**Scrum**

Agilis szoftverfejlesztés összetevője.

Irányításra alkalmas szoftver?

Nem kizárólag termékek létrehozására alkalmas keretrendszer.

-o-

**Kialakulása**

Új termékek fejlesztésének felgyorsítása, rugalmasabbá tétele.

Vízesés modell váltófutáshoz hasonlítása

* Egyszerre csak egy ember fut és a futók egymásnak adják a stafétát
  + T1 -> T2 -> T3 -> T4

-o-

**Jellemzői**

A Scrum keretrendszer magában foglal tevékenységeket és szerepeket.

Szerepek:

* Scrum master
  + Folyamat felügyelete
  + Önálló munkavégzést coachként segíti
* Product Owner
  + Projektben érdekelt döntéshozókat képviseli
* Csapat
  + 3-9 főből áll
  + Lefedik az összes munkafolyamatot

Sprint

* Output: Működő szoftver
* Product Backlog –kikerül belőle> Megvalósítandó funkció
  + Másnéven a product backlog termék teendőlistája
  + Az elvégzendő munka magas szintű követelményeiből álló, fontossági sorrendbe állított lista
  + Hogy a Sprint során a lista melyik elemei kerülnek megvalósításra, azt a sprint elején tartott "sprinttervező" megbeszélés során választják ki.
* Sprinttervező megbeszélés
  + Product Owner –Csapattal közli> A teendők listájának fontossági sorrendjét
    - Magától értetődő, hogy a fontossági sorrend meghatározza a folyamatok elvégzésének sorrendjét
  + Csapat –eldönti> Következő Sprintbe tartozó Sprint teendőlista
    - A Csapat ígéretet tesz a folyamatok listájának elvégzésére
    - A Sprint folyamán a "Sprint teendőlistáját"nem lehet megváltoztatni, a sprint során elvégzett tevékenységek rögzítettek.
* Sprint elkészült
  + A csapat bemutatja az elkészült funkciókat

-o-

Fontos alapelve a Scrumnak, hogy felismeri és elfogadja, hogy

* A megrendelő a fejlesztés során meggondolhatja magát a követelményekkel kapcsolatban
* Váratlan változások nem kezelhetők könnyen a hagyományos, előzetes tervezési fázison alapuló módszerekkel
* Nincs lehetőség a probléma teljes megértésére és definiálására.
  + Ehelyett célja: Funkciók gyors megvalósítása
  + Gyors reakció a változó körülményekre

**Szerepek**

**Scrum Csapat**

* Pontos részei: Terméktulajdonos, Fejlesztők, Scrum Master
* Együttesen felelősek a tagok a létrehozott eredményekért

**Terméktulajdonos**

* A megrendelőt képviseli.
* Fontossági sorrendet meghatároz a feladatok elvégzésére vonatkozóan.
* A termék teendőlistát bővíti a megrendelő szempontjából megfogalmazott igényekkel
  + User story eleme/tétele a termék teendőlistának
  + a teendőlista egy eleme meghatározza, hogy (Ki, mit, miért szeretne)
* Felelős a termék értékének maximalizálásáért.
* Az egyedüli személy, aki felelős a Termék Backlog kezeléséért
* A Termék Backlog kezelése a következő folyamatokat foglalja magában:
  1. A Termék Backlog tételeinek egyértelmű leírása
  2. A Termék Backlogban szereplő tételek fontossági sorrendjének meghatározása
  3. A fejlesztők által végzett munka értékének optimalizálása
  4. Biztosítása a feltételnek, hogy a Termék Backlog elérhető, könnyen áttekinthető és mindenki számára világos legyen.
  5. A Scrum Csapat következő munkájának egyértelműsítése
  6. Biztosítása, hogy a fejlesztők legalább a munkavégzéshez szükséges szinten értsék a Termék Backlog egyes tételeit.
* A Terméktulajdonos nem egy bizottság, hanem egyetlen személy.
  + A terméktulajdonos képviselheti egy bizottság kívánságait a Termék Backlogban, de ha a bizottság meg szeretné változtatni valamelyik Termék Backlog elem prioritását, akkor ezt csak a Terméktulajdonoson keresztül teheti meg.
    - Egy bizottság kérdés megvitatására, döntés hozatalára, ügy előkészítésére, megvizsgálására, felelős irányítására, elintézésére vagy ellenőrzésére alakult választott, kiküldött csoport.
* A teljes szervezetnek tiszteletben kell tartania a Terméktulajdonos döntéseit.

**Fejlesztők**

* Felelősségük, hogy a termék elkészüljön
* Azon dolgoznak, hogy minden egyes Sprint végén leszállítható legyen a termék egy "Kész" potenciálisan kibocsátható növekménye.
  + Leszállítás és terméknövekmény jelentése? Milyen tulajdonságai vannak egy terméknövekménynek?
    - Létezik olyan sorozat, aminek az elemei terméknövekmények
* A fejlesztők maguk szervezik és menedzselik a saját munkájukat
* A Termék Backlog elemeit képezik terméknövekményre
* Attribútumai egy fejlesztőnek
  + Önszerveződő
    - Nem mondják meg nekik, hogy miként hozzanak létre a Termék Backlogból potenciálisan szállítható funkcionalitást tartalmazó növekményeket
  + Felkészült
    - Minden olyan készséggel rendelkeznek, ami szükséges a termék növekmények elkészítéséhez.
  + Nem rendelkeznek titulussal
  + Nincsenek alcsoportok célfeladatok elvégzésére
  + A fejlesztők körében az egyes tagok speciális ismeretekkel, készségekkel és szakterületi tudással rendelkezhetnek, de a felelősség a teljes Scrum csapatra, mint egy egységre hárul

**Scrum Mester**

* Scrum népszerűsítéséért és támogatásáért felelős, a Scrum Útmutatóban foglaltaknak megfelelően.
* Mindenkinek segítenek megérteni a Scrum elméletét, gyakorlati elemeit, szabályait és értékeit.
* Segíti a Scrum Csapaton kívülieknek megérteni azt, hogy mely Scrum Csapattal való interakciójuk lesz hasznos és melyik nem.
  + A Scrum Mester mindenkinek segít olyan módon megváltoztatni ezeket az interakciókat azért, hogy azok a Scrum Csapat által létrehozott értéket maximalizálják.
* Támogató, vezető szerepkör
  + Csapat teljesítményének növelése
  + Munkát akadályozó tényezők elhárítása
    - Munkát akadályozó tényező az, ami gátolja a Csapatot abban, hogy a Sprint célját megvalósítsa.
  + A Scrum mester nem A Csapat vezetője, hanem A Csapat és a külső tényezők közötti szereplő.
  + Ügyel a Scrum folyamat megfelelő alkalmazására.
  + Betartja a Scrum szabályait.
  + Védi a csapatot és biztosítja, hogy a csapat az elvégzendő feladatokra koncentráljon.
  + Mozgásban tartja a csapatot
* A nyolc szerepkör

1. Támogató vezető
   1. A csapat tagjaira és az általuk megteremtett ügyfél-érték megteremtésre összpontosít annak érdekében, hogy eredményeket érjenek el a szervezet értékeivel, alapelvekkel és üzleti célokkal összangban.
2. Facilitátor
   1. Keretek tiszta meghatározásával segíti az együttműködést.
   2. Megbeszélések:
      1. Adminisztrációja
      2. Moderálása
      3. Dokumentálása
3. Coach
   1. Az egyéneknek segít fejlődni a gondolkodásmódra és a viselkedésre összpontosítva.
   2. A csapatot a folyamatos fejlődésben, míg a szervezetet a Scrum csapattal való valódi együttműködésben támogatja.
4. Menedzser
   1. Akadályok kezelése
   2. Pazarló folyamatok csökkentése
   3. Csapat egészségének kezelése
   4. Önszerveződés határainak kezelése
   5. Kultúra támogatása.
   6. Támogatás megrendelők számára a megfelelő riportok és eszkalációs anyagok összeállításával -> (Vízesés modell, dokumentáció)
   7. Sprint állapotát követi
      1. Veszélyt észreveszi
      2. Csapattal közösen kitalálja, hogy megoldható a team-en belül a probléma, vagy eszkalálni kell.
      3. Utóbbi esetben megoldási javaslattal együtt eszkalál.
5. Mentor
   1. Agilis tudást és tapasztalatot átadja a csapatnak
   2. Segíti testreszabni a Scrum keretrendszert
      1. Nem hagyja el a lényeges elemeket
      2. Nem megy szembe a lényeges elemekkel
6. Tanár
   1. Biztosítja a Scrum és más vonatkozó módszerek megértését és bevezetését
      1. M.: Egy tanárra szükség van ☺
7. Akadálymentesítő
8. Változáskezelő

Más érintettek:

Nem részeik a Scrum folyamatnak, de figyelembe kell venni őket -> Paraméterként szerepelhetnek a Scrum folyamatban.

Felhasználótól érkező visszajelzésekkel figyelembe veszik a sprintek tervezését.

**Felhasználók**

* Akik a szoftvert használni fogják
* A felhasználó hangját megtestesítő személy

**Üzleti szereplők**

* Azok az emberek, akik lehetővé teszik a projekt létrehozását és akiknek a termék hasznot fog hozni.
* Közvetlenül csak a sprintáttekintő megbeszélésen (demo) vesznek részt a folyamatban.
* Ez az adminisztrátorokat és az igazgatókat is magába foglalja.

**Menedzserek**

* A fejlesztésben részt vevő szervezeti egységek munkakörnyezetét teremtik meg.
* Nem csak a funkcionális vezetőket jelenti.

Kérdések:

* Egy szereplőnek milyen interfész jut?

**Megbeszélések**

**Sprint Tervezés(Sprint Planning)**

* Minden Sprint előtt:
  + Elvégzendő feladatok kijelölése a termék teendőlistájáról a terméktulajdonos közreműködésével
  + Sprint teendőlistájának előkészítése, amely a csapat figyelembevételével részletezi az egyes részfeladatok időszükségleteit.
  + Várhatóan elvégezhető feladatok meghatározása és kommunikálása az aktuális sprint során
  + 4 hetes sprint esetén maximum 8 óra hosszúságú, rövidebb sprint esetén ez az esemény is rövidebb.

**Napi Scrum-megbeszélés**

* Időpontra nincs szabály
* Napindító megbeszélés tartható
* Státuszmegbeszélésre népszerű időpont az ebéd utáni időszak.
  + Ebéd előtt az ember dolgozott már, így az ember nem a privát dolgain gondolkozik éppen, hanem munkáján.
* Időpont kiválasztását befolyásoló paraméterek:
  + None
* Szabályok:
  + Főszabály, hogy csak a Scrum csapat tagjai beszélhetnek
  + 15 perces a megbeszélés ideje
  + A résztvevők állnak a megbeszélés során
  + Minden nap ugyanazon a helyen és ugyanabban az időpontban tartják
  + Minden résztvevő ehhez hasonló kérdéseket válaszol meg:
    - Mi az, amit a tegnapi megbeszélés óta csináltam? (vagy mit tanultam?)
    - Mi az, amit a mai nap tervezek csinálni?
    - Vannak-e akadályok, amik gátolnak a saját és a sprint célok elérésében?
* Az akadályok elhárításában a Scrum Mester segít a csapatnak.
* A sprint végén két megbeszélést tartanak:

1. Sprint Áttekintés
   1. Elkészült és nem elkészült munkák áttekintése
   2. Az elkészült munka bemutatása a terméktulajdonos és a fejlesztésben érdekelt részére
   3. A legkisebb, értéket adó, működő terméket már be lehet mutatni. A még nem működő elemeket nem lehet bemutatni.
   4. 4 hetes sprint esetén maximum 4 óra hosszúságú, rövidebb sprint esetén ez az esemény is rövidebb.
2. Sprint Visszatekintés
   1. A csapattagok véleményt alkotnak az elmúlt sprintről. A vélemény lehet egy puszta benyomás is, nem kell kidolgozott, szilárd álláspontnak lennie
   2. Javaslatokat tesznek a folyamatok továbbfejlesztésére. A javaslatoknak nem kell kiérleltnek lenniük, a kidolgozás nem a visszatekintés része.
   3. Két kérdés:
      1. Mi az, ami jól ment a sprint alatt?
      2. Mi az, amit a következő sprint során jobban lehetne csinálni?
   4. 4 hetes sprint esetén maximum 3 óra hosszúságú, rövidebb sprint esetén ez az esemény is rövidebb.

* Note: A Coach vagy a menedzser segíthet a folyamatok futásidejének optimalizálásában.

**Scrum munkaanyagok (Scrum artifacts)**

Az agilis fejlesztés a működő szoftvert előnyben részesíti az átfogó dokumentációval szemben.

Három munkaanyag meglétét írja elő a Scrum 3:

**Termék követelménylistája**

* Az egész termékre vonatkozó magas szintű követelményleírások.
* Elemek lehetnek:
  + Funkciók leírásai
  + Kívánságok
  + Ötletek
* A követelménylista elemei rendezettek
* A fejlesztés célját tartalmazza
* Szabadon szerkeszthető bárki által
* Becsléseket tartalmaz a bejegyzések üzleti értékére és ráfordításigényére vonatkozóan.
* A becslések abban segítik a terméktulajdonost, hogy meghatározza a bejegyzések sorrendjét, és bizonyos mértékig a prioritásukat
  + Becslések eredményei bejegyzések sorrendje és bizonyos mértékben prioritása
    - Mi a különbség a prioritás és a bejegyzés sorrendje között?
    - Ha például a "helyesírás-ellenőrzés" és "repülőgép-szimulátor" funkcióknak azonos az üzleti értékük, akkor a kevesebb fejlesztési ráfordítást igénylő funkció fog magasabb prioritást kapni
      * A jobb megtérülésű funkció kap magasabb prioritást azonos üzleti értékkel rendelkező funkciók közül.

**Sprint teendőlistája**

* Dokumentum, mely tartalmazza, hogy a csapat hogyan fogja elkészíteni a sprint során megvalósítandó funkciókat.
* Egyes funkciókat részfeladatokra bont.
  + A felbontást célszerű úgy elvégezni, hogy egy részfeladat 4-16 óráig tartson.
  + Ilyen részletezettség mellett a csapat összes tagja pontosan érti, hogy:
    - Mit kell elvégezni
    - Mindenki kiválasztja a neki legmegfelelőbb részfeladatot.
* A sprint teendőlistájában lévő részfeladatokat nem rendelik személyhez
  + Ehelyett a csapattagok választják ki azokat a meghatározott prioritások, szükségletek és a csapattag képességeinek megfelelően.
* A csapat kezelése alatt áll.

**Növekmény**

* A Sprinben leszállított Termék Backlog elemeknek és az összes megelőző Sprint során szállított növekmények értékének összessége.
* A Sprint végére az új növekménynek:
  + "Kész"-nek, azaz használhatónak kell lennie
* Meg kell felelnie a Scrum Csapat által meghatározott "Kész" definíciójának.
* A növekmény egy ellenőrizhető, "Kész" munkatermék, ami a Sprint végén az empirikus működést támogatja.
* A növekmény egy lépés egy vízió vagy cél felé.
* Felhasználható állapotban kell lennie független attól, hogy a Terméktulajdonos úgy dönt, hogy ténylegesen kibocsátja-e azt.

Kérdések:

* A feltételek a fejlesztésre/projektre nézve megnyilvánulnak-e egy dokumentációban, vagy a Backlog kezeléséért felelős Terméktulajdonos feladata, hogy olyan tételek kerüljenek be a backlogba, melyekért növekményekre történő leképezésért jó minőségű munkaterméket kapunk?

**Haladás követése a Scrumban**

Hasznos információt tartalmaznak az alábbi eszközök a folyamatokat érintő döntések meghozatalához.

**Burn down chart (egyfajta napi eredménykimutatás)**

* Grafikon
  + Mutatja a sprint teendőinek a listájából hátralevő munka mennyiségét
  + Minden nap frissítik
  + Egyszerű módon jeleníti meg a sprint állapotát

**Burn up chart**

* Grafikon
  + A Sprint feladatlistáján szereplő feladatok számát és az elvégzett feladatok számát ábrázolja
  + Naponta frissül
  + A Burn Down chart dekompozíciója.

**Sebesség**

* Csapat termék-teendőlista -> R, x-> f(x), ahol f az x elem kivitelezésének várható nehézségének függvénye, x eleme a Csapat termék-teendőlistának
  + A becslésre több kidolgozott módszer is rendelkezésre áll
* A sprint során elkészített elemek összpontszáma a csapat sebessége.
* Új csapatok esetén a sebesség jellemzően sprintről-sprintre nő, tapasztalt csapatok esetén közel állandó.
* A sebesség segítségével jól becsülhető, hogy az adott pillanatban ismert termék-teendőlista mikor fogy el.

-o-

**Adaptív projektmenedzsment**

* A megrendelő a fejlesztő csapat részévé válik
* Gyakoriak a köztes szállítások működő funkcionalitással
* A fejlesztés növekményes
* A köztes állapotokat is validálják és ellenőrzik, hogy ne csak a végén derüljenek ki a problémák, legyen idő kijavítani őket.
* Gyakori kockázatelemzés a fejlesztőcsapat részéről.
* Napi megbeszélés, ahol mindenkit megkérdeznek, hogy:
  + Mit csinált tegnap óta
  + Mit tervez holnapra
  + Vannak-e problémái, amik meggátolják a célja elérésében
* Átlátható tervezés és modularizáció, azaz lássa mindenki, hogy ki miért felel és milyen határidővel
* Gyakori találkozók, amelyeken figyelemmel kísérik a haladást.
* Semmilyen problémát nem söpörnek a szőnyeg alá. Mindenkit meghallgatnak, aki felismer és ismertet egy váratlan problémát.
* A munkahely és munkaidő legyen hatékony. A több munkaóra nem feltétlenül vezet több eredményre

Kérdések:

* Mi számít projektdefiníciónak egy Scrum keretrendszeren belül?
* Hogyan működik közösen a Scrum keretrendszer a vízesés modellel?
* Mi a kapcsolat a fejlesztési modell, keretrendszer és interfész között?

**Solo Scrum**

* Kis csapat.
* Csapattagok kommunikációját próbálja javítani.

**Scrum bullying**

* A zaklatók az agilis keretrendszert előíró dogmaként értelmezik.

**Az agilis coach 12 pontja**

A scrum bullying visszaszorításának érdekében

1. Elégítsük ki ügyfeleink, valamint az általunk támogatott szervezet agilitással kapcsolatos igényeit a napi gyakorlatban történi példamutatással a mikéntekre
2. Kis lépések hozzávezetnek a folyamatos értékteremtéshez
3. Közös munka a támogatott csapattal

…

-o-

**Kifejezések**

Story – A termék funkcióinak magas szintű, megrendelőközpontú leírása

* Eszköz a Story gyártásához?

Product backlog

* A projekt során megvalósításra váró teendők listája, fontossági sorrendben

Sprint backlog

* Konkrét feladatok a következő sprintre

Backlog item

* Teendő

Sprint

* A sprinttervezés során kiválasztott teendők megvalósítására szánt rövid iteráció

Daily standup

* Rövid találkozó, ahol megbeszélik az eredményeket, az akadályokat és a következő teendőket

Sprint planning session

* Megbeszélés, amelyen a következő sprint teendőit definiálják

Sprint retrospective

* Visszatekintés
* Goal:
  + Fejlesztési folyamat gyengeségeinek elhárítása
  + A hatékonyság javítása
  + Minden csapattag elmondja a véleményét az utolsó sprinttel kapcsolatban
    - A csapat ezután megegyezik, hogy mit változtatnak a fejlesztési folyamaton a következő sprint során (?)

Sprint burn down chart

* Kimutatás a napi eredményekről a sprint során

-o-

Milyen kapcsolatban van a Jira a Scrum keretrendszerrel?

**Összefoglaló a Scrum-ról**

Egy termékek létrehozására alkalmas keretrendszer.

Magában foglal tevékenységeket és szerepeket

**Szerepek**

* Projektmenedzser
* Scrum Master
* Fejlesztők

**Tevékenységek**

* Megbeszélések
* Sprint

**Munkaanyagok**:

* Termék követelménylistája
* Sprint Teendőlistája
* Növekmény

**Haladás követése**:

* Burn down chart
* Burn up chart
* Velocity

**Adaptív projektmenedzsment**

Fejlesztés során előbukkanó problémák előrejelzése, azokra megoldások találása.

-o-

Jira

**Scrum board**

A Jira Scrum board is the single source through which all the work of a team can be accessed by a team member at any time.

Scrum is the framework and Scrum board is the visual display of the progress during the development cycle.

* Operating Sprint commitments
* Visualizing a Burn –Down chart
  + Burning down stories
  + What are burnable objects?
* Definitions
  + 1
    - Name
    - Projects
  + 2
    - Name
    - Jira Filter
      * Shares
      * Owner

Questions:

* What is a Jira Filter?
  + A filter contains objects named Shares
    - Key, Summary, Status, Epic Link, Components, Labels, Sprint, Created, Environment, Affects

**CONFIGURATION**

Columns

* Columns contain items
* An item is an issue
* Sprints contain columns
* Items move between columns

Swimlanes

* Query
  + Name
  + JQL
  + Description
* Stories

Quick Filters

* Name
* JQL
* Description

…

Question:\_1

* What is a JQL?

Question\_2:

* What is an issue?

Answer\_2:

An issue is a building block of any Jira project.

An issue could represent a story, a bug, a task or another issue type in your project.

Selected issue

* Select an issue to view its details

Backlog

* Estimate issues and plan your sprints

Issue actions

* Move
* Clone
* Log
* Other

Issue details

* Description
* Comment
* Content

Project and Issue keys

* Id for every piece of work tracked with Jira
* Issue keys are
  + Labels on issues themeselves
  + In search results and saved filters
  + On cards on boards in a project's backlog
  + In links connecting pieces of work
  + In the issue's URL
  + Anywhere to reference the work you're tracking
* Parts of issue keys:
  + Project key
    - String
    - Describes what pieces of work are related to project
    - a piece of work is a \_ relared to an issue
  + A sequential number

Answer\_1:

JQL stands for Jira Query Language

* It is a way to search for issues

JQL is used to form a Query

Queries are series of simple elements strung together to form a more complex question.

A query has three basic parts

* Fields
  + Different types of information in the system
    - Example.: Prioirty, fixVersion, issue type
* Operators
  + Operators relate the field to the value
  + Examples: Equals, not equals, less than, IN, etc.
  + Operators are relations between Fields and Values
* Values
  + Data in the actual query
  + Usually the item for which we are looking
* Keyword
  + Specific words in the language with special meaning
    - Example.: AND, OR

FIELD – OPERATOR – VALUE + KEYWORD

-o-

**CARDS**

Card colours can be used for differentiation.

* Questions:
* How can card colours differentiate?
* What is a card?

Cards have a layout