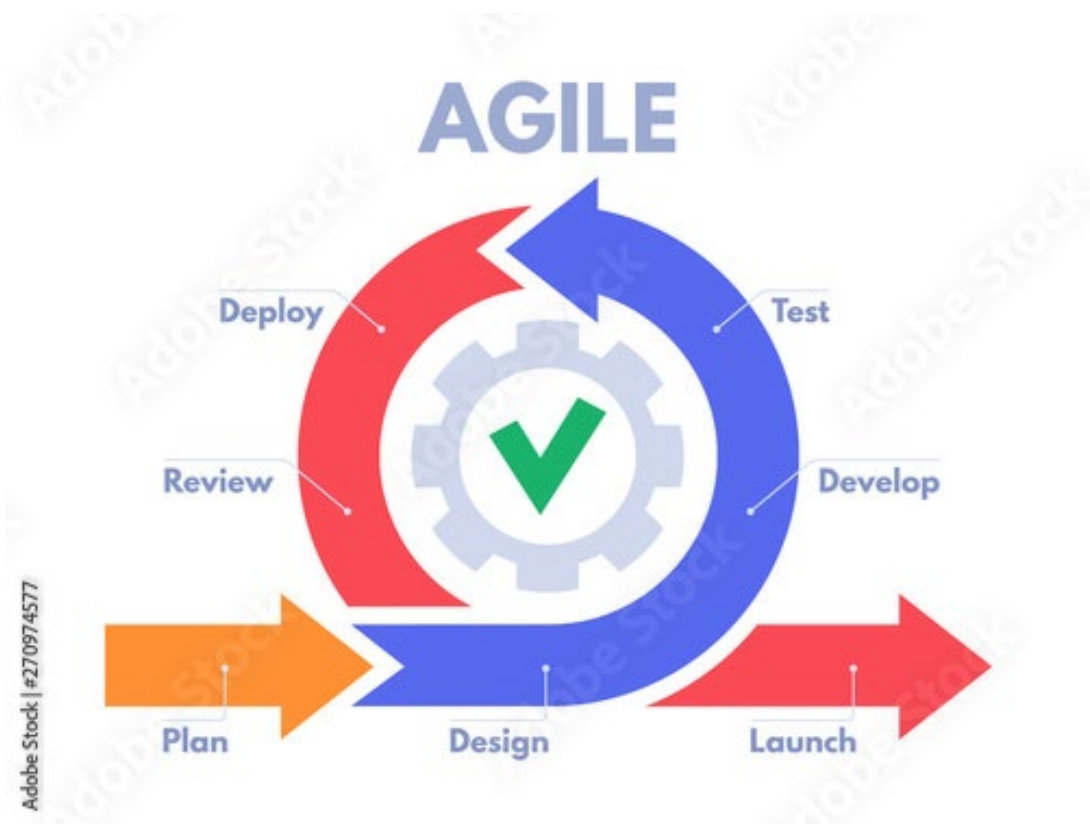


Metodoloxías áxiles

O áxil defínese como a habilidade de responder de forma versátil aos cambios para maximizar los beneficios. Responde con rapidez e da un bo resultado.

As metodoloxías áxiles son unha forma de traballar en equipo, para obter resultados rápidos, con moita interacción entre os membros do equipo de desenvolvemento de software e o cliente.



Manifesto áxil

Agile Manifesto, Manifesto áxil ou Manifesto para o desenvolvemento de software áxil é unha declaración de valores e principios sobre novas formas de desenvolver software que xurdiu en 2001, como reacción aos tradicionais métodos formais cos que se traballaba entón na industria.

1. Valorar máis aos individuos e as súas interaccións que aos procesos e as ferramentas
2. Valorar máis o software funcionando que a documentación exhaustiva
3. Valorar máis a colaboración co cliente que a negociación contractual
4. Valorar máis a resposta ante o cambio que seguir un plan



Doce principios do manifesto áxil

Os 12 principios nos que se soporta o manifesto áxil:

1. A nosa maior prioridade é satisfacer ao cliente mediante a entrega temperá e continua de software con valor.
2. Aceptamos que os requisitos cambien, mesmo en etapas tardías do desenvolvemento. Os procesos Áxiles aproveitan o cambio para proporcionar vantaxe competitiva ao cliente.
3. Entregamos software funcional frecuentemente, entre dúas semanas e dous meses, con preferencia ao período de tempo máis curto posible.
4. Os responsables de negocio e os desenvolvedores traballamos xuntos de forma cotiá durante todo o proxecto.
5. Os proxectos desenvólvense ao redor de individuos motivados. Hai que darlles a contorna e o apoio que necesitan, e confiarlles a execución do traballo.
6. O método máis eficiente e efectivo de comunicar información ao equipo de desenvolvemento e entre os seus membros é a conversación cara a cara.
7. O software funcionando é a medida principal de progreso.
8. Os procesos Áxiles promoven o desenvolvemento sostible. Os promotores, desenvolvedores e usuarios debemos ser capaces de manter un ritmo constante de forma indefinida.
9. A atención continua á excelencia técnica e ao bo deseño mellora a Axilidade.
10. A simplicidade, ou a arte de maximizar a cantidade de traballo non realizado, é esencial.
11. As mellores arquitecturas, requisitos e deseños emerxen de equipos auto-organizados.
12. A intervalos regulares o equipo reflexiona sobre como ser máis efectivo para a continuación axustar e perfeccionar o seu comportamento en consecuencia.

12 Principios del Manifiesto Ágil



Satisfacer al cliente mediante la entrega temprana y continua.

01



Aceptamos que los requisitos cambien.

02



Entregamos software funcional frecuentemente.

03



Responsables de negocio y los desarrolladores deben trabajar juntos.

04



Proyectos se desarrollan en torno a individuos motivados.

05



Método más eficiente y efectivo de comunicar cara a cara.

06



El software funcionando es la medida principal.

07



Los procesos ágiles promueven el desarrollo sostenible.

08



La atención continua a la excelencia técnica.

09



La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

10



Las mejores arquitecturas emergen de equipos autoorganizados.

11



A intervalos regulares, el equipo reflexiona sobre cómo ser más eficaz.

12



¿Que é Scrum?

Scrum é unha forma de traballar, non un proceso, no que se aplican de maneira regular un conxunto de procesos para traballar en equipo, e obter o mellor resultado posible dun proxecto. Aplícase xeralmente a desenvolvemento de software, pero pode ser aplicado para o desenvolvemento de calquera produto que implique traballo intelectual e relación co cliente. Traballos en cadea non requiren Scrum.

Dicimos que non é un proceso, di "que hai que facer" e non "como hai que facelo". A metodoloxía é fácil de entender, o importante é implantala correctamente xa que a interacción entre os participantes é fundamental.

Ideas fundamentais de Scrum son:

- Adoptar unha idea total da realización do produto, en lugar da planificación e execución completa do produto.
- Enfocarse máis nas zonas de solapamento, en lugar de realizar unha tras outra nun ciclo de cascada. Entregas progresivas do produto a modo de prototipos.
- Pon por encima aos individuos e a súa interacción por encima de procesos e ferramentas, colaboración co cliente por encima da negociación contractual.
- Eficaz resposta ante cambios, xa sexan cambios de requirimentos ou incidencias.
- Outras metodoloxías céntranse na documentación, para facer o proceso o máis independente das persoas involucradas (cada un ten totalmente definido o que ten que facer e como facelo). Scrum céntrase no contrario, as persoas, a comunicación, a interacción.

Historia

Este modelo foi identificado e definido por Ikujiro Nonaka e Hirotaka Takeuchi a principios dos 80, ao analizar como desenvolvían os novos produtos as principais empresas de manufactura tecnolóxica.

No seu estudo, Nonaka e Takeuchi compararon a nova forma de traballo en equipo, co avance en formación de melé (Scrum en inglés) dos xogadores de Rugby, por mor do cal quedou acuñado o termo "Scrum" para referirse a ela.

¿Como se usa?

Coa metodoloxía Scrum o cliente entusiásmase e comprométese co proxecto dado que o ve crecer parte por parte. Así mesmo, permítelle en calquera momento re-alinear o software cos obxectivos de negocio da súa empresa, xa que pode introducir cambios funcionais ou de prioridade no inicio de cada nova iteración sen ningún problema.

Esta forma de traballo promove a motivación e compromiso do equipo que forma parte do proxecto, polo que os profesionais atopan un ámbito propicio para desenvolver as súas capacidades.

Os principios da metodoloxía Scrum son:

- Concentración
- Priorización
- Auto-organización
- Ritmo

Beneficios de Scrum

1. O cliente pode empezar a utilizar os resultados máis importantes do proxecto antes de que estea finalizado por completo.
2. O cliente establece as súas expectativas indicando o valor que lle achega cada requisito
3. Redución de riscos
4. Predicións de tempos
5. Maior produtividade

¿Que é un Sprint?

Sprint é o nome que vai recibir cada un dos ciclos ou iteracións que imos ter dentro de dentro dun proxecto Scrum.

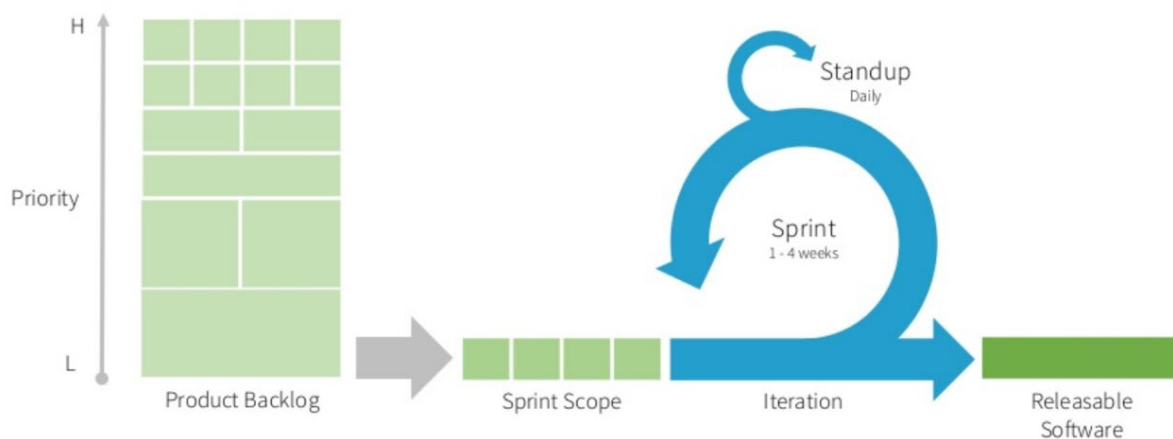
Van permitir ter un ritmo de traballo cun tempo prefixado, sendo a duración habitual dun Sprint vai entre unha e catro semanas, aínda que o que a metodoloxía di é que debería estar entre dúas semanas e un máximo de dous meses.

En cada Sprint ou cada ciclo de traballo o que imos conseguir é o que se denomina un entregable ou incremento do produto, que achegue valor ao cliente.

A idea é que cando temos un proxecto bastante longo, por exemplo, un proxecto de 12 meses, imos poder dividir ese proxecto en doce Sprints dun mes cada un. En cada un deses Sprints imos ir conseguindo un produto, que sempre, e isto é moi importante, sexa un produto que estea a funcionar.

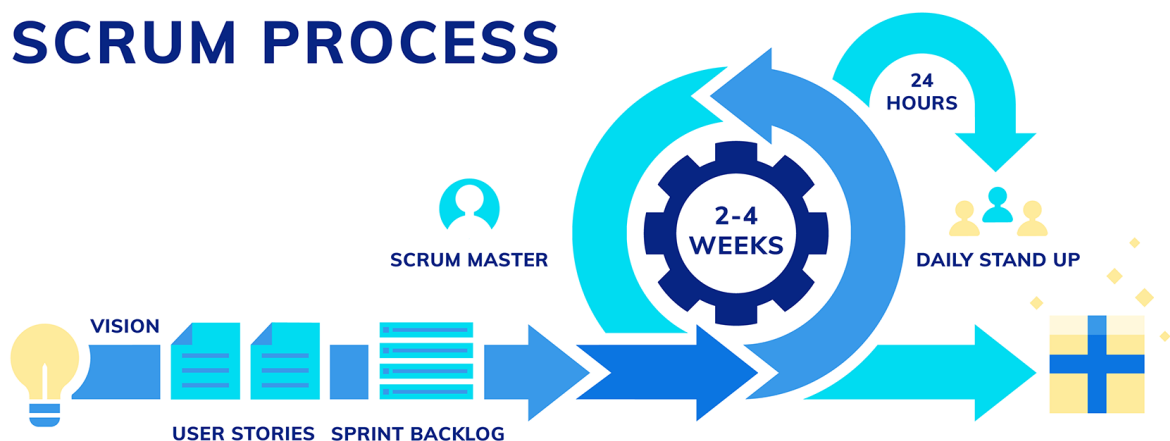
Imos velo de forma máis clara nesta imaxe:

Traditional Scrum



Na mesma temos unha pila ou Product Backlog, que serían todos os requisitos que nos pide o cliente, é dicir, o ano completo de traballo. A idea é ir seleccionando eses requisitos nos que temos a pila dividida, e ímolos a ir facendo en diferentes Sprints, e realizamos todos os pasos que conforman un Sprint, é dicir, a toma de requisitos, deseño, implementación, probas e despregamento no prazo establecido, e así imos ter sempre un software que sexa válido, un software funcionando.

SCRUM PROCESS



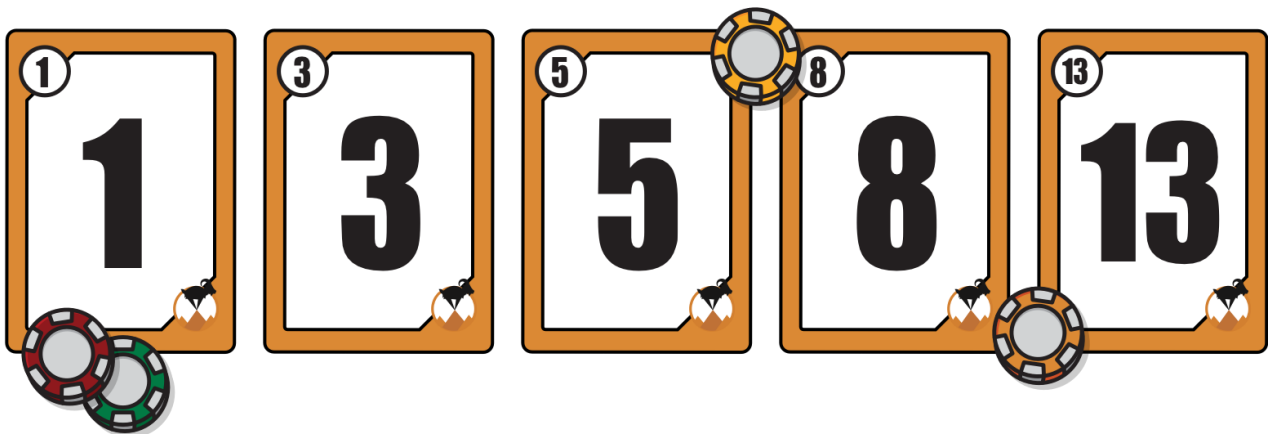
Técnicas de Priorización:

1. Planning Póker

O Planning Póker é un proceso de estimación de duración ou esforzo nas tarefas dun proxecto por parte de todo o equipo implicado. Para levar a cabo esta técnica, cada participante conta cunha baralla de cartas con números (baseados en Fibonacci: 1,2,3,5,8,13, ou similar) representando o grao de complexidade (esforzo) necesario para completar unha tarefa.

Comeza o cliente explicando un obxectivo. De forma individual, cada membro do equipo pon sobre a mesa a carta que representa a puntuación que el considera necesaria para levar a cabo dita tarefa do proxecto en cuestión. De haber puntuacións moi dispares, cada participante explica os motivos da súa decisión. E se fose necesario, pode repetirse a votación ata chegar a un consenso. Así, o resultado é unha estimación consensuada e validada por todo o equipo.

Ten moitas vantaxes: todos os membros do equipo expresan a súa opinión sen sentirse condicionados polo resto, ése máis consciente do esforzo que require unha tarefa, o que mellora a implicación. Ao sentirse partícipes, o grao de compromiso co proxecto tamén aumenta.



2. Tallas de camisetas

O sistema das tallas de camisetas é unha ferramenta de estimación de proxectos e de planificación da que che axudará a rexistrar canto tempo ou esforzo tomará unha iniciativa. Para facelo, debes asignar unha talla de camiseta a cada proxecto ou tarefa —desde XS ata XXL— que representará o esforzo relativo desa tarefa. Dependendo de como elixas utilizar esta ferramenta, a talla da camiseta pode representar o alcance da tarefa, o esforzo, a complexidade, as horas de traballo, os cálculos de tempo, ou todo o anterior.

Os 6 pasos para usar as tallas de camisetas para a estimación dos proxectos

Para usar o sistema das tallas de camisetas de maneira eficaz, é importante establecer desde o principio que representa cada talla de camiseta e cando os membros do equipo deben aclarar a talla relativa.

1. Empezar por decidir que tallas usarás

Antes de presentar o sistema das tallas de camisetas ao teu equipo, decide que tamaños usarás. Evita usar demasiados tamaños para que os membros do equipo non se confundan. Se recentemente estás a comezar, usa as tallas pequeno (S), mediano (M), grande (L) e extra-grande (XL). Se o alcance dos teus proxectos varía moito, usa as tallas extra-pequeno (XS) ou extra-extra-grande (XXL)

tamén.

2. Póñanse de acordo en canto ao que representa cada talla

O sistema das tallas de camisetas só funciona se todos entenden o que representa cada talla. Por tanto, cando presentes esta práctica de estimación de proxectos ao teu equipo, asegúrate de que todos estean en sintonía con respecto ao que representa. Pode ser útil proporcionar exemplos comúns nos que compares un proxecto pequeno e un grande. Por exemplo, no equipo de contido de Asana, un artigo como este é unha talla mediana. Doutra banda, unha iniciativa de contido que inclúe a varios departamentos con varias partes e diferentes participantes no proxecto é grande ou, nalgúns casos, incluso extra-grande.

3. Decide quen asigna as tallas de camisetas

Dependendo da estrutura do teu equipo, considera se vas limitar quen pode asignar tallas de camiseta ou se permitirás que todo o equipo fágao.

4. Asigna tallas de camiseta a cada iniciativa

Se recentemente estás a comezar, é posible que debas asignar as tallas de camiseta de maneira retroactiva ao traballo en curso. Pero, de agora en máis, a cada traballo debe asociarse unha talla de camiseta ao momento de asignalo, para que ti e o membro do equipo entendan canto esforzo representa ese traballo.

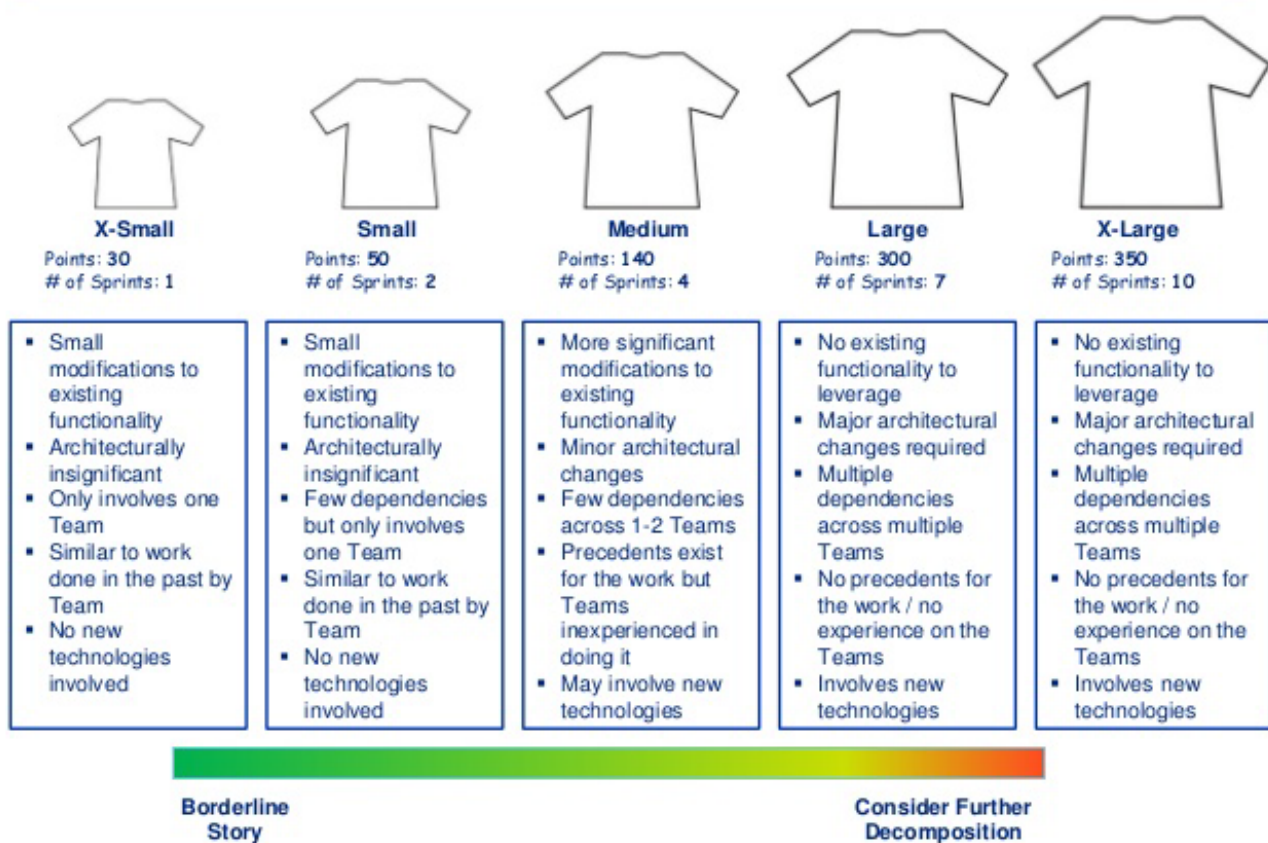
5. Dá seguimento ás tallas de camisetas cunha ferramenta de xestión do traballo

Asociar tallas de camiseta ao traballo é bo, pero ver esa información nunha ferramenta clara e de acceso centralizado é moito, moito mellor. Asegúrate de que o teu equipo de seguimento ás tallas de camisetas nunha ferramenta de xestión do traballo compartida. Dese xeito obtense información dunha ollada sobre a carga de traballo actual de cada membro do equipo.

6. Usa o sistema das tallas de camiseta para medir a carga de traballo

Unha vez que se asocie unha talla de camiseta a cada tarefa, tamén poderás dar seguimento á carga de traballo do teu equipo. Usa unha ferramenta de xestión do traballo, , para medir o esforzo asociado e ter unha idea de canto traballo debe realizar cada membro do equipo.

Sample T-Shirt sizes facilitate consistent estimation



3. MosCoW



MoSCoW é unha combinación das primeiras letras de: debe ter (Must have), debería ter (Should Have), podería ter (Could Have), e non terá nesta ocasión (Would not Have).

Nesta técnica, asignamos unha das letras a cada unha das características, en base á seguinte definición:

- **M** (Must have) Debe ter: Requisito que ten que estar implementado na versión final do produto para que a mesma poida ser considerada un éxito.

- **S** (Should have) Debería ter: Requisito de alta prioridade que na medida do posible debería ser incluído na solución final, pero que chegado o momento e se fose necesario, podería ser prescindible se houboese algunha causa que o xustificase.
- **C** (Could have) Podería ter: Requisito desexable pero non necesario, implementaríase se houboese posibilidades orzamentarias e temporais.
- **W** (Won't have) non terá esta vez: Fai referencia a requisitos que están descartados de momento pero que nun futuro poderían ser tidos en conta e ser reclasificados nunha das categorías anteriores

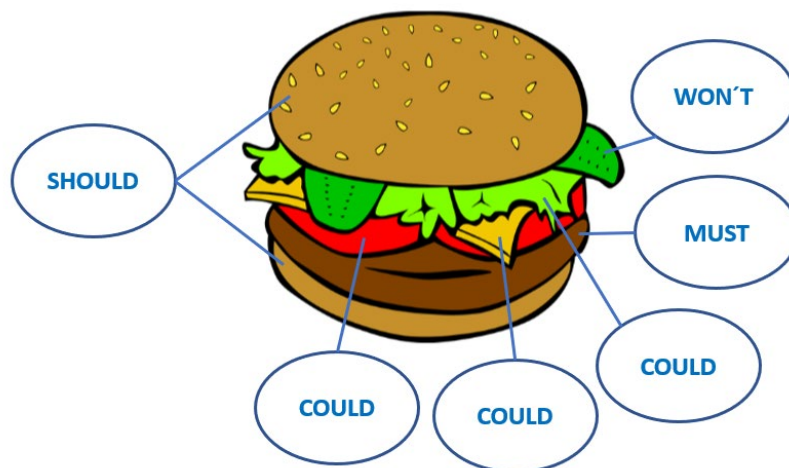
Se asignamos estas prioridades sendo realistas, saberemos que o noso produto mínimo aceptable é o que contén todas as historias de usuario coa letra M.

O produto desexable sería aquel que contén todas as historias de usuario M e S, e o produto ideal é a que contén todas as historias de usuario M, S e C.

Exemplo de MoSCoW: a hamburguesa

Se tiveses que preparar unha hamburguesa, como priorizarías os seus ingredientes?

- A carne (ou substitutivo, se es vexetariano) é imprescindible, é o must. Sen iso, non hai hamburguesa que valla.
- O pan non é imprescindible, xa que te podes comer a hamburguesa nun prato con coitelo e garfo. Pero estarás comigo en que sen o pan de sésamo a cousa perde moitísimo. Sería un should.
- Queixo, tomate, leituga... dan bastante graza á hamburguesa. Enriquécena, pero non ocupan o posto de importancia dos dous ingredientes previos. Son un could.
- Os pepinillos. A non ser que sexas moi fanático deles son un ingrediente máis ben prescindible. Serían un won't.



4. Matriz valor-viabilidade ou valor-coste

Neste modelo non só se ten en conta o valor de negocio do requisito, tamén o custo de levalo a cabo. En función de ambos os parámetros, valor e custo, clasificamos os requirimentos dentro de catro áreas, na chamada Matriz de Valor vs. Custo.

As Historias de Usuario puntúanse segundo o seu valor e custo de implementación. En base a esta puntuación, a historia de usuario caerá nunha destas catro áreas da matriz e determinará o seu grao de prioridade.

O custo non ten que significar necesariamente diñeiro. Pode facer referencia ao esforzo ou complexidade de desenvolvemento, polo que este método tamén se denomina a miúdo Valor vs. Esforzo

ou Valor vs. Complejidade ou Bang for the buck, que referencia a unha expresión coloquial que poderíamos traducir como saca o maior proveito pola túa pasta, ou algo así.

O obxectivo principal deste método é que tentamos maximizar a entrega de valor ao longo do tempo. É dicir, se traballamos cun prazo de lanzamento determinado, traballamos nos elementos máis valiosos que podemos incluír no período. Desta forma non centramos no máis valioso co menor esforzo posible.



Produto Mínimo Viable (MVP)

O Produto Mínimo Viable (MVP) é un produto moi básico, coas funcionalidades esenciais e que permite probar a reacción que ten o público obxectivo. Co seu feedback, é posible reconstruír e mellorar o produto, para lanzar unha nova versión: o MVP2, e realizar o mesmo proceso. Este ciclo repítese varias veces para ir puíndo o produto en función do que o cliente demanda e identificar por que está disposto a pagar.

As súas principais características son:

O MVP non trata de crear produtos "mínimos". O MVP é o produto final pero só coas características imprescindibles para que se poida poñer a proba. Debe ter todo o que necesita o usuario, e o seu testeo por parte dos early adopters fará que se inclúan ou exclúan características.

O MVP é unha primeira versión do produto. É a versión que permitirá recoller o máximo de información posible, validando todas as súas características co mínimo esforzo de tempo e recursos.

Conseguir o máximo de feedback a través do MVP require esforzo por parte da empresa: necesítase falar con clientes, obter métricas, analizar a resposta dos early adopters, etc.

Unha boa definición de 'Produto Mínimo Viable' sería "o mínimo polo que un cliente estaría disposto a pagar".

¿Cómo desarrollar un MVP o producto mínimo viable?



Fases de un Sprint

Cando falamos de Sprint en Scrum técnico, engloba todo o proceso, é dicir, desde que decidimos que imos facer para ese Sprint, ata que estudamos como traballamos nese Sprint.



Cando estamos nun proxecto Scrum para cada Sprint imos ter unha serie de reunións:

1. **Reunión de Planificación do Sprint.** É a primeira reunión do Sprint, e nela imos decidir o que imos facer e como o imos a facer, o número de tarefas ou de historias de usuario que imos realizar no Sprint.
2. **Reunións de Scrum diario,** que van ser pequenas reunións cos membros do equipo.
3. **Revisión do Sprint,** na que imos aceptar ou denegar o Sprint.
4. **Reunión de retrospectiva,** onde imos ver como traballou o equipo e que problemas tivo durante o desenvolvemento e como o podemos corrixir.
5. **Sprint grooming o refinamento**

O Sprint engloba todo lo anterior, desde que comeza o mesmo hasta que es aceptado ou denegado, e o equipo se pregunta como traballou.

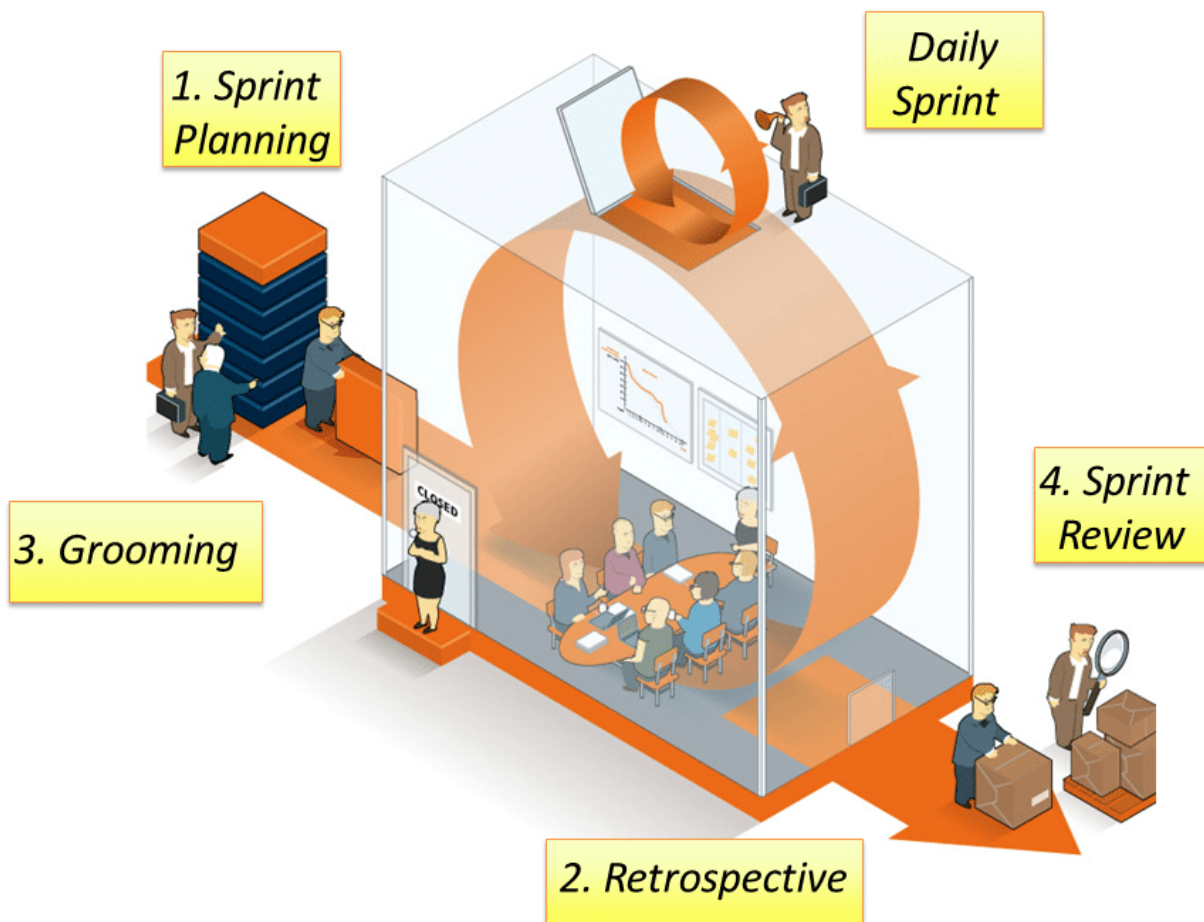


Image from : SoftHouse

1.- Reunión de Planificación del Sprint

- Lévese a cabo ao comezo do ciclo
- Seleccionar que traballo se fará (tarefas do Product Backlog)
- O Product Owner presenta as funcionalidades (User stories) e as súas prioridades (valoradas en story points). O equipo acorda con el as actividades para desenvolver nese Sprint. Ten que haber consenso (somos capaces de facer todas esas actividades no período que dura este sprint?)
- Esas actividades seleccionadas almacénanse no Sprint Backlog e conforman o obxectivo (Goal) do sprint actual. Ata aquí a primeira parte da reunión.
- Agora o equipo planifica as actividades, stories, e pode ser necesario volver a re-estimar as actividades que se poden chegar a realizar neste sprint. Cada unha destas actividades se "despezan" en tarefas que poden ser desenvoltas por cada membro técnico do equipo. É cada membro do equipo o que toma a tarefa ou tarefas que vai realizar.
- Esta reunión ten unha duración máxima de oito horas.
- A partir deste intre, todo o equipo ponse a traballar nas tarefas asignadas.

Queda claro que o feedback de pártia cliente, co Product Owner, é unha parte fundamental desta metodoloxía. O seguimento constante impide que se chegue a unha entrega final de produto e o cliente non estea satisfeito, como pode ocorrer cunha metodoloxía non áxil.

Sprint Planning Meeting

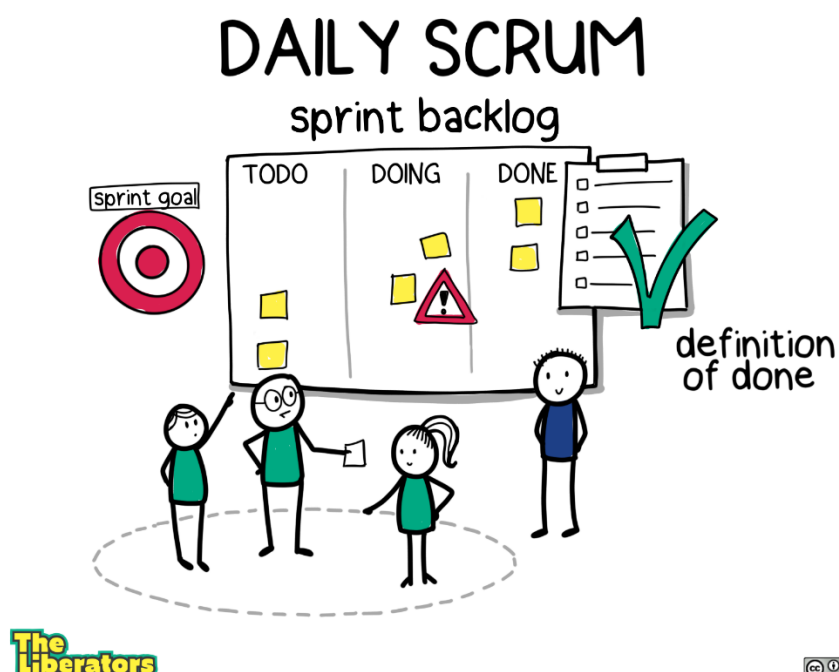
A



2.- Daily Scrum

É unha reunión diaria coas seguintes características:

- A reunión comeza puntualmente á súa hora.
- Todos os asistentes deben manterse de pé.
- A reunión debe ocorrer na mesma localización e á mesma hora todos os días.
- Tempo límite da reunión 15 minutos.
- Durante a reunión, cada membro do equipo contesta a tres preguntas:
 1. Que fixeches desde onte?
 2. Que é o que estás a planear facer hoxe?
 3. Tiveches algún problema que che impedira alcanzar o teu obxectivo? Esta última pregunta pode modificar o traballo dese día dalgún membro do equipo, se isto é necesario para que outro membro poida realizar as súas tarefas.



3.- Reunión de Revisión del Sprint

É unha reunión para revisar o traballo que foi completado e non completado.

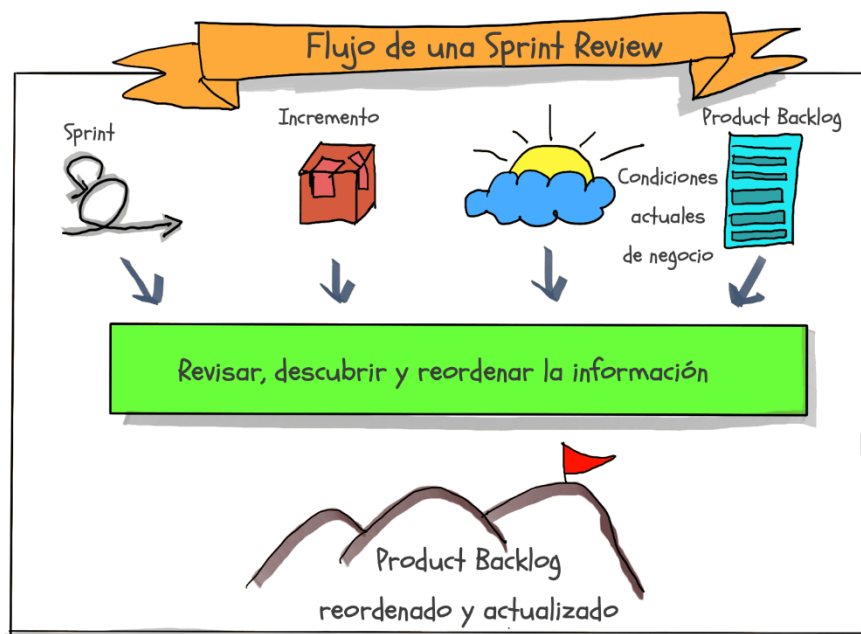
Debe estar presente o Propietario do Produto. A relación co cliente é máis intensa que en metodoloxías tradicionais.

Presentar o traballo ao Propietario do Produto, mediante unha demo para a súa aprobación. Aínda que teña algún erro emendable, o produto pode ser aceptado.

O traballo incompleto non pode ser demostrado.

A falta de éxito dun sprint vén dada porque o produto presentado non corresponde ás necesidades do cliente ou ben que non deu tempo a resolvelo completamente. Haberá que telo en conta para que non ocorra no seguinte sprint.

Catro horas como límite.

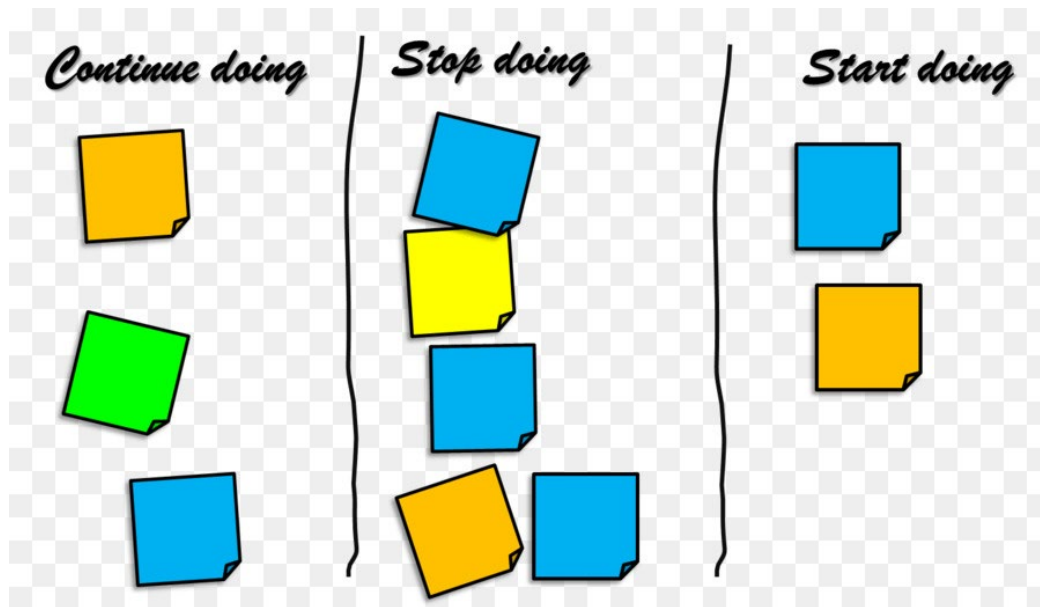


4.- Retrospectiva del Sprint

Despois de cada sprint, leva a cabo unha retrospectiva do sprint, na cal todos os membros do equipo deixan as súas impresións sobre o sprint recentemente superado. O propósito da retrospectiva é realizar unha mellora continua do proceso. As tres preguntas desta reunión son:

1. En que melloramos desde o noso último sprint? É dicir, que debemos seguir facendo.
2. En que Podemos mellorar para os próximos sprints?
3. Que deberíamos parar de facer?

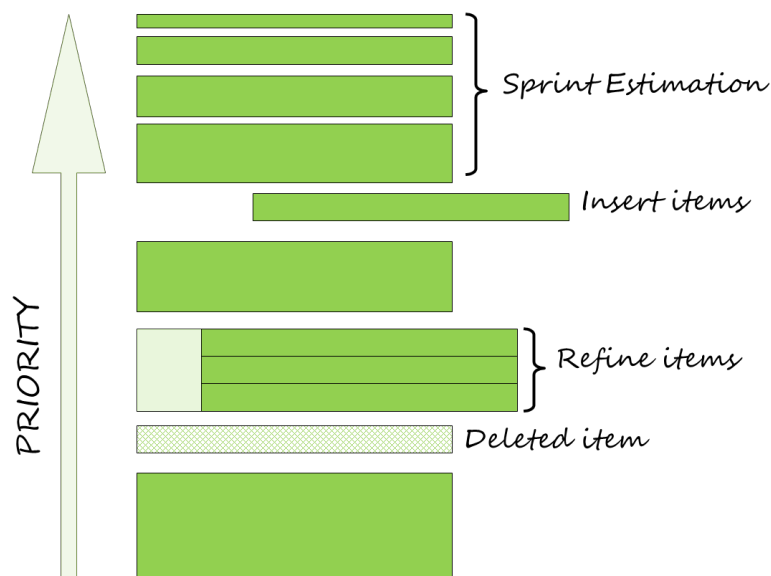
Esta reunión ten un tempo fixo de catro horas.



5.- Sprint grooming o refinamento

O refinamento do Product Backlog, é un dos eventos desenvolvidos dentro dos marcos áxiles, principalmente Scrum, onde temos a oportunidade de revisar, agregar, remover e cambiar detalles, así como de estimar ou re-estimar e ordenar as historias de usuario que están pendentes. Non existe unha forma predeterminada para realizar un refinamento do Product Backlog, isto en parte, porque é un proceso que se desenvolve de forma continua, e por tanto vólvese único dentro de cada equipo e para cada proxecto. Estas sesións normalmente son dirixidas polo PO (Product Owner). Pero en ocasións dependerá a estrutura organizacional da empresa, onde a sesión de refinamento pode ser dirixida por un Project manager ou por un Scrum máster, así como calquera outro membro do equipo.

PRODUCT BACKLOG GROOMING



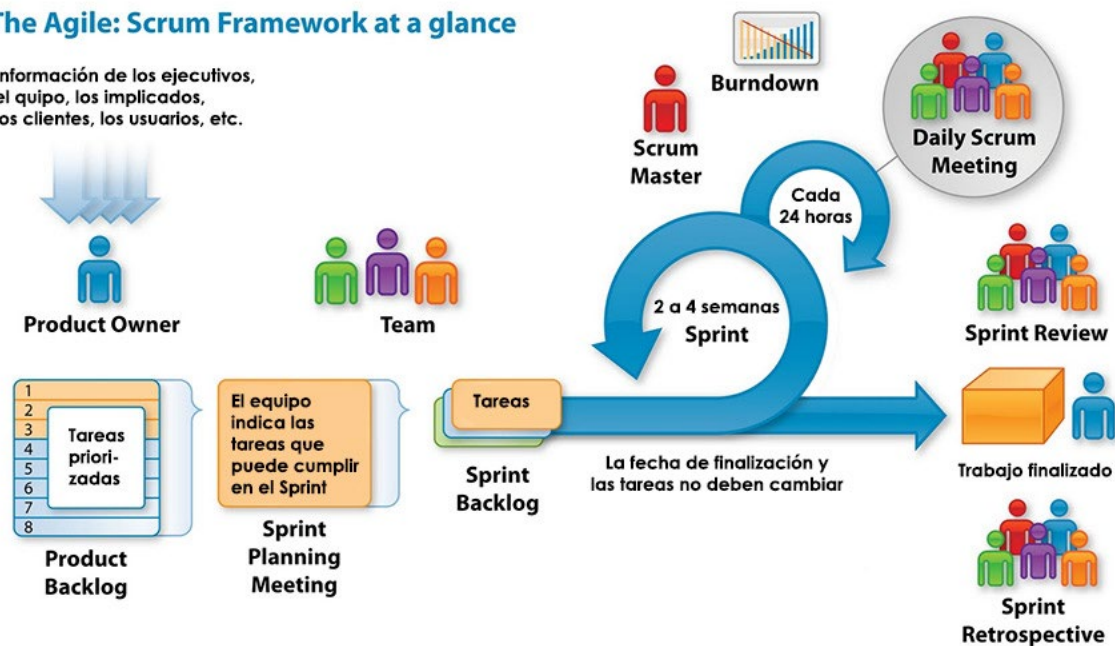
www.agile-scrum.be

Documentos del Scrum

No marco de trabajo Scrum, denominamos Artefacto a aquellos elementos físicos que se producen como resultado da aplicación de Scrum. Os tres principais artefactos ou ferramentas Scrum son: o Product Backlog, Sprint Backlog e o Incremento.

The Agile: Scrum Framework at a glance

Información de los ejecutivos,
el quipo, los implicados,
los clientes, los usuarios, etc.

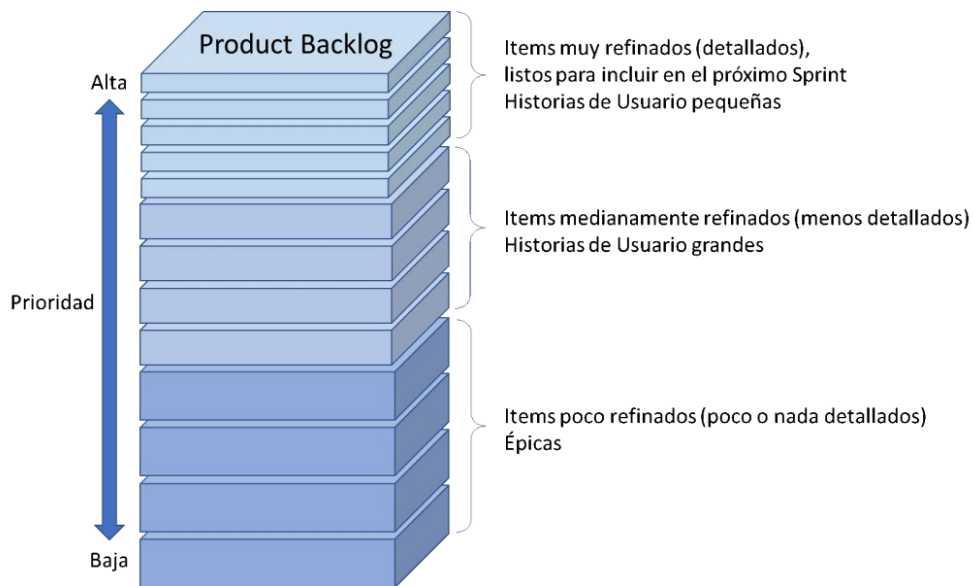


Product Backlog

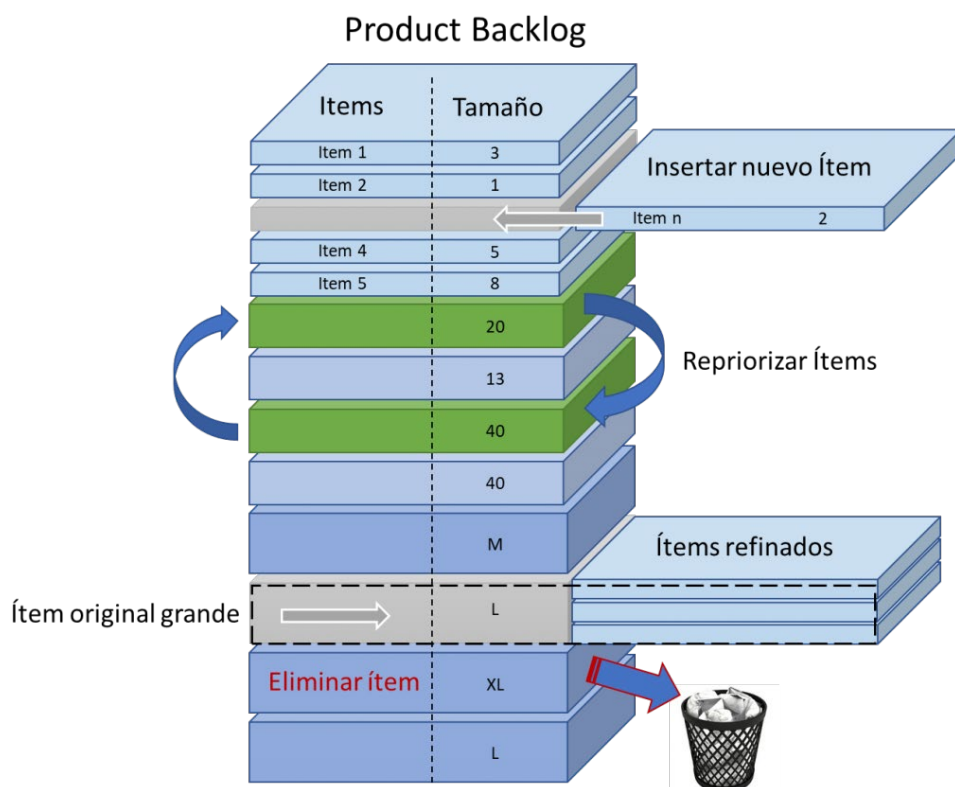
▼ Product backlog		New story	
<input type="checkbox"/>	253 List tasks with no parent stories under a quasi-story in bac...	New	
<input type="checkbox"/>	138 Indicate tracker type (via issue id's bgcolor)	New	2
<input type="checkbox"/>	145 Download pretty-fied roadmap report	New	6
<input type="checkbox"/>	55 Show/hide closed items or subtasks	New	2
<input type="checkbox"/>	40 Show or hide the issue #	New	1
<input type="checkbox"/>	135 Show/hide the main backlog	New	0
<input type="checkbox"/>	21 Show warning when server connection is lost	New	4
<input type="checkbox"/>	223 Support versions from parent projects	Feedback	0
<input type="checkbox"/>	213 Hint text on "talk" tab doesn't display in Safari	New	0
<input type="checkbox"/>	214 "Talk" button often fails	New	0
<input type="checkbox"/>	137 Append an email to the item's discussion section	New	6
<input type="checkbox"/>	224 Spreadsheet import for stories/tasks	New	0
<input type="checkbox"/>	225 Offer to copy tasks when copying a story	New	0
<input type="checkbox"/>	226 Time log widget	New	0

O product backlog é un documento de alto nivel para todo o proxecto. Contén descrições xenéricas de todos os requirimentos, funcionalidades desexables, etc., priorizadas segundo o seu valor para o negocio (business value). É aberto e calquera pode modificalo, aínda que o Product Owner é o seu

responsable. É frecuente que ao final dun sprint sufra algunha modificación.



O Product Backlog en Scrum modifícase durante todo o proxecto. Se é necesario pódense agregar novas funcionalidades e requisitos. As entradas existentes poden modificarse, definirse con máis detalle ou mesmo eliminarse. As funcionalidades e requisitos (alcance) non están completamente definidos desde o principio. Pola contra, o Backlog desenvólvese de forma iterativa, xunto co software resultante. Esta forma de traballar é completamente diferente ao desenvolvemento tradicional en ferverza, onde o alcance e a planificación están completamente definidos e detallados desde o inicio, pero en cambio, permite maximizar a entrega de valor ao cliente de forma iterativa e minimiza o esforzo de desenvolvemento. Isto permite detectar erros na súa etapa máis temperá e por tanto minimizar os riscos.



Sprint backlog

O sprint backlog é un documento detallado onde se describe como o equipo vai implementar os requisitos durante o seguinte sprint. Consiste en tomar as actividades de maior prioridade do Product Backlog. As tarefas ordénanse por prioridade, coas súas horas estimadas de dedicación, pero ningunha tarefa de duración superior a 16 horas. Se unha tarefa é maior de 16 horas, deberá ser rota en maior detalle. As tarefas no sprint backlog nunca son asignadas, son tomadas polos membros do equipo do modo que lles pareza oportuno.

Pódese engadir a este documento a porcentaxe de tarefa realizada, para ver de forma fácil e visual o progreso do sprint.

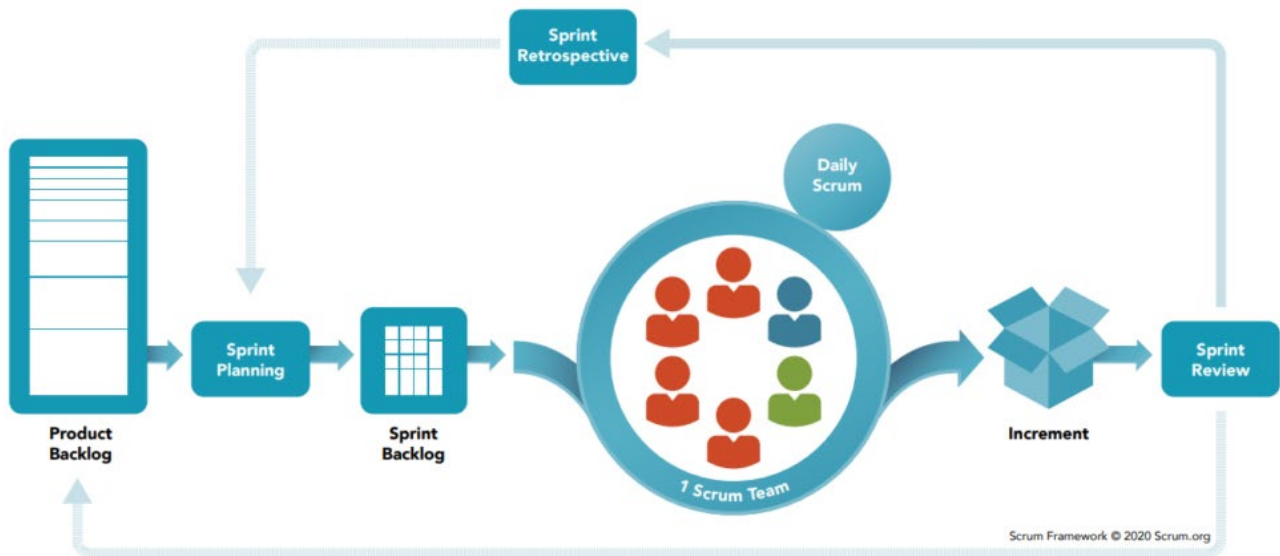
User Story	Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	...
As a member, I can read profiles of other members so that I can find someone to date.	Code the ...	8	4	8	0		
	Design the ...	16	12	10	4		
	Meet with Mary about ...	8	16	16	11		
	Design the UI	12	6	0	0		
	Automate tests ...	4	4	1	0		
	Code the other ...	8	8	8	8		
As a member, I can update my billing information.	Update security tests	6	6	4	0		
	Design a solution to ...	12	6	0	0		
	Write test plan	8	8	4	0		
	Automate tests ...	12	12	10	6		
	Code the ...	8	8	8	4		

Outra forma de mostrar as tarefas de sprint é mediante una *pizarra de tarefas*:



Incremento

Un Incremento é o resultado do Sprint, é a suma de todas as tarefas, casos de uso, historias de usuario e calquera elemento que se desenvolveu durante o Sprint e que será posto ao dispor do usuario final en forma de software, achegando un valor de negocio ao produto que se está desenvolvendo.



User Story

Scrum non atende a un formato ou contido concreto do Backlog. Con todo, impónse un formalismo e este atópase case de maneira sistemática en todos os proxectos Scrum: trátase da noción de User Narrativa, que se toma prestada de XP.

Unha User Narrativa é unha descrición sinxela e comprensible dun elemento de funcionalidade con valor "de negocio" para o sistema. Por tanto, débese expresar correctamente desde o punto de vista do usuario. Se guía polas respostas a estas tres preguntas:

- Quen realiza a petición ou quen se beneficia da petición? (rol usuario)
- Cal é a petición? (a necesidade)
- Cal é o valor para o negocio que se deriva da realización desta necesidade?

A continuación enuméranse algúns exemplos (veremos máis adiante como redactar correctamente as User Stories):

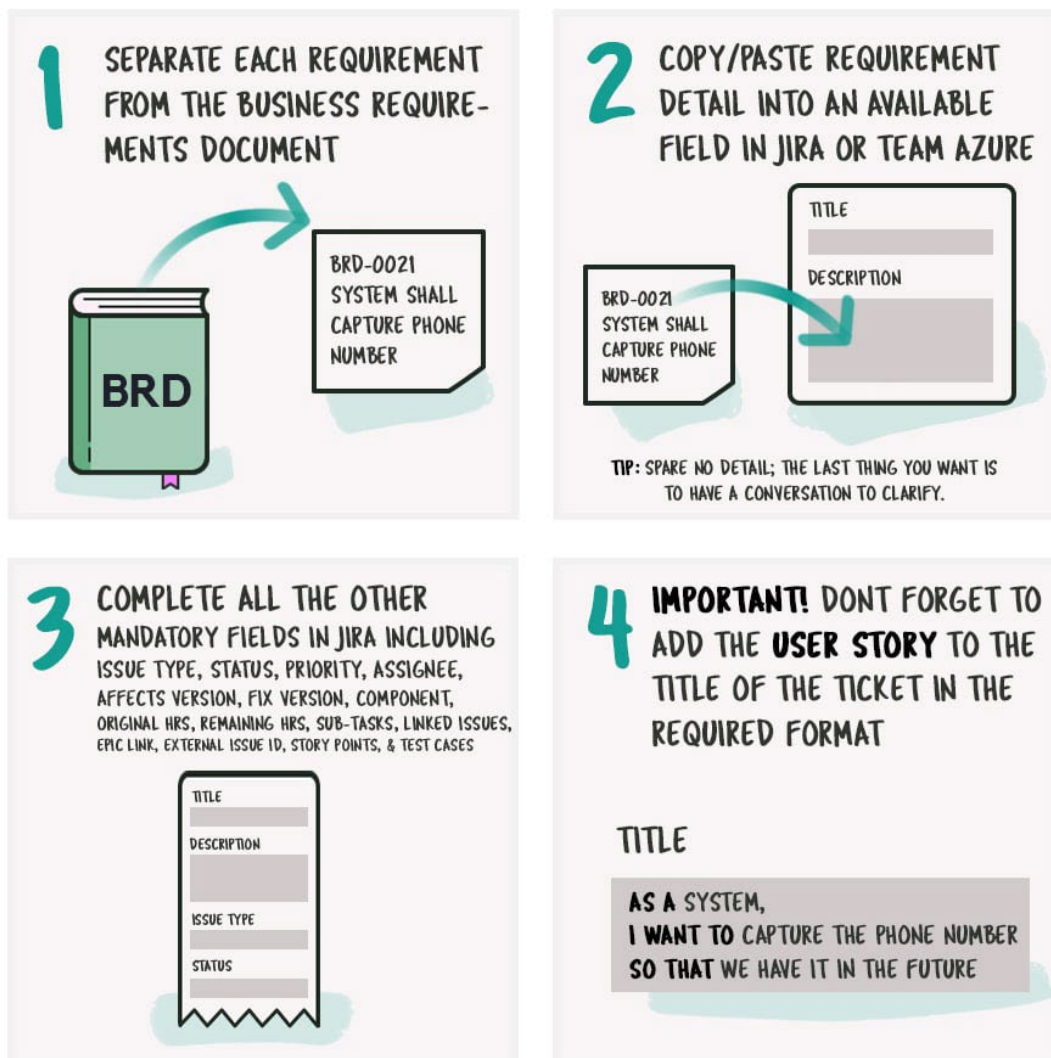
- "Desexo que o IVE se calcule automaticamente nas facturas".
- "Desexo poder eliminar os clientes que realizan pedidos fai máis dun ano".

As historias de usuario teñen varios beneficios crave:

- As historias centran a atención no usuario. Unha lista de tarefas pendentes mantén ao equipo centrado en tarefas que deben completarse, pero un conxunto de historias manteno centrado en solucionar problemas para usuarios reais.

- As historias permiten a colaboración. Co obxectivo definido, o equipo pode colaborar para decidir como ofrecer un mellor servizo ao usuario e cumprir co devandito obxectivo.
- As historias impulsan solucións creativas. As historias fomentan que o equipo pense de forma crítica e creativa sobre como lograr mellor un obxectivo.
- As historias motivan. Con cada historia o equipo de desenvolvemento goza dun pequeno reto e unha pequena vitoria, o que aumenta a motivación.

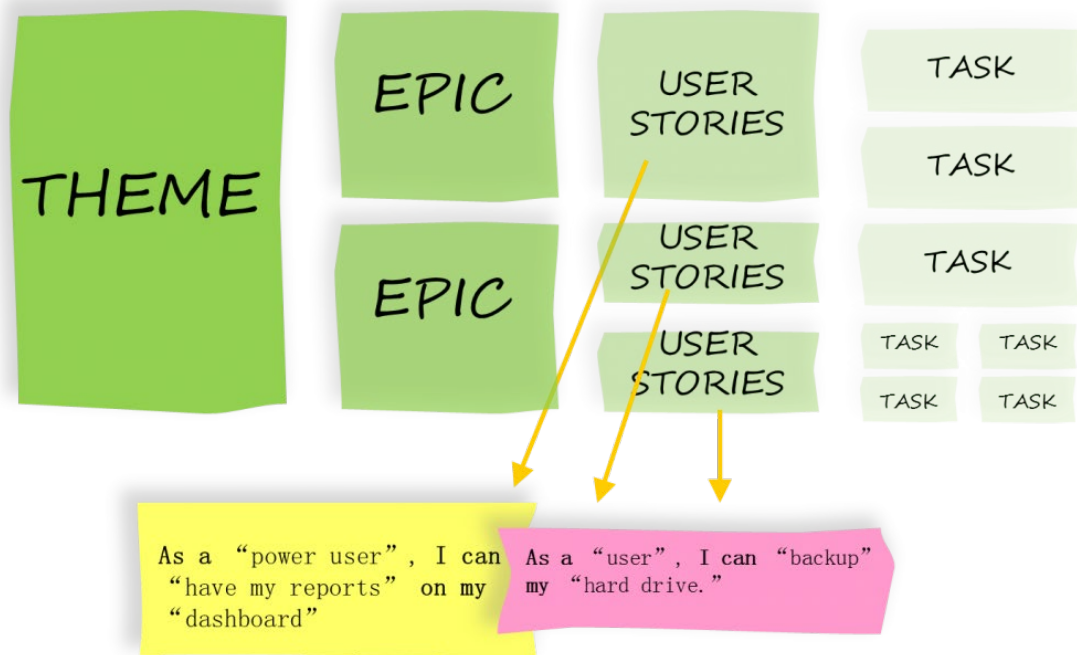
HOW TO USER STORY



WWW.VITALITYCHICAGO.COM

Ademais, tamén se usa a noción de Epic Narrativa. Pódese considerar unha Epic como unha "macro User Story", é dicir, engloba na súa definición un subconxunto de User Stories.

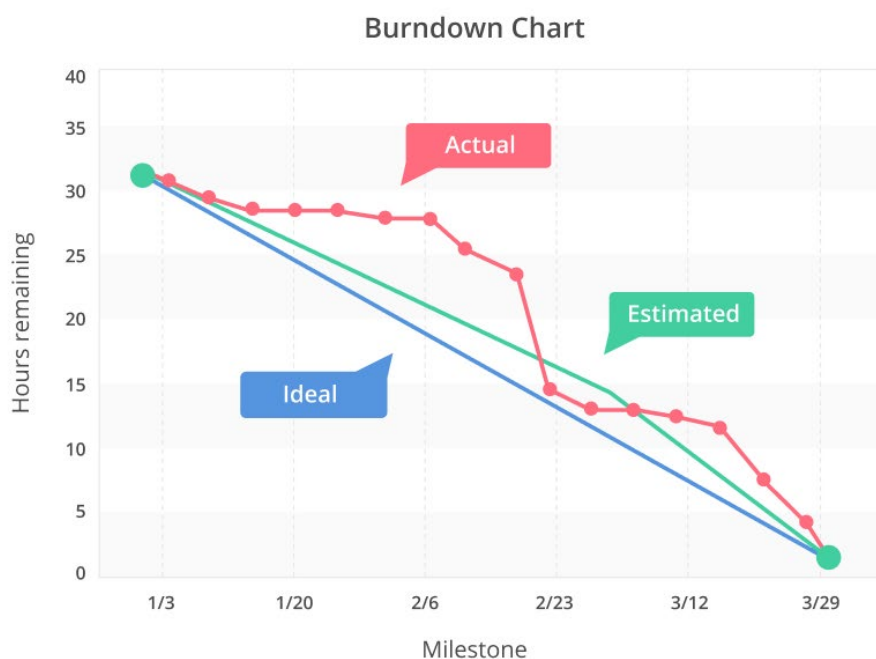
USER STORIES



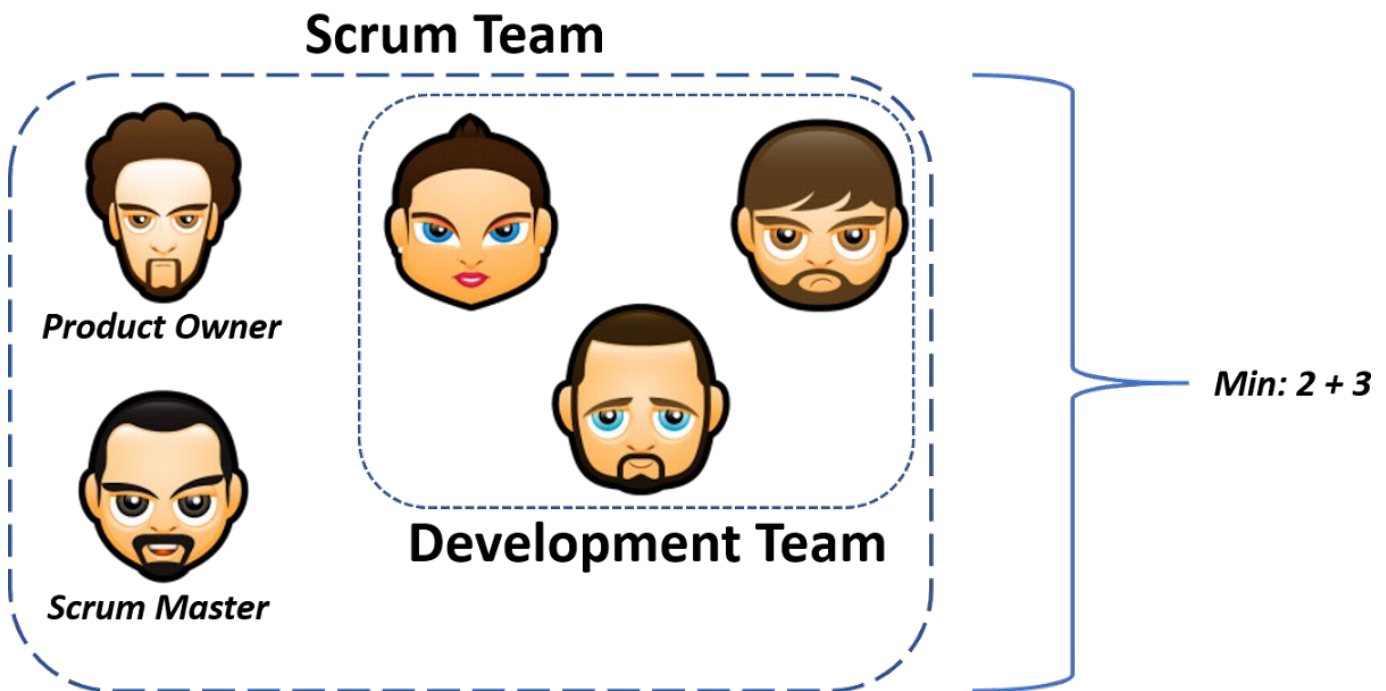
www.agile-scrum.be

Burn down chart

A burn down chart é unha gráfica mostrada publicamente que mide a cantidade de requisitos no Backlog do proxecto pendentes ao comezo de cada Sprint. Debuxando unha liña que conecte os puntos de todos os Sprints completados, poderemos ver o progreso do proxecto. O normal é que esta liña sexa descendente, ata chegar ao eixo horizontal, momento no cal o proxecto se terminou.



Roles



ScrumMaster

O ScrumMaster é o responsable de xestionar o proxecto, facilitar a comunicación entre os compoñentes e que se sigan as pautas da metodoloxía. Tamén ten que resolver todas incidencias que se vaian presentando.

O ScrumMaster ten que explicar e convencer ao quipo do uso dos procesos de Scrum, cando en moitos casos non ten autoridade xerárquica sobre eles.

As incidencias que ten que resolver poden estar relacionadas co cliente, ou con terceiras partes involucradas no produto, por exemplo, provedores.



Product Owner

O Product Owner representa a voz do cliente. Sabe exactamente que é o que o cliente quiere. Nun mundo perfecto, sería o mesmo cliente, pero en xeral adoita ser un intermediario. O Product Owner ten que coñecer en detalle todos os requisitos que se teñen que desenvolver, coa prioridade de cada un deles e as súas dependencias.

Debe ser tamén capaz de contestar a todo tipo de cuestións relacionadas co produto e é o responsable do mantemento do Product Backlog e a descrición das actividades.

Outra responsabilidade é decidir, negociado co equipo, as actividades de cada Sprint.

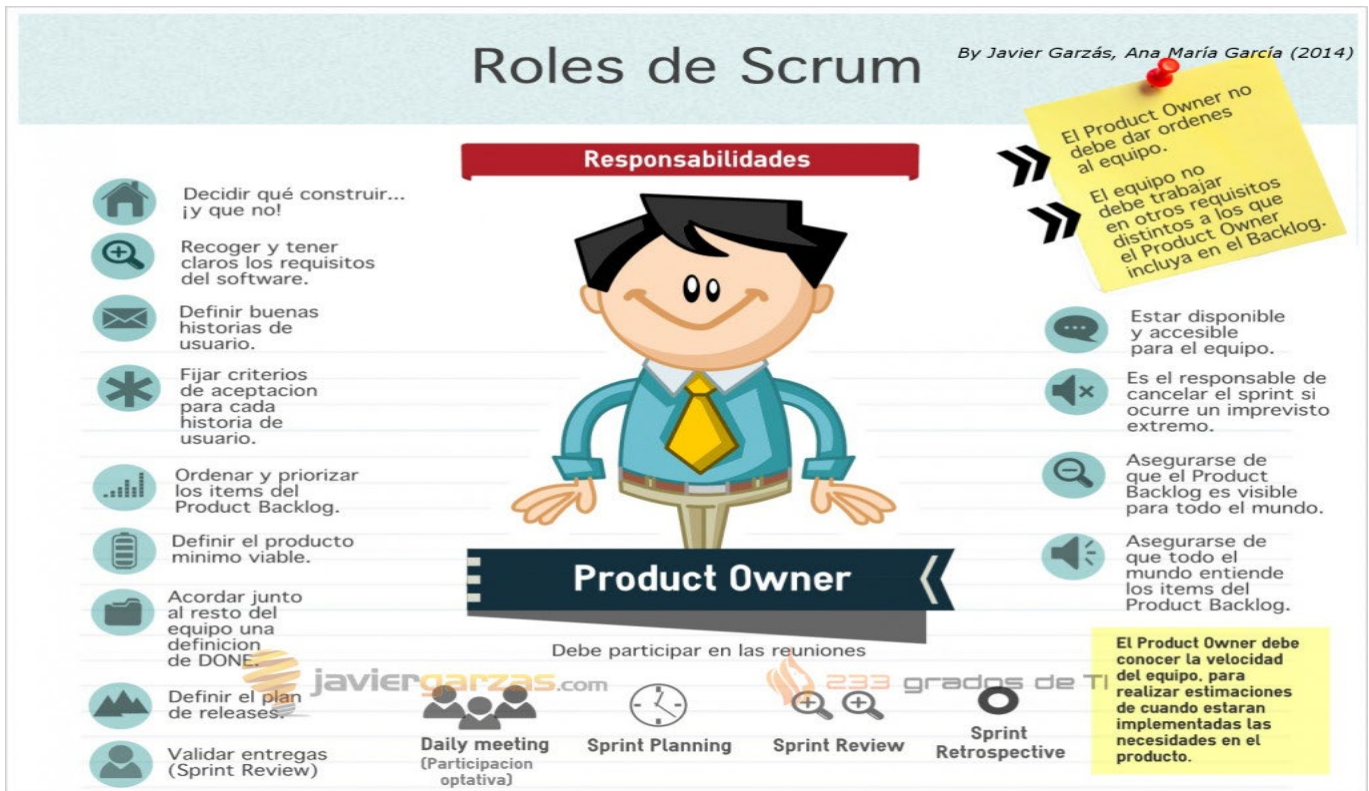
Ao final de cada sprint, na Reunión de Retrospectiva, o Product Owner aproba ou rexeita as características desenvolvidas. E tamén pode dar suxerencias para seguintes Sprints.

Finalmente, tamén é responsable de planificar as distintas entregas de produto, as distintas versións ou prototipos. El sabe como ten que ser o produto.

Ten que haber moita interacción co equipo e retro-alimentación mutua.

Roles de Scrum

By Javier Garz s, Ana Mar a Garc a (2014)



Team

Son as persoas que desenvolven o equipo, son responsables de dese ar, desenvolver e probar o produto que est n a implementar.

O Product Owner alimenta con informaci n ao equipo e aseg rase de que o equipo segue o cami o correcto e coa calidade adecuada.

O ScrumMaster lidera ao equipo para que este siga a metodolox a Scrum.

	Equipo de Desarrollo	Scrum Master	Product Owner	Stakeholders
Sprint	SI	SI	SI	
Planeaci�n del Sprint	SI	SI	SI	
Scrum Diario	SI	Opcional		
Revisi�n del Sprint	SI	SI	SI	SI
Retrospectiva del Sprint	SI	SI	Opcional	

Diferencias entre Scrum t cnico y Scrum avanzado

A diferenza principal entre Scrum t cnico e Scrum pragm tico ou avanzado,   que mentres Scrum

avanzado baséase en aplicar unha serie de valores para conseguir a axilidade no desenvolvemento, o técnico o que aplica son unha serie de regras para conseguir levar a cabo eses valores.

Scrum avanzado creouse en 1986, mentres que Scrum técnico é un marco de regras creado por dúas persoas en 1995 e enfocado no desenvolvemento de software.

Scrum avanzado baséase nestes valores:

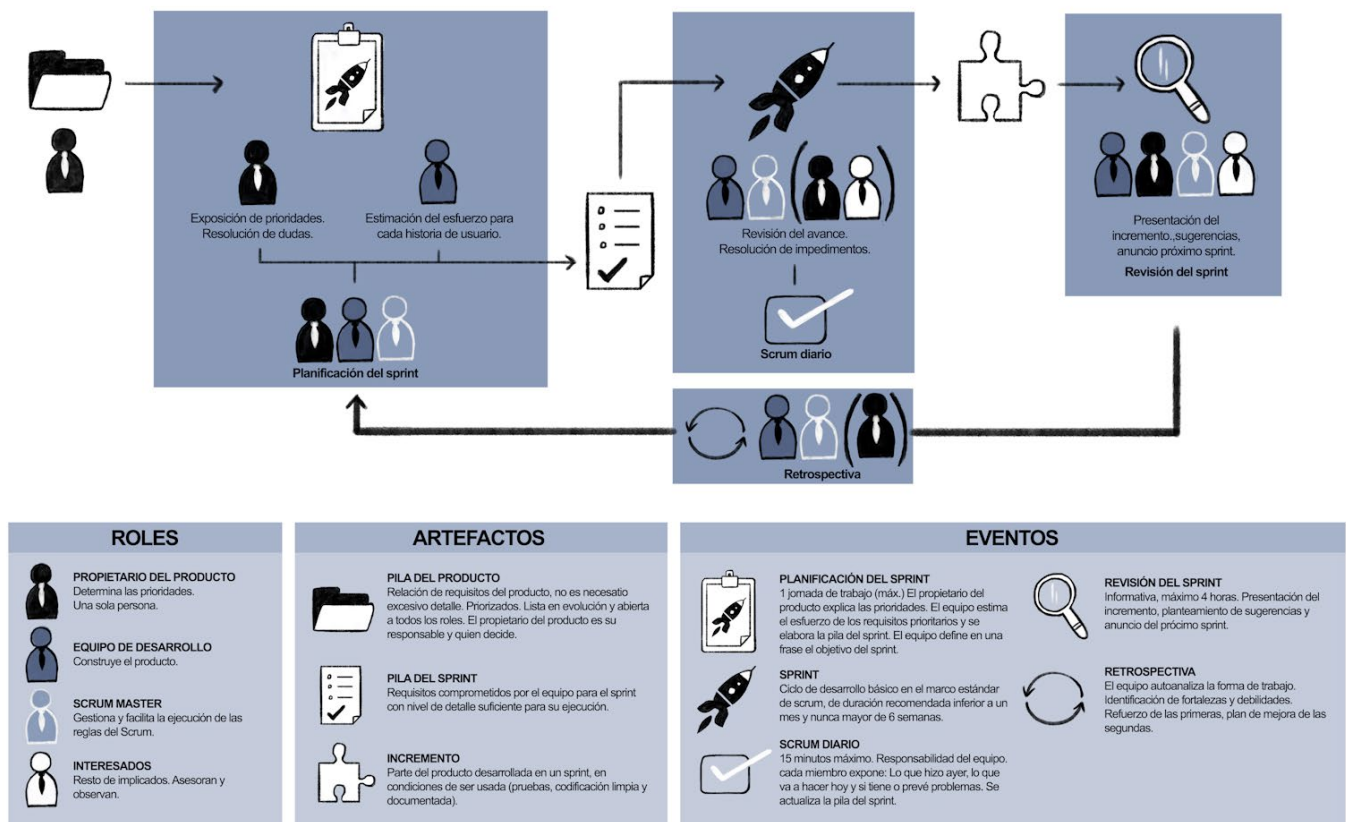
1. Prémíase máis o coñecemento das persoas que o funcionamento ou o valor dos procesos.
2. Dáse máis valor ao resultado que obteñamos que á documentación ou a burocracia pura.
3. Dáse máis valor á colaboración co cliente que á negociación.
4. Acéptanse os cambios fronte á planificación estrita.

Para conseguir estes valores con Scrum técnico defínense unha serie de:

1. Roles: Dono do produto, Scrum Máster, Equipo de desenvolvemento.
2. Eventos: Sprint, Reunión de planificación, Scrum diario, Revisión do Sprint, Retrospectiva do Sprint.
3. Artefactos: Pila do produto, Pila do Sprint, Incremento.

A idea principal é comezar realizando os nosos proxectos aplicando Scrum técnico, aplicando toda esa serie de regras que se propoñen, e unha vez que teñamos bastante dominio da metodoloxía e en que consiste, poderemos pasar a aplicar Scrum avanzado, eliminar esa serie de regras e continuar cos seus valores.

LAS REGLAS DE SCRUM



Últimas Consideracións

É válido Scrum para todo o mundo?

Definitivamente non. Algunhas persoas están demasiado afeitas á súa forma de traballar, e pode estar ben. Se o que estás a facer está funcionando ben pode que non haxa ningunha razón para cambiar o proceso. Non obstante, se observa que non se cumpren os requisitos, o produto final non é útil para o cliente, é posible que deba pensar máis sobre o uso de Scrum.

Cando non empregar Scrum?

Scrum non se debe usar en equipos pequenos, de ata tres persoas. Neste caso, a experiencia demostra que o marco non é tan útil por algunhas razóns. Os membros do equipo probablemente falan entre eles todo o tempo, facendo que as reunións diarias non teñan sentido. Ademais, dado que o equipo é bastante pequeno, as melloras son fáciles de detectar e adoitan implementarse en canto se identifican. Isto non significa, con todo, que non se deben empregar boas prácticas de desenvolvemento áxil. Só significa que non é necesario todo o marco Scrum.

Outra situación na que Scrum pode non ser necesario é cando o alcance do produto é ben coñecido e seguramente non cambiará. Ademais, o equipo é competente na tecnoloxía empregada, polo que non poden xurdir sorpresas por incertezas. Como se pode observar, este escenario é bastante raro. Non obstante, nesta situación específica, non son necesarios comentarios constantes, mellora continua do proceso nin cambios na priorización porque todo está ben determinado.

É Scrum perfecto, unha bala de prata?

Absolutamente non. Como se dixo antes, Scrum é moi difícil de poñer en práctica debido á interacción das persoas que poden xurdir. Ademais, é posible que as organizacións non estean abertas a todo o necesario para seguir Scrum.

Ademais, Scrum non resolve o problema das estimacións a longo prazo. Aínda é moi difícil estimar un proxecto longo con precisión, que aínda é un problema para os xestores que necesitan vender un produto.

Como comezar a empregar Scrum?

O mellor consello é facelo paso a paso, en pequenos proxectos preferentemente con membros acostumados a Scrum. Quizais comezar con Sprint, Daily Meetings e despois engadir comentarios. Probablemente estes mostrarán problemas no seu proxecto, como a comunicación e demasiados erros con funcións que xa se supoñía que estaban listas. As reunións diarias probablemente non serán tan curtas como se supoñía, polo que isto é algo que hai que solucionar.

Máis tarde, introducir unha retrospectiva sobre o proceso a mellorar. Despois de que o equipo estea acostumado para seguir todas as cerimonias de Scrum, probablemente o software en si sexa mellor. Non obstante, notará que haberá espazo para melloras, como probas automatizadas, mellor codificación e menos erros.

Isto non será doado: o cambio de mentalidade é enorme e hai xente que simplemente non se adapta a iso. Así, o importante é non frustrarse e tentar esforzarse en implementar Scrum se se pensa que paga a pena.

Ademais, hai que ter en conta que non adoptar todo o marco Scrum non é malo. O que é bo é o que

funciona para ti. Quizais na túa empresa a company ownership funcione mellor para unha persoa que para un equipo. Ou quizais Sprints teña que variar os seus marcos de tempo. O importante é empregar as ferramentas proporcionadas por Scrum para mellorar o seu proceso de desenvolvemento de software e facelo áxil.