

Article original a: [Tutsplus - SCRUM: The Story of an Agile Team](#)

Scrum és una de les tècniques àgils més usades. No va de programar; sinó que s'enfoca en l'organització i la gestió del projecte. Si tens una estona, deixa'm que t'expliqui com funcionem al nostre equip i com hem adoptat Scrum.

Una breu història

Les arrels de Scrum s'extenen més enllà de l'Era Àgil. La primera menció a aquesta tècnica és del 1986 (Takeuchi i Nonaka) pel desenvolupament de productes comercials. El primer paper on es descriu oficialment Scrum, escrit pel Jeff Sutherland i en Ken Schwaber és del 1995. Poc després (2001) van aparèixer l'[Agile Manifesto](#) i el llibre [Agile Software Development with Scrum](#) (Schwaber i Beedle).

AIXÍ ÉS COM VAM COMENÇAR

Tot depen de la dedicació i de la voluntat. Si vols que el teu equip sigui eficient, productiu i lliuri a temps la feina, hauràs d'adoptar d'alguna forma les tècniques àgils. Scrum pot ser ideal o no per vosaltres, però és probablement una de les millors maneres de començar.

L'equip de desenvolupament (l'equip) on treballa ja coneixia força d'Àgil. Vam canviar del desenvolupament en Cascada a processos més àgils i alliberàvem amb freqüència. Vam aconseguir treure versions periòdicament entre els tres i sis mesos, amb un número de bugs decentment baix. Tot i així estàvem lluny del que hem conseguit. Trobàvem a faltar algun procés, o regles, que ens forcessin a canviar la nostra perspectiva sobre el desenvolupament. Aquest era el moment quan el nostre coordinador d'equip ens va introduir Scrum. Aquesta persona va prendre el rol de Scrum Master.

El Scrum Master

El **Scrum Master** és un dels rols més importants. És responsable de crear un pont entre el **Product Owner** i l'**Equip**. Aquesta persona té habitualment un fort coneixement tècnic i participa activament dins del desenvolupament. És comunica amb el **Product Owner** sobre **Històries d'usuari** (històries) i com s'organitza el **Backlog**.

El Scrum Master coordina l'equip de desenvolupament però no "microgestiona" (l'equip s'autoorganitza). Al principi, però, el Scrum Master pot "controlar" a una part de l'equip per millorar la seva interacció i autoorganització.

INTRODUÏNT EL CONCEPTE DE "SPRINT"

Abans de conèixer el concepte Sprint no podiem imaginar alliberar versions entre tres i sis mesos, pensàvem que era massa ràpid i que no ens donava temps per integrar i polir els errors dels nostres productes.

Sembla contradictori, com podem tenir el producte estable cada setmana? Però realment és possible. En primer lloc vam escurçar els nostres cicles de producció de tres mesos a sis setmanes i finalment a quatre. Això es va fer sense canviar el nostre estil, simplement

introduïnt regles per fer Sprints en menys d'un mes. No pot haver màgia, han de ser regles que beneficiïn el vostre projecte. El primer canvi va ser aplicar el concepte de **Sprint Planning**.

Sprint Planning

Aquest és una de les reunions que proposa Scrum. Abans de començar cada Sprint es reuneixen l'equip, el product owner i el scrum master per planificar el Sprint. La durada de la reunió pot allargar-se d'un parell d'hores a un dia complet. El que fem es revisar el product backlog i decidir quines històries s'inclouran al Sprint.

The Product Owner

És la persona responsable de definir les històries i mantenir el product backlog. També és un enllaç entre l'equip i la direcció. El product backlog avalua les peticions dels stakeholders (persones rellevants), la direcció, els usuaris i d'altres peticions (informes de bug, enquestes als usuaris, etc.)

Aquesta persona prioritza el backlog, proporcionant el màxim valor als stakeholders en el temps més curt. Aconsegueix això assignant la feina del màxim valor als sprints de l'equip. Sembla sofisticat i... ho és! De fet, donar la direcció al desenvolupament del producte endevinant quines funcionalitats són les que aporten més valor és la part més difícil del procés. De vegades és molt senzill, quan molts usuaris demanen una funcionalitat concreta.

Si es proporciona una funcionalitat molt demanada, s'augmenta la lleialtat dels usuaris que la demanen. Però això pot ser difícil perquè implementar una altra funcionalitat pot fer el sistema útil a molts més usuaris. Quina és la elecció correcta llavors? Depen de la direcció que porti el negoci i el Product Owner ha de prendre la decisió d'on vol portar el producte. A la nostra empresa aquestes decisions es comuniquen amb l'equip, no és un requeriment de Scrum però és molt útil per orientar als nous equips. Compartir informació ajuda a comprendre perquè es prenen certes decisions i perquè algunes funcionalitats que semblen òbvies s'endarrereixen o "cauen".

EL TAULER DE PLANIFICACIÓ

Recordo el matí que va passar: vaig arribar a l'oficina i em vaig trobar el nostre Scrum Master preparant un tauler de planificació de tasques amb un full A4 i un "celo". No tenia ni idea de que estava fent. Com cada matí vaig anar a fer-me un café i em vaig quedar mirant que feia. Quan va acabar una pissarra blanca estava enganxada a la paret. Tenia varies columnes farcides de "post-it" (notes) de colors. D'això fa dos anys.



La pizarra ara dona suport al Procés de Desenvolupament Àgil (*Lean Development Process*) que usem avui. Recordeu, Agile va de canviar i adaptar-se al canvi. **Mai seguieu regles cegament sense saber perquè.**

Així doncs, que hi ha al tauler?

Columnes pel procés de desenvolupament

Tenim quatre columnes principals

- **Backlog de propera versió** – És on hi ha totes les històries de la “propera versió” del producte. Sí, el producte està preparat per lliurar al final de cada sprint, però això no vol dir que el lliurem normalment. Nosaltres tenim habitualment sprints setmanals.
- **Backlog del Sprint** – Quan planifiquem, negociem amb el product owner que vol que fem al sprint. Com sabem el que ens dona temps a fer? Estimant la dificultat de cada història (veure més avall “estimant històries”). El resultat final d’aquesta planificació és moure les notes de les històries que farem al sprint setmanal.
- **Treballant** – Aquesta és simple. Quan un membre de l’equip agafa una història, la passa a aquesta columna per mostrar a tothom que ell se n’està encarregant. Això no és per controlar a ningú sinó per la comunicació de l’equip. Tothom sap en que estan treballant els seus companys. A la foto de dalt, les enganxines de colors mostren qui està treballant a la història.
- **Fet!** – Aquí és on deixem les històries fetes. És molt important definir amb precisió “qué està fet”. Quan acaba el sprint totes les històries i “bugs” del sprint backlog han d’estar a aquesta columna.

Altres equips divideixen la columna de “treballant” en d’altres com disseny, programació i test d’acceptació. Cada equip pot triar quines li van bé!

La definició de “fet”

Que està “fet”? Quan podem confiar que una història està realment completa? Com ho sabem? “Fet” s’ha de definir de manera clara i precisa per que tothom sapigui quan una història està completa. Aquesta definició és pròpia de cada equip i també pot variar entre projectes del mateix equip. No hi ha cap regla exacta, i és una idea que s’ha de comentar en les reunions i decidir en grup que vol dir una “història feta”. Alguns aspectes que poden formar part són:

- Crear un disseny minimalista.
- Crear una maqueta de la interfície.
- Usar TDD o assegurar-se que estan fets els tests unitaris.
- Escriure el codi.
- Fer les proves d’acceptació creuades entre membres de l’equip.
- El sistema complet es pot compilar i muntar amb el nou codi.
- Els tests d’acceptació se superen i el codi s’ha integrat al sistema.

Hi ha múltiples idees del que es pot incloure en la definició de fet. Pren la part que consideris necessària pel teu equip, fes-la servir a fons i fes-la evolucionar amb el temps. Si noteu que la definició s’està quedant desfasada, podeu eliminar o afegir idees útils. A la foto de dalt, els notes verds descriuen el que considerem que s’ha fet, per cada part del procés.

OMPLINT EL TAULER

Com ho farem, ens preguntàvem? Fins llavors no feiem servir els notes per planificar. Feiem servir un software per gestionar les històries d’usuari i els bugs, però res més. Després de dinar el scrum master es va presentar amb una muntanya de notes de colors. Ens va explicar que eren mentre en omplia una dotzena.

Les històries d’usuari

Una història d’usuari és una frase breu que defineix una característica o funcionalitat.

Aquestes representen les principals funcionalitats que volem implementar. Tenen el nom de històries d’usuari perquè representen la perspectiva d’un usuari. Naturalment amb

usuari ens referim a una persona que fa servir l'aplicació que representa un grup o rol (p.e. administrador del sistema, cap, usuari restringit, etc.

Un exemple d'història és algo com "Com un usuari, vull compartir una carpeta amb els meus companys d'equip".

Fins llavors no teniem un product owner definit i era el scrum master qui s'inventava les històries. Això és acceptable al principi, però **recomanem fortament** que se separin els dos rols. D'altra manera, com pot negociar el scrum master el backlog del sprint amb el product owner?

Podrieu pensar "Per què negociar? No és millor que hi hagi una única persona que decideixi que fer i quan?". No és així. Aquests dos rols quedarien influits punt de vista d'una única persona. Dues persones, per contrari, tenen més facilitat per proporcionar un camí més objectiu per l'equip i aconseguir que s'arriba a la meta final: software amb més valor.

Les històries d'usuari representen tot el que s'ha de fer, estan representades per notes grocs al tauler de la foto.

Errors, errors, errors

Fer un bon seguiment dels errors (bugs) és increïblement important.

Nosaltres també afegim els errors al tauler. Veus aquests notes vermells? Són els errors que cal que arreglem per la propera "versió".

Cada equip tracta els errors a la seva manera. El nostre equip els barreja amb la realització d'històries, però sempre comencem el Sprint resolent els errors assignats. Conec altres equips que acumulen els errors per un període de tres sprints i dediquen el quart sprint a resoldre'ls. D'altres divideixen els sprints entre 25% pels errors i el 75% per les històries, i d'altres treballen a les històries als matins i es posen amb els bugs a la tarda. Cada empresa ho fa a la seva manera.

Depen de tu trobar la millor solució pel teu equip, i per suposat donar seguiment als vostres errors. Escriviu els bugs a notess per donar seguiment als errors del vostre sistema i les tasques de solució dels errors. Donar seguiment als errors és increïblement important.

I know of other teams who pile up bugs for a period of three sprints, and spend every fourth sprint fixing them. Others split sprints into 25/75 for bugs/stories. Further, other teams may work on stories in the morning, and bugs after lunch; it simply depends on the company. It's up to you to find the best solution for your team, and of course, keep track of your bugs. Write them on sticky notes so that you can track your system's issues and the fixes for those issues. Tracking your bugs is incredibly important.

Tasques i sub-històries

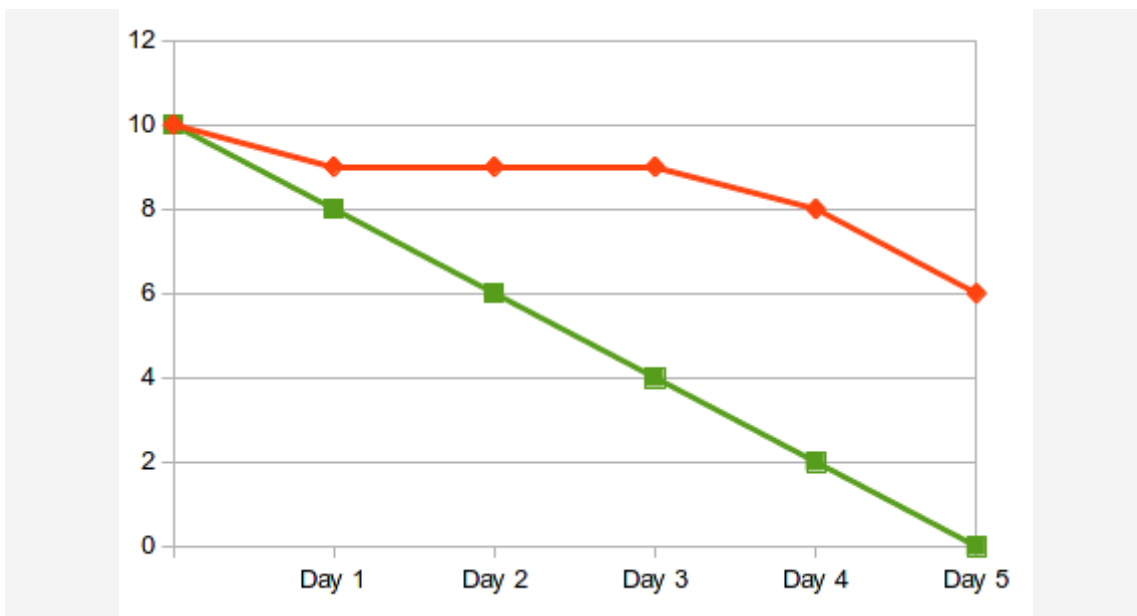
Les tasques s'escriuen com frases senzilles des del punt de vista del desenvolupador

Idealment, cada història hauria de ser prou curta per completar-se amb certa facilitat, però la divisió d'històries en d'altres més petites pot ser difícil. Alguns projectes simplement no poden permetre's aquesta possibilitat. Tot i així trobareu blocs grans de feina en els que

poden treballar més d'un membre de l'equip. És important dividir la feina en parts més petites i fàcils de gestionar.

Un enfocament és dividir històries grans en "tasques", frases simples des del punt de vista dels desenvolupadors. Per exemple, la història anterior sobre compartir una carpeta es podria dividir en varies tasques com ara: "Crear la UI per compartir", "Implementar el mecanisme de compartir en públic", "Implementar la funcionalitat de control d'accés", "Afegir checks de control d'accés a la UI", i d'altres. La idea és que es pensa més en la història cada vegada que es descomposa en tasques més petites i així el teu equip tindrà més control sobre la història.

Nosaltres fem servir notes blau per les tasques al tauler. Mireu la darrera columna ("Fet!") i veureu com agrupem les tasques a sota de les històries d'usuari. En l'exemple anterior vam dividir la història en quatre tasques.



Tasques tècniques

Algunes activitats s'han de completar per finalitzar una tasca, història o el sprint complet. Aquestes tasques són usualment relacionades amb la infraestructura, però us podeu trobar amb que les tasques requereixen canvis als sistemes. Aquest procés pot ser part o no de la història. Per exemple, pot ser que calgui instal·lar una aplicació externa per implementar la funcionalitat de compartició de carpeta. Això es considera part de la història? Podem fer-ho o no. Vam determinar que aquestes tasques s'haurien de separar de la història i això ens va ajudar a fer un millor seguiment d'aquestes tasques addicionals. Al tauler podreu trobar aquestes tasques extres representades amb notes verdes. N'hi ha una al sprint backlog i tres al testing.

El backlog tècnic

Per a un equip jove amb poca experiència amb Àgil i Scrum, és útil ressaltar aquestes tasques amb un mini-backlog.

Aquest backlog és per tasques de infraestructura, com ara actualitzar les proves automatitzades del sistema, configurar un nou servidor i d'altres coses que fan més fàcil el desenvolupament del dia a dia. Aquesta feina s'ha de completar en algun moment però no va lligada als sprints.

No cal que separeu aquests ítems a un backlog separat, de fet conec equips que els tenen al backlog comú. Nosaltres vam abandonar el backlog tècnic separat ja que considerem que aquests ítems són tan importants com la resta de tasques d'històries. Tot i així per un equip nou és útil separar-les. Proveu si us funciona, sinó barregeu-les al mateix tauler amb notes d'un altre color.