

Scrum

hogyan növeljük fejlesztőink
hatékonyságát 1000%-kal?

Mi az a Scrum?

- a *Scrum* egy módszertan, amivel nem létező problémákat lehet megoldani nagyon gyorsan
- a *Scrum* az, amit *Scrum*nak hívnak
- *Scrum* bármi lehet, de ha valamit nem akarunk *Scrum*nak hívni, akkor hívjuk *Kanban*-nak

Milyen a jó Scrum?

a jó *Scrum* ezen elemek bármelyikét tartalmazza; minél több elemet tartalmaz a felsoroltak közül, annál jobb:

- *standup*
- *estimation*
- *complexity*
- *backlog*
- *sprint*
- *demó*
- *retrospektív*

a Scrum legnépszerűbb elemei

a felsorolt elemekből bármi szabadon kiválasztható, ezeket az alábbi módon valósíthatjuk meg:

Standup

- A *standup* a nap bármelyik részén tartható meeting, amin részt vesz a csapat
- Minél kevesebben vannak a meetingen, annál tovább tartson (átlagban 5 fő = fél óra, de minél hosszabb, annál jobb)
- A *standup*on mindenki elsorolhatja napi sérelmeit, panaszkodhat, beszélhet bármiről, amiről szeretne (amíg a többiek el nem unják és azt nem mondják, hogy "*ez nem standup*")

- Egy csapat *standup*ján részt vehetnek más csapattagok is, ők is panaszkodhatnak, mesélhetnek, beszélhetnek bármiről
- A *standup* néha átalakulhat *esztimálássá*, vagy *demózássá*, vagy *retrospektívvé*, vagy akármilyen más meetinggé

Estimation

- Az *estimation* az a meeting, amikor a csapat a jövőben esetlegesen várható, ködös feladatokról beszélget
- Az *estimation*-ön mindenki elmondhatja véleményét a jövőbeli feladatról, bármilyen szakterülethez hozzászólhat, bármilyen megrendelő, vagy architect által felvetett ötletéről elmagyarázhatja, hogy az ostobaság
- Egy sikeres beszélgetés végén a résztvevők számokat mondanak, amit aztán többszöri nekifutásra a legkisebb kikiáltott számra alakítanak

Complexity

- Az *esztimáción* adott szám a *complexity*
- A *complexity* bármilyen szám lehet, de általában 1 és 13 közötti egész szám (leggyakrabban azonban 13)
- A *complexity* jelenthet órát, napot, hetet; a szám bármilyen időegységgel felszorozható és egy adott feladat bonyolultságát is tükrözheti
- Érdeemes minél nagyobb számokat mondani, hogy aztán a fejlesztés alatt lehessen panaszkodni, hogy nincs elég munka (ez kiváló témát ad a *standupra*)

- Minden fejlesztési kör (*sprint*) alatt az összeadott *komplexitások* értéke változó, átlagos eltérésük +/-40
- Bármely csapat által adott *komplexitás* áttehető egy másik csapathoz
- Ha egy csapat nem ér rá, akkor csoportvezető, architect, menedzser, esetleg recepciós lány is adhat *komplexitást*

Backlog

- A *backlog* egy elképzelt (virtuális) tábla, amin az lenne látható, hogy ki mivel foglalkozik (ezeket a dolgokat *story*-knak is hívhatjuk)
- Egy *backlog* implementálható excelben, jirában, txt fájlban, vagy bármilyen digitális formátumban, a lényeg, hogy minél átláthatatlanabb legyen
- A *backlog* tartalmazza a feladatokat, amikre a csapat megkísérelt *komplexitások*at mondani, de egy sprint alatt bármi elhelyezhető rajta (ilyenkor azonban érdemes *Kanban*-ról beszélni)

- fontos, hogy a virtuális *backlog* kezeléséhez csak egy embernek legyen joga, vagy lehetőleg még neki se
- egy virtuális *backlog* akárhány elképzelt státuszt tartalmazhat. A legnépszerűbbek: waiting for testing, waiting for deployment, unprioritized, unassigned, tested on dev, tested on live, deployed to dev, deployed to beta stb.
- Ha egy elem elkészült, akkor a státusza változzon meg valami másra, mint ami volt

Sprint

- A *sprint* a fejlesztési fázis (fontos, hogy Scrumban lehetőleg ne használjunk magyar nyelvű kifejezéseket!)
- Egy *sprint* lehet pár nap, vagy pár hét, de általában a vége és az eleje egy bizonyos napon szokott lenni, kivéve ha ez valamiért kényelmetlen
- A *sprint* ún. *sprint planning*-gel kezdődik és demózással is véget érhet

- A *sprint* során az elvállalt (tehát menedzserment által halaszthatatlannak ítélt) *story*kon szokott dolgozni a csapat
- Egy *sprint*be bármilyen új feladat (*task*, *story*) bevehető, ha az fontosabbá válik az aktuális *story*knál
- Ha nagyon sok ilyen feladat érkezik, akkor azt *swat*-olásnak is hívhatjuk
- Ha valamiért a csapat mégsem akar megcsinálni egy *story*-t, akkor azt meg kell *invesztigálni*

Demó

- Általában a *sprint* végén a csapat *demózik*: ilyenkor félkész *story*-kat próbál egy nehezen kezelhető kivetítőn bemutatni
- A menedzser, vagy csoportvezető (akit általában udvariasságból *product owner*-nek, röviden *po*-nak hívunk) ezeket a félkész *story*kat elfogadja
- Ha egy *story* nem készül el több egymást követő *sprint*ben sem, akkor azt a *po* nem fogadja el és ismét beveteti a következő *sprint*be

Retrospektív

- A *retrospektív* egy olyan standup, ahol nem kell elmondani, hogy ki min dolgozott, vagy min fog dolgozni
- A *retrospektíven* általában többet szokás panaszkodni, mint a *standupon*
- A panaszokra udvariasságból javaslatokat kell adni, ezek az *action item*-ek (az *action itemek* arra valók, hogy egy fájlba lementsük őket, majd a fájlt letöröljük)
- Ha egy csapat túl sokat panaszkodik, akkor büntetésből azt is el kell mondaniuk, hogy mi volt jó az adott *sprint*ben

Honnan lehet tudni, hogy a Scrum bevezetése sikeres volt?

- a fejlesztők két táborra szakadnak: vannak akik minden fórumon elmondják, hogy ez hülyeség és vannak akik minden fórumon elmondják, hogy a *Scrum* nagyon jó, "ha betartják"
- a fejlesztésre fordított idő kevesebb, mint a meetingekre fordított idő
- ha már nem dolgozik a cégnél projektmenedzser (kegyeleti okokból bármelyik pm "scrumosítható", ehhez a titulust *scrummaster*-re kell átírni, bár általában bárki lehet *scrummaster*, a projektmenedzseri tudás inkább hátrány)

- ha többen állítják magukról, hogy *kanbanoznak*, mint akik azt állítják, hogy *scrumoznak*
- ha végleg megszűnnek a specifikációk és egzakt kérések
- ha már minden héten (vagy még többször) van éles szerverekre kitöltés (még ha azt azonnal vissza is kell szedni)

Sok sikert!

Soha ne felejtssük el, bármi hívható
Scrumnak és egy cégnél akárhány
csapat scrumozhat!

A Scrum rugalmas és minden
problémára megoldást nyújt!