során
- Informális megbeszélés
- SM gondoskodik arról, hogy az esemény létrejöjjön és a résztvevők megértsék annak célját
- Tartalma:
 - PO megállapítja melyik backlog tétel lett “kész” és melyik nem
 - PO bemutatja a Backlog aktuális állapotát és előrevetíti a várható befejezés dátumát a haladás alapján
 - Az összes résztvevő közös munkával meghatározza, hogy mik legyenek a következő feladatok
—> értékes bemenet a Sprint Tervezéshez
- Eredménye egy módosított backlog, ami meghatározza a következő sprint tartalmát

SCRUM események - Sprint Visszatekintés

- Sprint Retrospective
- Lehetőség a Sprint Csapatnak, hogy elemezze saját tevékenységét
- Sprint áttekintés —> Sprint Visszatekintés —> Sprint Tervezés
- SM biztosítja az esemény létrejöttét és, hogy a résztvevők megértsék annak célját
- Célja:
 - Megvizsgálják, hogy mennyire volt sikeres a legutóbbi Sprint
 - Azonosítsák és sorba rendezzék a jól működő főbb elemeket és a lehetséges javításokat
 - Tervet készítsenek a Scrum Csapat működésének javítására
- A Scrum Csapat minden egyes visszatekintés során különféle tervet készít a termék minőségének javítására
- A Sprint Visszatekintés végére a Scrum Csapat-nak meg kell határoznia azokat a szükséges javításokat, amiket a következő Sprintben meg fog valósítani
- A Sprint Visszatekintés egy formális lehetőséget biztosít, hogy a csapat magára az ellenőrzésre és a korrekcióra összpontosítson

SCRUM a mindennapokban

- Kicsi, 3-9 emberből álló csapat
- Önszerveződő csapat
- Nem lépésről lépésre kidolgozott módszer
- Meghatározott időintervallumú termék/funkció/javítás létrehozás —> Folyamatosan tartható, reális határidők



KANBAN



KANBAN

A Kanban népszerű keretrendszer az agilis szoftverfejlesztés megvalósításához. A kapacitások valós idejű kommunikációját és a munka teljes átláthatóságát igényli. A munkadarabok vizuálisan ábrázolva vannak a kanban táblán, így a csapat tagjai bármikor megnézhetik minden munka állapotát.

Előnye a sw fejlesztésben:

A folyamatban lévő munka mennyiségét könnyen hozzá lehet igazítani a csapat kapacitásához

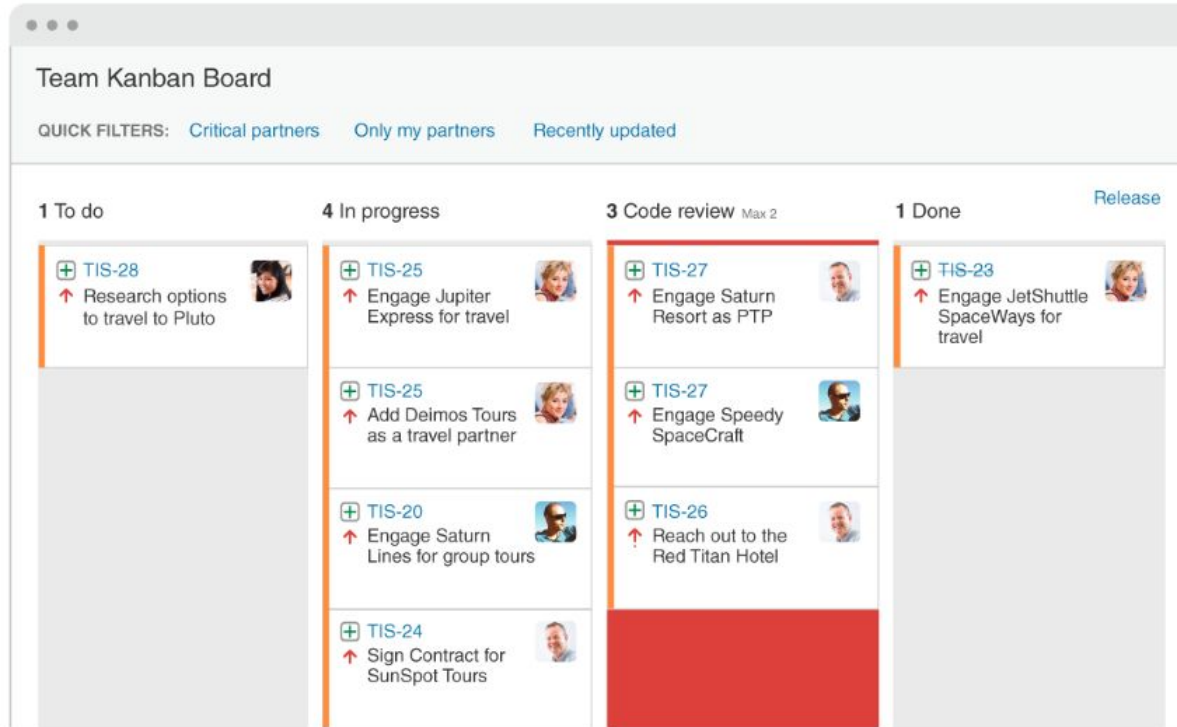
Ez rugalmasabb tervezési lehetőségeket, gyorsabb outputot, világosabb összpontosítást és átláthatóságot biztosít a csapatok számára a fejlesztési ciklus során.

Kanban táblák

Az összes kanban-csapat munkája egy kanban-tábla körül forog, egy eszköz, amely a munka megjelenítéséhez és a munka folyamatának optimalizálásához használható a csapaton belül. Míg a fizikai táblák népszerűek néhány csapat körében, a virtuális táblák kulcsfontosságú funkciók minden agilis szoftverfejlesztő eszközben nyomon követhetőségük, könnyebb együttműködésük és több helyről elérhető elérhetőségük érdekében.

Függetlenül attól, hogy a csapat táblája fizikai vagy digitális - feladata annak biztosítása, hogy a csapat munkája megjelenjen, a munkafolyamat egységes legyen, és minden blokkolót és függőséget azonnal azonosítsanak és megoldjanak. Az alapvető kanban tábla három lépésből (oszlopból) áll a munkafolyamat: teendő (to do), folyamatban (in progress) és kész (done). A csapat méretétől, felépítésétől és célkitűzéseitől függően azonban a munkafolyamat leképezhető, hogy megfeleljen az adott csapat egyedi folyamatának.

Kanban tábla



KANBAN kártyák

A munka kártyaként való bemutatásának fő célja a kanban táblán annak lehetővé tétele, hogy a csapat tagjai rendkívül vizuálisan követhessék a munka előrehaladását a kialakított munkafolyamaton keresztül.

A Kanban kártyák **kritikus információkat** tartalmaznak az adott munkaelemről, és teljes képet adnak a teljes csapat számára arról, hogy ki felelős az adott műért, az elvégzendő munka rövid ismertetése, a munka becsült időtartamára.

A virtuális kanban táblákon lévő kártyák képernyőképeket és egyéb műszaki részleteket (specs) tartalmaznak, amelyek értékesek a feladat végrehajtója számára.

Annak lehetővé tétele, hogy a csapat tagjai láthassák az egyes munkaelemek állapotát az adott időpontban, valamint az összes kapcsolódó részletet, fokozott összpontosítást, teljes nyomonkövethetőséget, valamint a blokkolók és függőségek gyors azonosítását biztosítják.

KANBAN tervezés flexibilitás

A kanban csapat csak az aktívan folyamatban lévő munkára összpontosít.

Miután a csapat elkészítette a munkaelemet (kártyát), áthelyezik a következő munkaelemet a backlog ("várólista") tetejéről.

A terméktulajdonos szabadon rendezheti a hátralévő munka prioritásait anélkül, hogy megzavarná a csapatot, mivel az aktuális kártyákon kívüli változások nem befolyásolják a csapatot. Mindaddig, amíg a terméktulajdonos a backlog tetején tartja a legfontosabb kártyákat, a fejlesztőcsoport garantáltan a fontos feladatokkal halad, így maximális értéket képviselnek a project számára. Tehát nincs szükség a scrumban található fix hosszúságú iterációkra.

SCRUM VS KANBAN

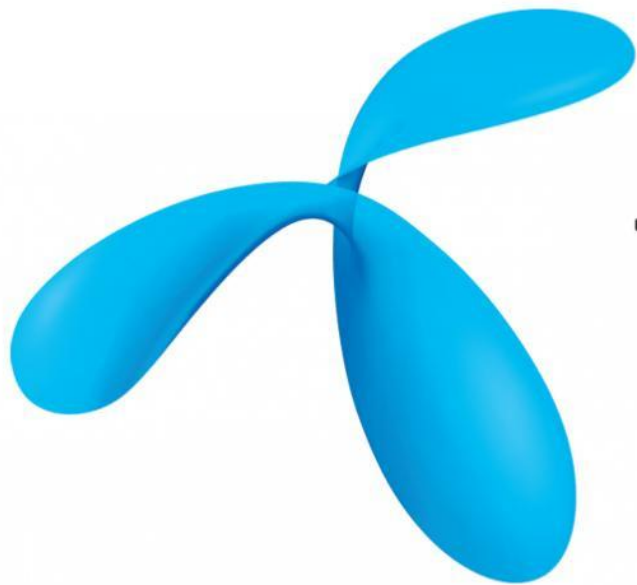
A Kanban és a scrum ugyanazon fogalmakkal rendelkezik, de nagyon eltérő megközelítéssel.

	SCRUM	KANBAN
munkaütem	Rendszeres rögzített hosszúságú sprint	folyamatos
"Release"	Akár minden sprint végén, ha a product owner jóváhagyja	Folyamatosan, a csapat döntése alapján
Szerepkörök	Product Owner Scrum master Dev team tester	Nincsenek kifejezett szerepkörök
Fő metrika	sebesség	Ciklusidő
Változásra reagálás	A csapatoknak arra kell törekedniük hogy a sprint során ne változtassanak a sprint elemein, különben a becslések, merikák és a kimenetel változhat	Bármikor történhet változás

SCRUM VS KANBAN

Egyes csapatok ötvözik a kanban eszményeit és a „scrumban” csapattá válnak.

Rögzített hosszúságú sprintjekeket és szerepeket vesznek a scrum-ból, a hangsúlyt pedig a folyamatban lévő munkára és a “ciklusban töltött időt” pedig a kanbanról.



telenor





Ma tegyük a holn:



Köszönöm a figyelmet!



Attrecto Zrt.
Attrecto Next Tech Digital Solutions

H-9024 Győr, Wesselényi str. 6.
info@attrecto.com

