<https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-HU.pdf>

<https://www.scrum.org/resources/scrum-guide>

Hivatalos Scrum útmutató:

Cím: Scrum Útmutató

Eredeti cím: The Scrum Guide

Kifejlesztette és karbantartja Ken Schwaber és Jeff Sutherland

Mi a scrum?

„A Scrum egy olyan keretrendszer, amelyet komplex termékek fejlesztésére és fenntartására hoztak létre.” 3. oldal - Mi a scrum rész

A 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 termékeket.” – 3. oldal –Scrum meghatározása rész

Sajátszavakkal való összefoglalás:

* Hatékonyság láthatóvá tétele
* könnyebb menedzsment
* folyamatellenőrzés
* viszonyok meghatározása munkafolyamatok, munkacsomagok, és a szerepkörök között

Leírás még miből és hány lépésből áll a Scrum:

„A Scrum egy iteratív (ismétlődő), inkrementális megközelítést alkalmaz a kiszámíthatóság optimalizálása és a kockázat kézben tartása érdekében. Az empirikus folyamatellenőrzés megvalósítása három pilléren nyugszik: transzparencia (átláthatóság), ellenőrzés és korrekció” -4.oldal teteje

Saját szóval: 4.oldal alapján

* Transzparencia (Átláthatóság):
  + közös megegyezés az úgynevezett doD-ben azaz ( definition of Done)-ban, hogy a fejlesztőcsapat, és a termék vevője mit tekint „Kész”-nek, itt közös konszenzusra kell jutni.
  + A fejlesztési folyamatnak átláthatónak, és visszakereshetőnek kell lennie a csapat részéről.
* Ellenőrzés:
  + A munkaanyagokat gyakorta ellenőrizni kell, de nem olyan gyakran, hogy ez gátolja, vagy akadályozza a fejlesztés menetét, és ezáltal a termék a szállítás idejére nem fog eleget tenni a doD-nek.
  + Ezeket az ellenőrzéseket érdemes a tervezés után a munka megkezdése előtt végrehajtani.
* Korrekció:
  + Ha az ellenőrzés rávilágít, hogy a végtermék nem felel meg a doD-nek, vagy egyéb szempontoknak, így a munkacsomag , vagy munkafolyamat újratervezése szükségessé válik, amit minél előbb végre kell hajtani.
  + Ez a folyamat újratervezést és újra priorizálást hoz magával.

Scrum 4 formális eseménye: - 4.oldal alja

* Sprint tervezés (Sprint planning)
* Napi Scrum ( Daily Scrum / Daily Stand up)
* Sprint Áttekintés ( Sprint Rewiev )
* Sprint visszatekintés ( Sprint Retrospective )

Csapat leírása: - 4.oldal vége 5. oldal eleje

„A Scrum Csapat A Scrum Csapat a Terméktulajdonosból, Fejlesztőcsapatból és Scrum Mesterből (Scrum Master) áll. A Scrum Csapatok önszerveződőek és kereszt-funkcionálisak. Az önszerveződő csapatok döntik el, hogy milyen módon tudják a legjobban elvégezni a munkát szemben azzal, hogy valaki kívülről irányítaná őket. A kereszt-funkcionális csapatok a munka elvégzéséhez minden szükséges kompetenciával rendelkeznek, és nem függnek olyanoktól, akik nem részei a csapatnak. A csapat modellt a Scrumban a rugalmasság, a kreativitás és a produktivitás optimalizálása érdekében tervezték meg. A Scrum Csapatok iteratív módon és fokozatos lépésekben (inkrementálisan) szállítják a terméket, maximalizálva a visszajelzés lehetőségét. A “Kész” termék fokozatos leszállításai biztosítják, hogy a termékből mindig elérhető egy jó eséllyel hasznosítható változat.”

Saját szavaimmal: de az 5. oldal alapján, viszont nem idézem onnan

* Terméktulajdonos – Product Owner
  + Ő felelős a fejlesztendő termék Backlog ( Termék Teendőlista ) –ért, ami tartalmazza a:
    - Vevői igényeknek megfelelő feladatok leírását
    - Ezeket a PO prioritás szerint sorba rendezi, hogy a legfontosabb, és a doD-nak legmegfelelőbb, valamit amik a legnagyobb értéket tudják teremteni.
    - Termék backlog „tisztán”, és frissen tartása, a fejlesztői és vevői igények figyelembevételével.
    - Ő felelős a termékért vagy szolgáltatásért, és annak az értékét képviselő minőségért.
    - Egyetlen személy, aki szakmai háttérrel rendelkezik.
    - A fejlesztőcsapat csak a Terméktulajdonostól kaphat utasítást
* Fejlesztőcsapat
  + Önszerveződő, önfenntartó „ökoszisztéma”
  + Technikai szakemberek csoportja, akiknek a célja a doD-nak megfelelő termék előállítása minden sprint végére, és annak való szállítása és prezentálása a vevő felé.
  + A fejlesztőcsapat egy önálló entitás, ami önszerveződő jellege miatt saját maga hozza meg a munkacsomagok alapján a munkafolyamatok szempontjából releváns döntéseket és inkrementumokat.
  + Kereszt funkcionálisak
  + Scrum esetén nem különböztetünk meg pozíciókat a fejlesztőcsapat tagja között, és nincs hierarchia se, és nincsenek kisebb dedikált csapatok a nagy csapaton belül, mint pl tesztelők, vagy integrátorok stb..
  + Kollektív felelősség terheli az egész csapatot függetlenül az egyes személyek tudásától, képzettségétől.
  + Fejlesztőcsapat mérete változó, de az opcionális csapatlétszám 3 és 9 között található, mivel 3 csapattag és az alatti létszám esetén nem elég nagy a csapat, hogy komplex rendszereket tudjon szállítani minden egyes sprint végén, ami kielégíti a vevői igényeket. 9 fő fölötti létszám esetén nehezen menedzselhetővé válik a csapat.
* Scrum Master
  + Feladata a Scrum fejlsztési módszertan szabályainak betartatása, és ezek alkalmazása az egész Scrum csapatra nézve.
  + A Scrum Master úgynevezett szolgáló-vezető szerepet tölt be, amit mind a külső felek, mind a belső csapattagok és a terméktulajdonos felé közvetít.
  + Feladata:
    - Módszerek kialakítása a Backlog átláthatóbbá és hatékonyabbá tételére.
    - Megszervezi és lebonyolítja a Scrum eseményeket.
    - Segíti a csapatot, és elhárítja a problémákat és akadályokat
    - Segíti a vevőket megérteni a scrum módszertanát, és hogy a vevő hogyan tud a módszertan alapján segíteni, a még jobb fejlesztés érdekében.

Események:

Minden esemény maximális hosszal rendelkezik, ezek betartatása a scrum master felelőssége. Ezen események általában fix időpontban és napon vannak minden sprintben, ezzel rendszerességet teremtve a csapat életében, és a nagyobb átláthatóság érdekében. Ezen eseményekre a munkacsomagok és folyamatok átláthatóbbá tétele és a csapat munkájának gördülékenyebbé és probléma mentessé tétele miatt van szükség. Ezen események elhagyása, hosszútávon problémát okozhat.

* Sprint
  + magában foglalja a Napi Scrumot, a Sprint tervezést, a Sprint áttekintést és a Sprint visszatekintést, valamint a fejlesztést.
  + Történhetnek változások, amik alapján újratervezés lehetséges, vagy a folyamatok újraszervezése. Ezen változtatások során a csapat es terméktulajdonos tisztázzák a felmerülő aggályokat.
  + Fontos, hogy a sprint célja és a doD nem változhat, hiszen minden ennek van alárendelve viszont ezek alapesetben rugalmas tervek és célok, amik hagynak mozgásteret a fejlesztőcsapat számára.
* Sprint tervezés ( Sprint Planning )
  + Teljes csapat közös munkája, scrum master, fejlesztőcsapat, terméktulajdonos. Ezt az eseményt a scrum master szervezi és irányítja, és felelős, hogy mindenki megértse a célt.
  + Két kérdés köré összpontosul a tervezés:
    - Mit leszünk képesek szállítani a következő sprint végére?
    - Hogyan fogjuk ezt elérni?
  + A fejlesztőcsapat a sprint meghatározott céljainak veti alá magát, a technológiát, és a funkcionalitást.
* Napi Scrum ( Daily Scrum / Daily Stand up )
  + Maximum 15 perc hosszúságú megbeszélés, minden nap ugyanabban az időpontban, ahol a csapat összehangolja a működését, megbeszélik a felmerült aggályokat, és problémákat, valamint megtervezik, és egyénekre osztják a következő napi munkafolyamatokat és munkacsomagokat.
  + Három kérdés köré szerveződik a napi scrum:
    - Mit ért el az adott fejlesztő az előző munkanap alatt, figyelembe véve a sprint célját?
    - Mi az, amit az adott fejlesztő a mai napon tenni fog, hogy megvalósuljon a sprint célja?
    - Vannak-e akadályozó tényezők, mik meggátolják, a sprint cél elérését?
* Sprint áttekintés ( Sprint rewiev )
  + Sprint végén, általában az utolsó nap végén tartják.
  + Cél, hogy erősítse a kapcsolatot a vevő és a scrum csapat között.
  + Scrum master szervezi és irányítja a meetinget.
  + Terméktulajdoson megállapítja az az elkészült munkacsomagokat az előzetes célok, és a doD alapján, hogy megfelelnek-e a kritériumoknak, és „kész” vagy „nem kész” státuszba kerüljenek.
  + Fejlesztőcsapat megvitatja, az aktuális sprintet, valamint válaszolnak a munkacsomagokhoz tartozó esetleges felmerülő kérdésekre.
* Backlog Refinement ( Backlog visszatekintés )
  + A scrum csapat megvizsgálja, hogy mennyire volt sikeres az adott sprint, és fejlesztési ciklus, különböző szempontok alapján. ( kapcsolatok, hatékonyság, munkamorál, motiváltság, érthetőség stb.. )
  + Ismerjék fel, és találják meg, majd rendszerezzék a csapat által beérkező véleményeket, amik mind pozitívak és negatívak is lehetnek. Ezek a visszajelzések főleg a terméktulajdonos felé irányulnak.
  + Az előbb felsoroltak alapján a csapat közösen előáll egy tervvel a problémák kiküszöbölésére, és a hatékonyság növelésére.

<https://agilemanifesto.org/iso/hu/manifesto.html>

**Kiáltvány az agilis szoftverfejlesztésért**

A szoftverfejlesztés hatékonyabb módját tárjuk fel saját tevékenységünk és a másoknak nyújtott segítség útján. E munka eredményeképpen megtanultuk értékelni:

1. Az egyéneket és a személyes kommunikációt a módszertanokkal és eszközökkel szemben
2. A működő szoftvert az átfogó dokumentációval szemben
3. A megrendelővel történő együttműködést a szerződéses egyeztetéssel szemben
4. A változás iránti készséget a tervek szolgai követésével szemben

Azaz, annak ellenére, hogy a jobb oldalon szereplő tételek is értékkel bírnak, mi többre tartjuk a bal oldalon feltüntetetteket.

<https://agilemanifesto.org/iso/hu/principles.html>

Agilis szoftverfejlesztés 12 alapelve:

1. Legfontosabbnak azt tartjuk, hogy az ügyfél elégedettségét a működő szoftver mielőbbi és folyamatos szállításával vívjuk ki.
2. Elfogadjuk, hogy a követelmények változhatnak akár a fejlesztés vége felé is. Az agilis eljárások a változásból versenyelőnyt kovácsolnak az ügyfél számára.
3. Szállíts működő szoftvert gyakran, azaz néhány hetenként vagy havonként, lehetőség szerint a gyakoribb szállítást választva.
4. Az üzleti szakértők és a szoftverfejlesztők dolgozzanak együtt minden nap, a projekt teljes időtartamában.
5. Építsd a projektet sikerorientált egyénekre. Biztosítsd számukra a szükséges környezetet és támogatást, és bízz meg bennük, hogy elvégzik a munkát.
6. A leghatásosabb és leghatékonyabb módszer az információ átadásának a fejlesztési csapaton belül, a személyes beszélgetés.
7. A működő szoftver az elsődleges mércéje az előrehaladásnak.
8. Az agilis eljárások a fenntartható fejlesztést pártolják. Fontos, hogy a szponzorok, a fejlesztők és a felhasználók folytonosan képesek legyenek tartani egy állandó ütemet.
9. A műszaki kiválóság és a jó terv folyamatos szem előtt tartása fokozza az agilitást.
10. Elengedhetetlen az egyszerűség, azaz az elvégezetlen munkamennyiség maximalizálásának művészete.
11. A legjobb architektúrák, követelmények és rendszertervek az önszerveződő csapatoktól származnak.
12. A csapat rendszeresen mérlegeli, hogy miképpen lehet emelni a hatékonyságot, és ehhez hangolja és igazítja az működését.