

SCRUM

Libro: “Guía de Scrum”

Definición de Scrum

Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos.

El marco de trabajo Scrum es incompleto de manera intencional, solo define las partes necesarias para implementar la teoría de Scrum. Scrum se basa en la inteligencia colectiva de las personas que lo utilizan. En lugar de proporcionar a las personas instrucciones detalladas, las reglas de Scrum guían sus relaciones e interacciones.

Teoría de Scrum

Scrum se basa en el empirismo y el pensamiento Lean. El empirismo afirma que el conocimiento proviene de la experiencia y de la toma de decisiones con base en lo observado. El pensamiento Lean reduce el desperdicio y se enfoca en lo esencial. Emplea un enfoque iterativo e incremental para optimizar la previsibilidad y controlar el riesgo.

Scrum combina cuatro eventos formales para inspección y adaptación dentro de un evento contenedor, el Sprint. Estos eventos funcionan porque implementan los pilares empíricos de Scrum:

- Transparencia: el proceso y el trabajo emergentes deben ser visibles tanto para quienes realizan el trabajo como para quienes lo reciben. Los artefactos que tienen poca transparencia pueden llevar a decisiones que disminuyan el valor y aumenten el riesgo. La transparencia permite la inspección. Sin transparencia, es engañosa y derrochadora.
- Inspección: los artefactos de Scrum y el progreso hacia los objetivos acordados deben inspeccionarse con frecuencia y con diligencia para detectar variaciones o problemas potencialmente indeseables. La inspección permite la adaptación. La inspección sin adaptación se considera inútil.
- Adaptación: si algún aspecto de un proceso se desvía fuera de los límites aceptables o si el producto resultante es inaceptable, el proceso que se aplica o los materiales que se producen deben ajustarse. Se espera que un Team se adapte en el momento en que aprenda algo nuevo a través de la inspección.

Valores de Scrum

El uso exitoso de Scrum depende de que las personas se vuelvan más comprometidas en vivir cinco valores.

*“El Scrum Team se **compromete** a lograr sus objetivos y a apoyarse mutuamente. Su **foco** principal está en el trabajo del Sprint para lograr el mejor progreso hacia estos objetivos. El Scrum Team y sus interesados son **francos** sobre el trabajo y los desafíos. Los miembros del equipo del Scrum Team se **respetan** entre sí para ser personas capaces e independientes, y son respetados como tales por las personas con las que trabajan. Los miembros del Scrum Team tienen **coraje** de hacer lo correcto, para trabajar problemas difíciles.”*

Cuando el Scrum Team y las personas con las que trabajan incorporan estos valores, los pilares de Scrum cobran vida y generan confianza.

Scrum Team

Es una unidad cohesionada de profesionales enfocados en un objetivo a la vez, el objetivo del producto. Los Scrum Teams son multifuncionales, lo que significa que los miembros tienen todas las habilidades necesarias para crear valor en cada Sprint. También se autogestionan, lo que significa que deciden internamente quién hace qué, cuándo y cómo.

El Scrum Team es lo suficientemente pequeño como para seguir siendo ágil y lo suficientemente grande como para completar un trabajo significativo dentro de un Sprint, generalmente 10 personas o menos.

Scrum se define en tres responsabilidades específicas dentro del Scrum Team:

- Developers: las personas del Scrum Team que se comprometen a crear cualquier aspecto de un incremento utilizable en cada Sprint. Son responsables de:
 - Crear un plan para el Sprint, el Sprint Backlog
 - Inculcar calidad al adherirse a una definición de terminado

- Adaptar su plan cada día hacia el objetivo del Sprint
- Responsabilizarse mutuamente como profesionales.
- Product Owner: responsable de maximizar el valor del producto resultante del trabajo del Scrum Team. Es responsable de la gestión efectiva del Product Backlog, lo que incluye:
 - Desarrollar y comunicar explícitamente el objetivo del producto
 - Crear y comunicar claramente los elementos del Product Backlog
 - Ordenar los elementos del Product Backlog
 - Asegurarse de que el Product Backlog sea transparente, visible y se entienda

Para que los PO tengan éxito, toda la organización debe respetar sus decisiones.

- Scrum Master: responsable de lograr la efectividad del Scrum Team. Lo hace apoyando en la mejora de sus prácticas, dentro del marco de trabajo de Scrum.

Sirve al Scrum Team de varias formas:

- Guiar a los miembros del equipo en ser autogestionados y multifuncionales
- Ayudar al Scrum Team a enfocarse en crear incrementos de alto valor que cumplan con el DoD.
- Procurar la eliminación de impedimentos para el progreso del Scrum Team
- Asegurarse de que todos los eventos de Scrum se lleven a cabo y que sean productivos y se mantengan dentro de los límites de tiempo.

Sirve al PO de varias maneras:

- Ayudar a encontrar técnicas para una definición efectiva de objetivos del producto y la gestión del PB
- Ayudar al ST a comprender la necesidad de tener elementos del PB claros y concisos.
- Ayudar a establecer una planificación empírica de productos para un entorno complejo
- Facilitar la colaboración de los interesados según se solicite o necesite.

Sirve a la organización de varias maneras:

- Liderar, capacitar y guiar a la organización de su adopción de Scrum
- Planificar y asesorar implementaciones de Scrum dentro de la organización
- Ayudar a los empleados y los interesados a comprender y aplicar un enfoque empírico para el trabajo completo
- Eliminar las barreras entre los interesados y los ST.

Eventos de Scrum

El Sprint es un contenedor para todos los demás eventos. Cada evento es una oportunidad formal para inspeccionar y adaptar los artefactos Scrum. Estos eventos están diseñados específicamente para habilitar la transparencia requerida.

- Sprint

Son el corazón de Scrum, donde las ideas se convierten en valor. Todo el trabajo necesario para lograr el objetivo del producto, incluida la Sprint Planning, Daily Scrums, Sprint Review y Sprint Retrospective, ocurre dentro de los Sprints. Durante el Sprint:

- No se realizan cambios que pongan en peligro el objetivo del Sprint
- La calidad no disminuye
- El PB se refina según sea necesario
- El alcance se puede aclarar y renegociar con el PO a medida que se aprende más.

Se puede emplear Sprints más cortos para generar más ciclos de aprendizaje y limitar el riesgo de costo y esfuerzo a un período de tiempo menor. Cada Sprint puede considerarse un proyecto corto.

Existen varias prácticas para pronosticar el progreso, como el trabajo pendiente (burn-downs), trabajo completado (burn-ups) o flujos alternativos (cumulatives flows). Si bien han demostrado su utilidad, no reemplazan la importancia del empirismo.

Un Sprint podría cancelarse si el objetivo del Sprint se vuelve obsoleto. Solo el PO tiene la autoridad para cancelar el Sprint.

- Sprint Planning

Inicia el Sprint al establecer el trabajo que se realizará para el Sprint. El PO se asegura de que los asistentes estén preparados para discutir los elementos más importantes del PB y cómo se relacionan con el objetivo del producto. El ST también puede invitar a otras personas a asistir a la SP para brindar asesoramiento.

La SP aborda los siguientes temas:

- ¿Por qué es valioso este Sprint? El PO propone cómo el producto podría incrementar su valor y utilidad en el Sprint actual. Todo el ST colabora para definir un objetivo del sprint que comunica por qué el Sprint es valioso para los interesados.
- ¿Qué se puede hacer en este Sprint? A través de una conversación con el PO, los developers seleccionan elementos del PB para incluirlos en el Sprint actual.
- ¿Cómo se realizará el trabajo elegido? Para cada elemento del PB seleccionado, los developers planifican el trabajo necesario para crear un incremento que cumpla con la DoD. A menudo, esto se hace descomponiendo los elementos del PB en elementos de trabajo más pequeños de un día o menos.
El objetivo del Sprint, los elementos del PB seleccionados para el Sprint, más el plan para entregarlos se denominan juntos SB.

- Daily Scrum

El propósito de la DS es inspeccionar el progreso hacia el objetivo del Sprint y adaptar el SB según sea necesario, ajustando el trabajo planificado entrante. Es un evento de 15 minutos para los developers del ST.

Los developers pueden seleccionar la estructura y las técnicas que deseen, siempre que su DS se centre en el progreso hacia el objetivo del sprint.

Las DS mejoran la comunicación, identifican impedimentos, promueven la toma rápida de decisiones y, en consecuencia, eliminan la necesidad de otras reuniones.

La DS no es el único momento en el que los developers pueden ajustar su plan. A menudo se reúnen durante el día para discusiones más detalladas sobre cómo adaptar o volver a planificar el resto del trabajo del Sprint.

- Sprint Review

Durante el evento, el ST y los interesados revisan lo que se logró en el sprint y lo que ha cambiado en su entorno. El PB también se puede ajustar para satisfacer nuevas oportunidades. Tiene un límite de tiempo máximo de cuatro horas para un sprint de un mes.

- Sprint Retrospective

El propósito es planificar formas de aumentar la calidad y la efectividad. El ST analiza qué salió bien durante el Sprint, qué problemas ocurrieron y cómo se resolvieron (o no).

Las mejoras más impactantes se abordan lo antes posible. Incluso se pueden agregar al SB para el próximo sprint.

La SR concluye el Sprint. Tiene un tiempo limitado a máximo tres horas para un sprint de un mes.

Artefactos de Scrum

Representan trabajo o valor. Están diseñados para maximizar la transparencia de la información clave.

- Product Backlog

El Product Backlog es una lista emergente y ordenada de lo que se necesita para mejorar el producto. Es la única fuente del trabajo realizado por el Scrum Team.

Los elementos del Product Backlog que el Scrum Team puede dar por Terminados dentro de un Sprint se consideran preparados para ser seleccionados en un evento de Sprint Planning. Suelen adquirir este grado de transparencia tras las actividades de refinamiento. El refinamiento del Product Backlog es el acto de dividir y definir aún más los elementos del Product Backlog en elementos más pequeños y precisos. Esta es una actividad continua para agregar detalles, como una descripción, orden y tamaño. Los atributos suelen variar según el ámbito del trabajo.

El objetivo del producto está en el PB. El resto del PB emerge para definir qué cumplirá con el objetivo del producto. El OP es el objetivo a largo plazo del ST. Ellos deben cumplir (o abandonar) un objetivo antes de asumir el siguiente.

- Sprint Backlog

El SB se compone del objetivo del sprint (por qué), el conjunto de elementos del PB seleccionados para el Sprint (qué), así como un plan de acción para entregar el incremento (cómo).

Es una imagen muy visible y en tiempo real de trabajo que los developers planean realizar durante el sprint para lograr el objetivo del sprint. En consecuencia, el SB se actualiza a lo largo del Sprint a medida que se aprende más.

El objetivo del sprint se crea durante el evento del SP y se agrega al SB. Mientras los developers trabajan durante el sprint, tienen en mente el objetivo del sprint. Si el trabajo resulta ser diferente de lo que esperaban, colaboran con el PO para negociar el alcance del SB dentro del sprint sin afectar el OS.

- Incremento

Un Incremento es un peldaño concreto hacia el Objetivo del Producto. Cada Incremento se suma a todos los Incrementos anteriores y se verifica minuciosamente, lo que garantiza que todos los Incrementos funcionen juntos. Para proporcionar valor, el Incremento debe ser utilizable.

Se pueden crear múltiples Incrementos dentro de un Sprint. La suma de los Incrementos se presenta en la Sprint Review apoyando así el empirismo.

El trabajo no puede considerarse parte de un incremento a menos que cumpla con el DoD.

La DoD es una descripción formal del estado del incremento cuando cumple con las medidas de calidad requeridas para el producto. En el momento que un elemento del PB cumple con esto, nace un incremento.

Si un elemento del PB no cumple con la DoD, no se puede publicar ni presentar en la SR. En su lugar, vuelve al PB para su consideración futura.

Libro: “*Scrum y XP desde las trincheras*”

Capítulo 2: Cómo hacemos Pilas de Producto (PB)

La PB es, básicamente una lista priorizada de requisitos, o historias, o funcionalidades, o lo que sea. Cosas que el cliente quiere, descritas usando la terminología del cliente. Nuestras historias incluyen los siguientes campos:

- ID: identificador único, simplemente un número auto-incremental. Nos permite no perder la pista a las historias cuando cambiamos su nombre.
- Nombre: descripción corta de la historia.
- Importancia: el ratio de importancia que el PO da a esta historia.
- Estimación inicial: la valoración inicial del equipo acerca de cuanto trabajo es necesario para implementar la historia, comparada con otras historias. Las unidades son “SP”.
- Cómo probarlo: una descripción a alto nivel de cómo se demostrará esta historia en la demo al final del sprint. Si hacemos TDD esta descripción puede usarse como pseudo-código para nuestro test de aceptación.
- Notas: cualquier otra información, clarificación, referencia a otras fuentes de información, etc. Normalmente muy breve.

Oficialmente, el PO es el propietario del documento, pero no queremos dejar al resto de usuarios fuera. No colocamos este documento en el repositorio de control de versiones; en vez de eso, lo almacenamos en una unidad de red compartida.

Capítulo 3: Cómo nos preparamos para la planificación de Sprint

Asegúrate de que la PB está perfectamente lista antes de la reunión de planificación de Sprint. Esto significa que:

- Debería haber una PB y un PO (por producto)
- Todos los elementos importantes deberían tener ratios de importancia asignados, diferentes ratios de importancia.
- El PO debe comprender cada historia. No necesita saber cómo exactamente debe implementarse, pero debería entender por qué la historia está ahí.

Otras personas aparte del PO pueden añadir sus historias a la PB. Pero no pueden asignarles niveles de importancia, ese es un cometido exclusivo del PO. Tampoco pueden establecer estimaciones, ese un cometido exclusivo del equipo.

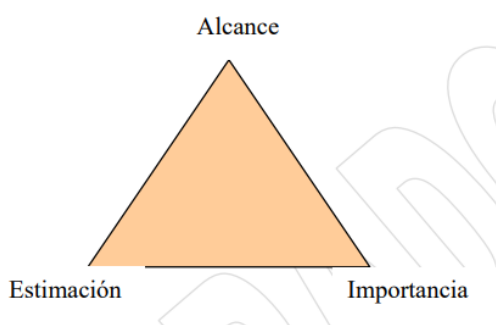
Capítulo 4: Cómo hacemos la planificación de Sprint

Una planificación de Sprint produce concretamente:

- Una meta de Sprint.
- Una lista de miembros
- Una SB
- Una fecha concreta para la demo del Sprint
- Un lugar y momentos definidos para el Scrum Diario.

Por qué debe asistir el PO

La razón por la que el equipo y el PO deben asistir a la planificación de Sprint es que cada historia contiene tres variables que son muy dependientes unas de otras.



El alcance y la importancia los fija el PO. La estimación proporciona el equipo. Durante una planificación de Sprint, estas variables sufren un ajuste fino y continuo a través del diálogo cara a cara entre el equipo y el PO.

Normalmente, el PO comienza la reunión resumiendo cuál es su meta para el Sprint y las historias más importantes. A continuación, el equipo las repasa y les asigna una estimación, comenzando con la más importante.

¿Qué ocurre si el PO insiste en que no tiene tiempo de unirse a las reuniones de planificación de Sprint? Usualmente intento alguna de las siguientes estrategias, en el siguiente orden:

- Intentar que el PO comprenda por qué su participación es crucial y esperar que cambie de parecer.
- Intentar que alguien del equipo se presente voluntario como delegado del PO durante la reunión.
- Tratar de convencer a la gerencia de que asigne un nuevo PO.
- Posponer el lanzamiento del Sprint hasta que el PO encuentre el tiempo para asistir a la reunión.

Por qué la calidad no es negociable

Existe lo que se denomina:

- Calidad externa: es lo que perciben los usuarios del sistema.
- Calidad interna: se refiere a aquellos aspectos que normalmente no son visibles al usuario, pero que tienen un profundo efecto en la mantenibilidad del sistema.

Generalizando, un sistema con alta calidad interna puede, aun así, tener una baja calidad externa. Pero un sistema con baja calidad interna rara vez tendrá una buena calidad externa.

La calidad interna es algo que no puede ser discutido. Es responsabilidad del equipo mantener la calidad del sistema bajo toda circunstancia y simplemente no es negociable.

El tiempo que se ahorra es mucho menor que el coste, tanto a corto como a largo plazo. Una vez que permites que una base de código comience a deteriorarse es muy duro volver a conseguir su calidad original más adelante.

Reuniones de planificación de Sprint que duran, y duran...

Todo en Scrum tiene una duración determinada (time-boxed). Me encanta esa única, simple y consistente regla. Intentamos cumplirla al cien por cien. ¿Qué hacemos cuando la reunión de planificación de Sprint está llegando al final y no hay señales de una meta de Sprint o SP?

Acabo con la reunión. Sí, el Sprint sufre. La ventaja, en cualquier caso, es que el equipo ha aprendido una lección muy valiosa, y en la próxima reunión de planificación de Sprint serán mucho más eficientes. Adicionalmente, ofrecerán menos resistencia cuando propongas una duración para las reuniones que anteriormente hubieran considerado excesiva.

Aprender a mantener tus duraciones determinadas, aprende a establecer duraciones realistas. Esto aplica tanto a las reuniones como a los Sprints.

Definiendo la duración del Sprint

Bueno, los Sprints cortos están bien. Permiten a la compañía ser “ágil”, es decir, cambiar de dirección frecuentemente. Sprints cortos = ciclo de feedback corto = más entregas y más frecuentes = más feedback del cliente = menos tiempo desarrollando en dirección incorrecta = aprender y mejorar más rápido, etc.

Pero los Sprints largos tampoco están mal. El equipo tiene más tiempo para conseguir impulso, tienen más espacio para recuperarse de los problemas que surjan y aun así cumplir la meta del Sprint, tiene menos carga de gestión en términos de reuniones de planificación de Sprints, Demos, etc.

Así que la duración del Sprint es un valor de compromiso. Hemos experimentado con varias duraciones y al final encontramos nuestra duración favorita: 3 semanas.

Definiendo la meta del Sprint

La meta de Sprint debería responder a la pregunta fundamental “¿Por qué hacemos este Sprint en vez de irnos todos de vacaciones?”

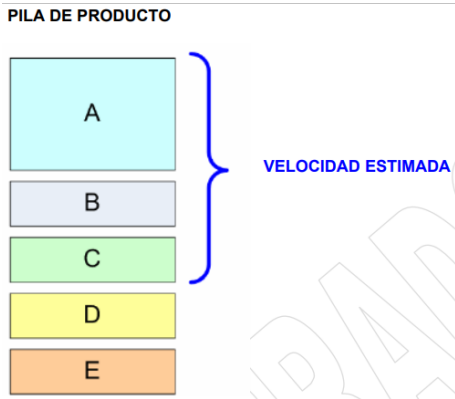
La meta del Sprint puede parecer bastante tonta y artificial durante la planificación de Sprint, pero muchas veces resulta útil a mediados de Sprint, cuando la gente comienza a sentirse confusa acerca de lo que deberían estar haciendo.

Decidiendo qué historias incluir en el Sprint

Una de las principales actividades durante la planificación de Sprint es decidir qué historias se incluyen en el Sprint. Más específicamente, qué historias de la PB copiar en la SB.

¿Cómo puede el PO alterar las historias que se incluyen en el Sprint?

Supongamos que tenemos la siguiente situación durante una reunión de planificación de Sprint.



Al PO no le gusta que la historia D no se vaya a incluir en el Sprint. ¿Cuáles son sus opciones durante la reunión de planificación de Sprint? El PO no puede controlar la velocidad estimada, cabe aclarar.

Una es re-priorizar. Si le da al elemento D la mayor importancia, el equipo se verá obligado a añadirlo al Sprint en primer lugar.

Opción 1:



La segunda opción es cambiar el alcance – reducir el alcance de la historia A hasta que el equipo crea que la historia D podría caber en el Sprint.

Opción 2:



La tercera es dividir una historia.

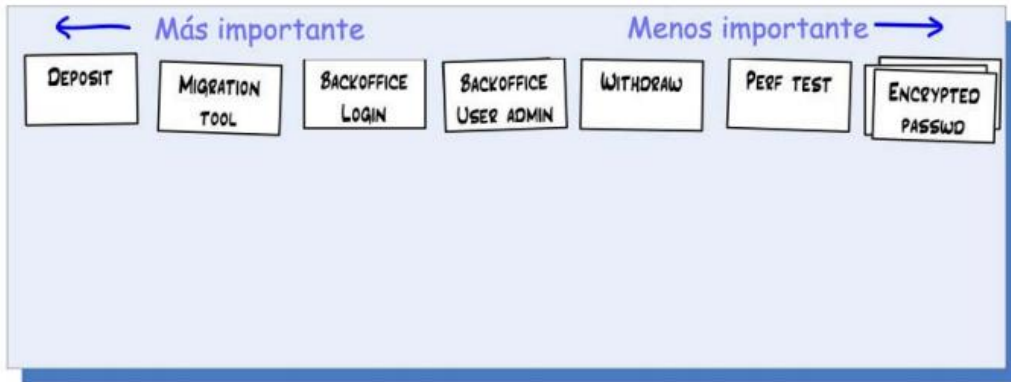
Opción 3:



Por qué usamos tarjetas

Este es un interfaz muy superior, comparado con un ordenador y un proyector, debido a que:

- La gente se pone de pie y camina alrededor, se mantienen más despiertos y alertas.
- Todo el mundo se siente personalmente más involucrado
- Se pueden editar múltiples historias simultáneamente
- Repriorizar es trivial, simplemente se trata de mover las tarjetas.
- Tras la reunión, las tarjetas pueden trasladarse directamente a la sala de equipo y usarse como un tablón de tareas en la pared.

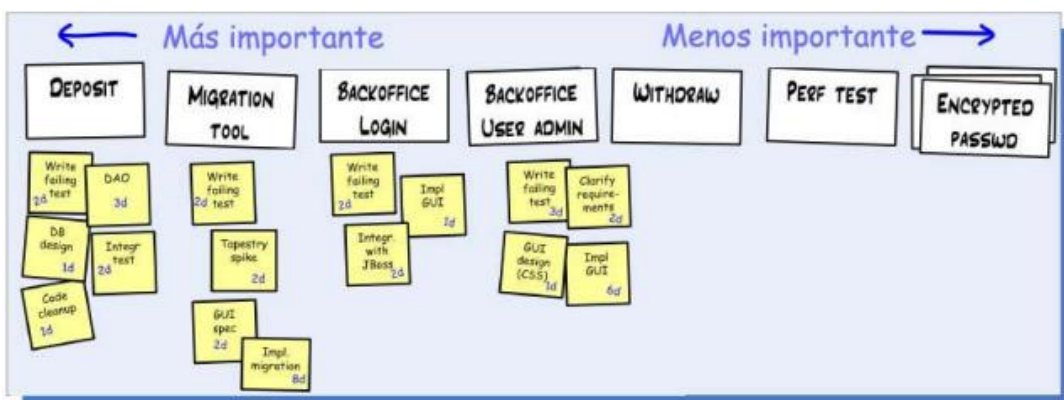


Elemento de Pila	Importancia
Depósito	30
Notas	Estimación
Necesita un diagrama secuencial UML. No hay que preocuparse aún por el cifrado.	
Como probarlo	
Entrar, abrir la página de depósito, depositar 10€, ir a la página de balance y comprobar que se ha incrementado en 10€	

Una nota sobre el campo importancia: se trata de la importancia tal y como figura en la PB en Excel en el momento de la impresión. Tenerla en las tarjetas hace que sea más sencillo ordenarlas físicamente por importancia.

Las estimaciones de tiempo son normalmente más fáciles de hacer (y más exactas) si una historia se subdivide en tareas.

Esto es algo que también se hace de forma sencilla y agradable con las tarjetas de historia. Puedes hacer que el equipo se divida en parejas y que cada una subdivide una historia en paralelo. Físicamente, hacemos esto añadiendo pequeñas notas post-it bajo cada historia, de forma que cada una de ellas representa una tarea dentro de dicha historia.



No actualizamos la PB en Excel respecto a nuestra división de tareas por dos razones:

- La división en tareas suele ser bastante volátil, es decir, se cambia y se refina con frecuencia durante el Sprint, así que es bastante molesto mantener la PB en Excel sincronizada.
- De todas formas, el PO no necesita estar involucrado a ese nivel de detalle.

Definición de Terminado (DoD)

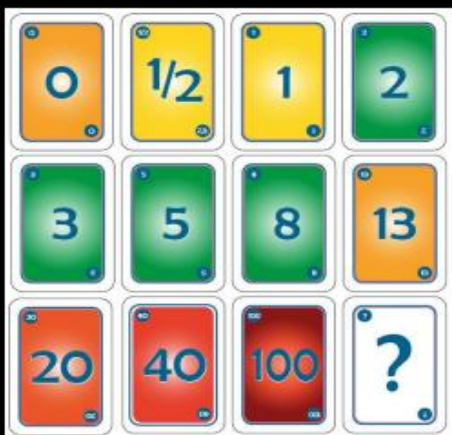
Es importante que el PO y el equipo estén de acuerdo en una definición clara de “terminado”. Siempre que es posible, utilizamos la “lista para pasar a producción”, aunque a veces debemos conformarnos con “instalado en preproducción y lista para las pruebas de aceptación”.

Estimación de tiempos usando planning poker

La estimación es una labor de equipo: todos los miembros del equipo deben involucrarse en estimar cada historia. ¿Por qué?

- A la hora de planificar, normalmente no sabemos exactamente quién implementará qué partes de cada historia.
- Las historias normalmente involucran a bastantes personas y de diferentes áreas de experiencia.
- Para poder proporcionar una estimación, un miembro del equipo necesita comprender de alguna forma de qué trata la historia. Pidiendo a todo el mundo que estime la historia nos aseguramos que cada miembro del equipo comprende de qué trata cada elemento.

¿Cómo “decodificar” las estimaciones?



- **0:** Quizás ud. no tenga idea de su producto o funcionalidad en este punto.
- **1/2, 1:** funcionalidad pequeña (usualmente cosmética).
- **2-3:** funcionalidad pequeña a mediana. Es lo que queremos. 😊
- **5:** Funcionalidad media. Es lo que queremos 😊
- **8:** Funcionalidad grande, de todas formas lo podemos hacer, pero hay que preguntarse sino se puede partir o dividir en algo más pequeño. No es lo mejor, pero todavía 😊
- **13:** Alguien puede explicar por que no lo podemos dividir?
- **20:**Cuál es la razón de negocio que justifica semejante story y más fuerte aún, por qué no se puede dividir?
- **40:** no hay forma de hacer esto en un sprint.
- **100:** confirmación de que está algo muy mal. Mejor ni arrancar.

Poker Planning

Prerrequisitos:

- Lista de features/stories a ser estimadas
- Cada estimador tiene un mazo de cartas.

Pasos:

1. Determine la **base story** (la canónica) que será usada para comprar con las otras stories. Digamos, Story Z.
 - 1.1 La story a ser estimada se lee a todo el equipo.
 - 1.2 Los estimadores discuten la story, haciendo preguntas al product owner (las que se necesiten).
 - 1.3 Cada estimador selecciona una carta y pone la carta boca abajo en la mesa.
 - 1.4 Cuando todos pusieron las cartas, las mismas se exponen al mismo tiempo.
 - 1.5 Si todos los estimadores selecciona el mismo valor, ese es el estimado. Sino, los estimadores discuten sus resultados, poniendo especial atención en los más altos y los más bajos. Después de la charla, GOTO to 1.3.
2. Se toma la próxima story, se discute con el product owner.
3. Cada estimador asigna a la story un valor por comparación contra la base story. “Cuan grande/pequeña, compleja, riesgosa es esta story comparada con Story Z?”. GOTO to 1.3



Dividiendo las historias en tareas

¿Cuál es la diferencia entre historias y tareas? La diferencia es muy simple. Las historias son entregables de los que el PO se preocupa. Las tareas son no entregables, o aspectos que el PO no se preocupa.

Este tipo de división frecuentemente revela trabajo adicional que hace que las estimaciones suban, con lo que se consigue que sea más realista. Hace que los Scrums diarios sean notablemente más eficientes.

Definiendo el sitio y la hora para el Scrum Diario

Uno de los productos frecuentemente olvidados en la planificación de Sprint es “un sitio y una hora determinados para el Scrum diario”. Sin ello, tu Sprint está condenado a un mal comienzo. El primer Scrum diario es esencialmente el lanzamiento, donde todo el mundo decide por dónde va a empezar a trabajar.

La desventaja de Scrums por las tardes es que cuando llegues al trabajo por la mañana tienes que acordarte de qué le dijiste a la gente sobre lo que deberían hacer hoy.

La desventaja de Scrums por las mañanas es que cuando llegas al trabajo por la mañana, debes acordarte de qué hiciste ayer para informar sobre ello hoy.

Historias técnicas

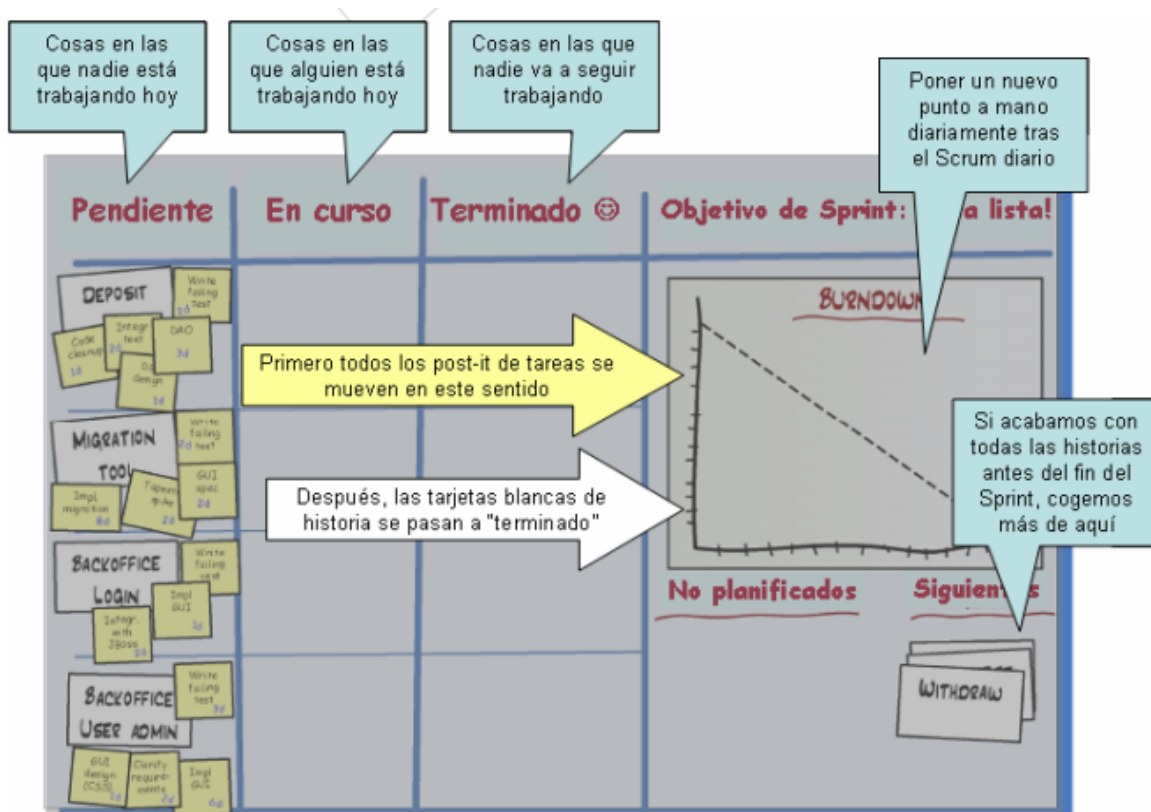
Cosas que deben hacerse pero que no son un entregable ni están directamente relacionadas con ninguna historia específica, y no son de valor inmediato para el PO son llamadas historias técnicas. Son historias con mínima prioridad.

Esto hacemos:

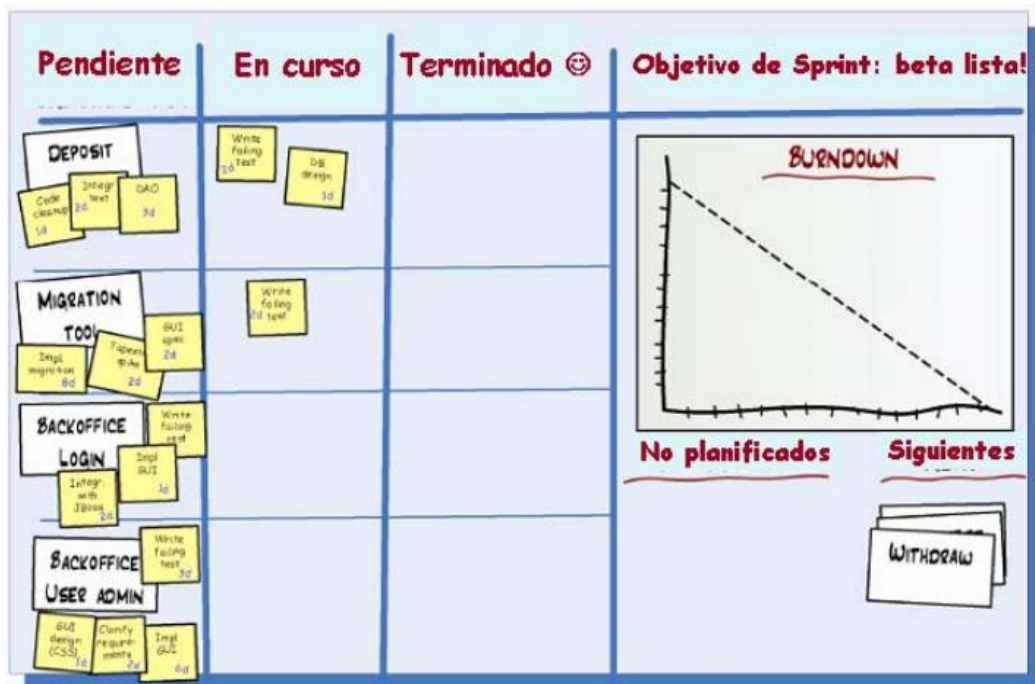
- 1) Intentamos evitar historias técnicas. Busca efusivamente formas de transformar las historias técnicas en historias normales con valor de negocio medible.
- 2) Si no podemos transformar una historia técnica en una normal, intentamos ver si es posible hacerla como una tarea dentro de otra historia.
- 3) Si lo anterior falla, definirla como historia técnica y mantener una lista separada con dichas historias. Permitimos al PO que vea dicha lista, pero no que la modifique.

Unidad 6: Cómo hacemos Pilas de Sprint (SB)

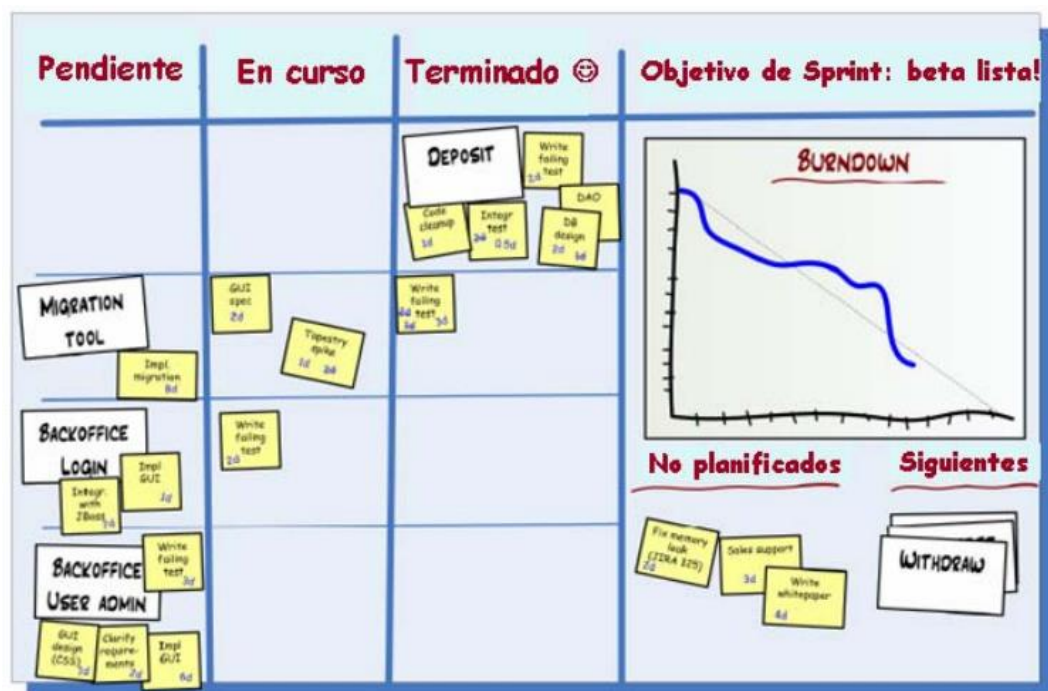
Formato de SB



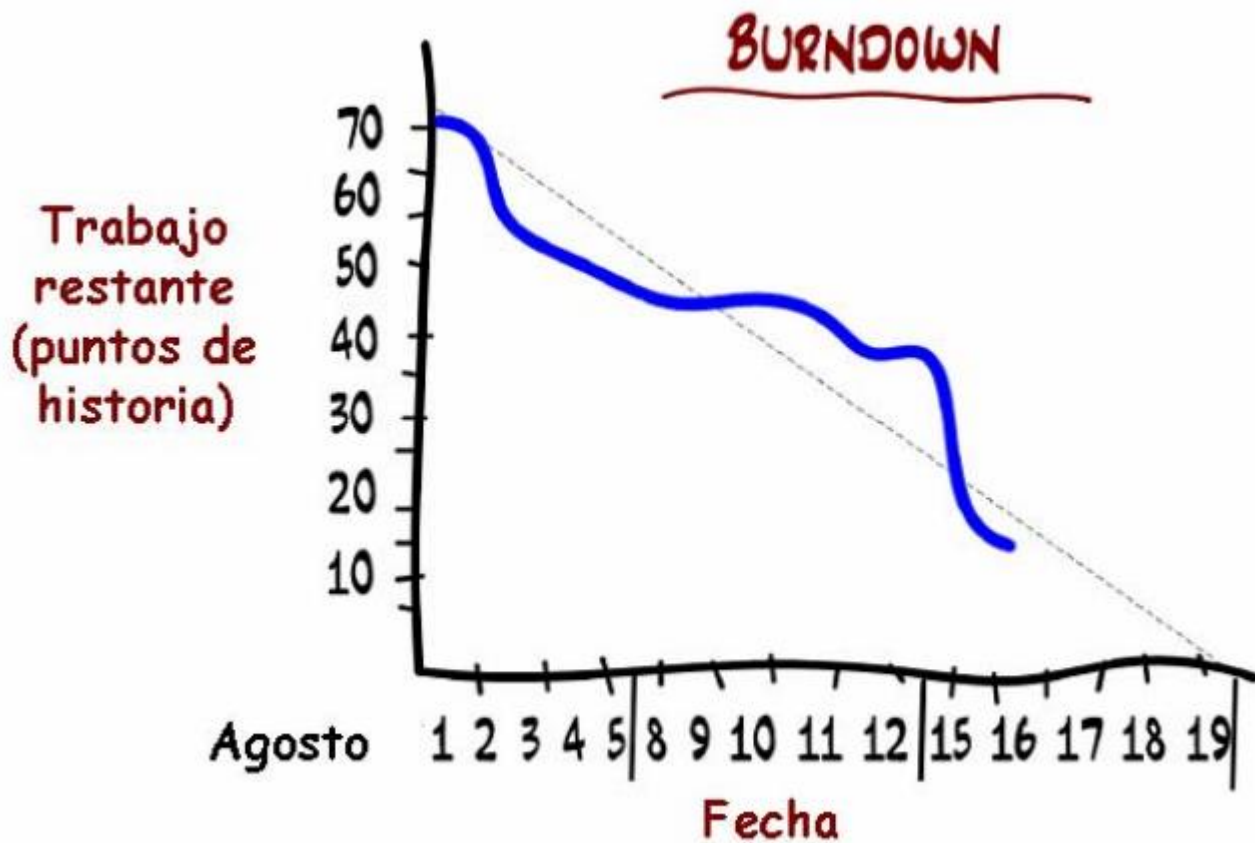
Tras un Scrum diario:



Tras varios días:



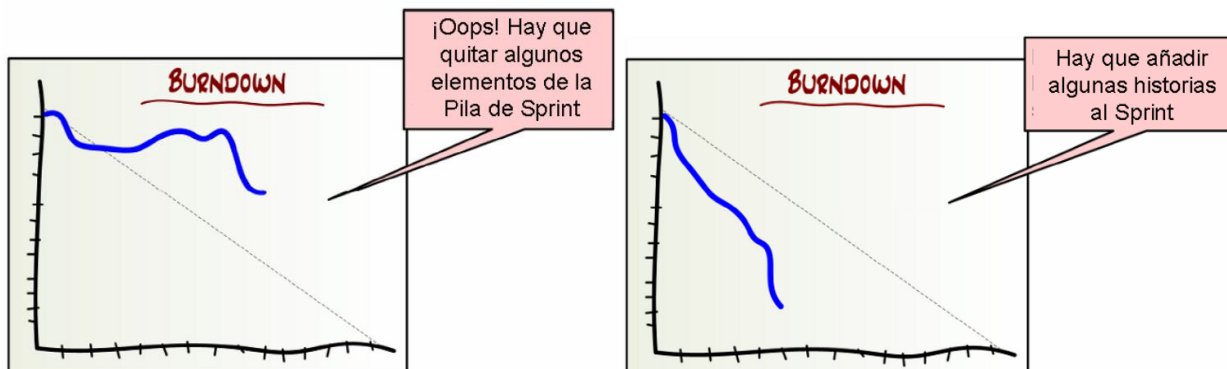
Cómo funciona el diagrama burn-down

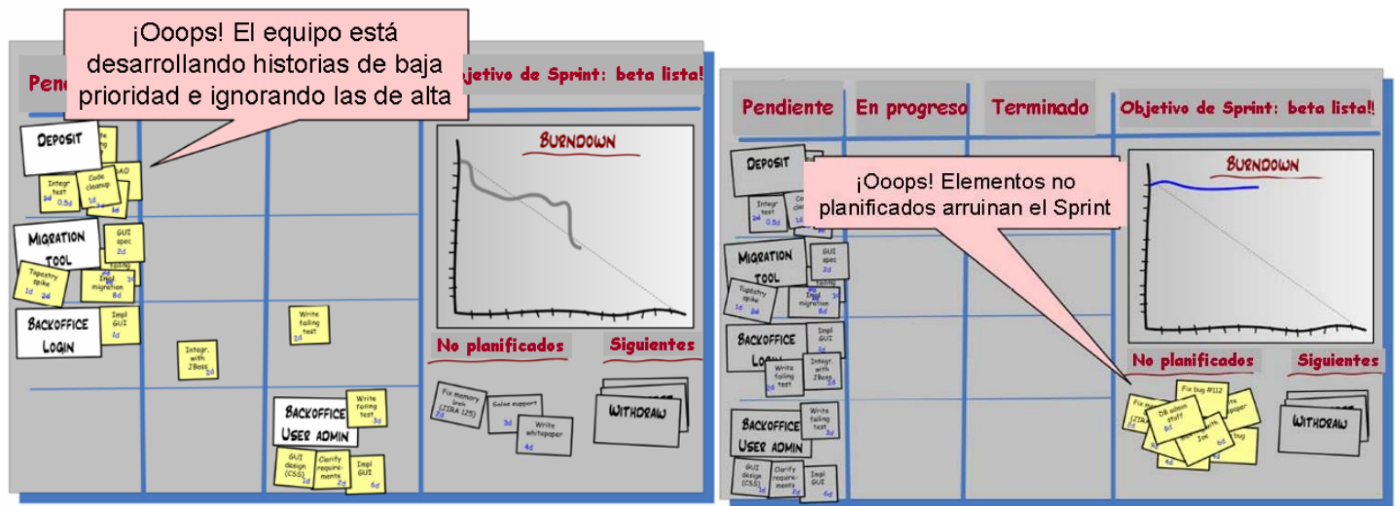


Este diagrama muestra que:

- En el primer día del Sprint, 1 de agosto, el equipo estimó que habían aproximadamente 70 puntos de historia en los que trabajar. Esta era consecuentemente, la velocidad estimada para todo el sprint.
- El 16 de agosto el equipo estima que quedan aproximadamente 15 puntos de historia por hacer. La línea de puntos nos muestra que estamos incluso algo avanzados respecto a la planificación, es decir, que a este paso completáramos todo al final del sprint.

El Scrum Master es responsable de asegurarse de que el equipo actúa ante señales de alarma como:





Unidad 7: Cómo distribuimos la sala de equipo

Cuando se trata de asientos y disposición de mesas, hay una cosa en la que nunca puede hacerse demasiado énfasis: Sentar al equipo junto. Cuanto menos sea la distancia, más resistencia.

Una vez tengas al equipo todo junto, los resultados serán inmediatos. Tras un solo Sprint el equipo estará de acuerdo en que fue una buena idea sentarse juntos.

“Juntos” significa:

- Audibilidad: cualquier miembro del equipo puede hablar con cualquier otro sin tener que levantarse.
- Visibilidad: todo el mundo puede verse. Todos pueden ver el tablón de tareas.
- Aislamiento: si todo el equipo necesitase levantarse y agruparse para una animada y espontánea discusión sobre el diseño, nadie fuera del equipo está tan cerca como para se molesto. Y viceversa.

Manten al PO a mano

El PO debería estar suficientemente cerca como para que el equipo pueda caminar hasta donde está y preguntarle algo, y para que él pueda acercarse también al tablón de tareas. Pero no debería sentarse junto al equipo, porque lo más probable es que no sea capaz de controlarse e interrumpa constantemente entrometiéndose en cualquier detalle, y el equipo no podrá fluir adecuadamente.

Manten a los gerentes y coachs a mano

Mi punto de vista es: Si eres un coach en Scrum (y quizás también un gerente), involúcrate lo máximo posible. Pero sólo por un periodo de tiempo limitado. Entonces quítate de en medio y deja que sea el equipo el que se autogestione. Chequea el equipo de vez en cuando (no demasiado a menudo) atendiendo a las demos de Sprint, mirando el tablón de tareas y escuchando en los Scrum diarios.

Si ves un área de mejora, reúnete con el Scrum Master y ayúdalo. No en frente de todo el equipo. Otra Buena idea es atender a las retrospectivas de los Sprints (ver “cómo hacemos retrospectivas de Sprint”), si tu equipo confía suficiente en ti como para que tu presencia les haga cerrarse en banda.

Unidad 8: Cómo hacemos Scrum diarios

Cómo actualizamos el tablón

Normalmente actualizamos el tablón de tareas durante los Scrum diarios. Conforme cada persona describe lo que hizo el día anterior y lo que hará hoy, mueve los post-it en el tablón. Conforme describe elementos no planificados, pone un pos-it nuevo para cada uno de ellos. Conforme actualiza sus estimaciones, escribe una nueva estimación en el post-it correspondiente y tacha la anterior estimación. A veces el Scrum Master hace todo esto mientras los demás hablan.

Intenta involucrar a todo el equipo en la labor de mantener la SB actualizada. Hemos intentado hacer Sprints en los que el SM es el único que mantiene la SB y las desventajas de este enfoque son:

- El SM pasa demasiado tiempo administrando asuntos, en vez de dar soporte al equipo y eliminando impedimentos.
- Los miembros del equipo no están al corriente del estado del Sprint, ya que la SB es algo de lo que no necesitan preocuparse. Esta falta de feedback reduce la agilidad y el enfoque generales del equipo. Si la SB está bien diseñada debería ser igual de fácil para cada miembro del equipo actualizarla el mismo.

Tratando con tardones

Algunos equipos tienen una lata de monedas y billetes. Cuando llegas tarde, incluso aunque sea sólo por un minuto, añades una cantidad prefijada en la lata. Sin preguntas. Si llamas antes de la reunión y avisas de que vas a llegar tarde, aun así tienes que pagar. Solo te salvas de la multa si tienes una buena excusa como una cita con el médico, tu propia boda o algo similar.

Tratando con “no sé qué hacer hoy”

Digamos que Joe y Lisa son los que no saben qué hacer hoy. Si yo soy el Scrum Master simplemente sigo con la reunión y dejo que el siguiente miembro del equipo hable, pero anoto quienes son las personas que no saben qué hacer. Después de que todo el mundo haya hablado, repaso el tablón de tareas con todo el equipo, de arriba a abajo, y me aseguro de que todo esté sincronizado, que todo el mundo sabe lo que significa cada elemento, etc. Invito a todo el mundo a añadir más post-its. Entonces vuelvo a las personas que no sabían qué hacer y les digo “ahora que hemos repasado el tablón de tareas, ¿tenéis idea de lo que podéis hacer hoy?”. Con algo de suerte será así. Si no, considera si no tienes aquí una buena oportunidad para hacer programación por parejas.

Joe y Lisa siguen negándose a especificar algo útil en lo que vayan a trabajar. Normalmente considero alguna de las siguientes estrategias:

- Vergüenza: “Bueno, sino tenéis idea de cómo podéis ayudar al equipo os sugiero que os vayáis a casa, o leáis un libro o algo. O simplemente sentaros por aquí hasta que alguien del equipo os pida ayuda”.
- Vieja escuela: simplemente asígnales una tarea.
- Presión de los compañeros: Di “sentiros libres de tomaros el tiempo que haga falta, Joe y Lisa, todos los demás simplemente nos quedaremos aquí de pie hasta que se os ocurra algo que pueda ayudarnos a conseguir el objetivo del Sprint”.
- Servidumbre: Di algo como “Bueno, podéis ayudar al equipo indirectamente siendo sus mayordomos hoy. Traedles café, dadles un masaje, limpiad sus mesas, preparadles algo de comida y cualquier cosa que os pidan durante el día”. Os sorprenderá lo rápido que Joe y Lisa encontrarán algunas tareas técnicas de provecho.

Si una persona te fuerza constantemente a ir tan lejos, probablemente deberías coger a esa persona en un aparte y hacer algo de coaching en serio. Si el problema continúa, necesitas evaluar si esa persona es importante para el equipo o no.

Si no es demasiado importante, intenta que lo aparten de tu equipo. Si es demasiado importante, entonces intenta emparejarlo con alguien que pueda actuar como su guía.

Unidad 9: Cómo hacemos la demo de Sprint

Por qué insistimos en que todos los Sprints acaben con una demo

Los efectos que producen son:

- El equipo obtiene reconocimiento por sus logros, se sienten bien.
- Otras personas se enteran de lo que está haciendo el equipo
- La demo consigue feedback vital de los interesados.
- Las demos son un evento social donde diferentes equipos pueden interactuar unos con otros y discutir su trabajo. Esto es muy valioso.
- Hacer una demo fuerza al equipo a acabar realmente las cosas y entregarlas.
- Si un equipo se ve obligado a hacer una demo de sprint, incluso aunque no tenga mucho que realmente esté funcionando, la demo será embarazosa. El equipo sabe que tendrán que hacer una demo pase lo que pase, lo que incrementa significativamente las posibilidades de que haya algo útil que demostrar.

Lista de comprobación para demos de Sprint

- Asegúrate de presentar claramente el objetivo del Sprint. Si hay personas en la demo que no saben nada sobre tu producto, tomate un par de minutos para describirlo.
- No pierdas mucho tiempo preparando la demo, especialmente en llamativas presentaciones. Déjate de milongas y concéntrate mostrar código funcionando.
- Mantén el paso rápido, es decir, concentra tu preparación en hacer que la demo sea rápida en lugar de bonita.
- Mantén la demo a nivel de negocio, deja los detalles técnicos aparte. Concéntrate en qué hemos hecho en lugar de como lo hemos hecho.
- En medida de lo posible, deja que la audiencia pruebe el producto en sí misma.
- No muestres un montón de pequeños errores solucionados y funcionalidades triviales. Menciónalos, pero no los muestres, ya que normalmente se tarda mucho y desvía la atención de las historias importantes.

Unidad 10: Cómo hacemos retrospectivas de Sprint

Cómo organizamos las retrospectivas

El formato general varía un poco, pero normalmente hacemos algo como esto:

- Reservamos 1 a 3 horas, dependiendo de cuánta discusión esperemos.
- Participantes: el PO, el equipo
- Nos vamos a una reunión cerrada, un rincón cómodo con sofás, el patio del tejado o algún sitio similar. Que podamos tener una discusión sin interrupciones.
- Normalmente no hacemos retros en la sala del equipo, ya que la atención de la gente suele diluirse.
- Alguien es designado secretario.
- El SM muestra la SB y, con ayuda del equipo, resume el Sprint, eventos importantes, decisiones, etc.
- Hacemos “la ronda”. Cada persona tiene una oportunidad de decir, sin ser interrumpida, qué piensan que ha ido bien, qué podría haber ido mejor y qué piensan que debería hacerse diferente en el próximo Sprint.
- Observamos la velocidad estimada frente a la real. Si hay una gran diferencia, intentamos analizar el por qué.
- Cuando el tiempo casi se ha acabado, el SM trata de resumir las sugerencias concretas sobre qué puede hacerse mejor el próximo Sprint.

Difundiendo las lecciones entre equipos

Una retrospectiva de Sprint, no trata sólo de cómo este equipo puede hacerlo mejor el próximo Sprint, tiene implicaciones más amplias que esa. Nuestra estrategia para manejar este hecho es muy simple. Una persona atiende a todas las reuniones de retrospectiva y actúa como puente de conocimiento. Muy informal.

Reglas importantes para la persona que actúa como “puente de conocimiento”:

- Debería ser escuchado.
- Si la retrospectiva es poco activa, debería estar listo para realizar preguntas simples pero puntuales para estimular la discusión dentro del grupo.
- Debe estar dispuesto a pasar tiempo visitando todas las retrospectivas de todos los equipos.
- Debería tener algún tipo de autoridad, de forma que pueda actuar sobre las sugerencias que estén fuera del control del propio equipo.

Cambiar o no cambiar

Hemos descubierto que, en muchos casos, simplemente identificar el problema es suficiente para que se resuelva por sí mismo automáticamente en el próximo Sprint. Especialmente si pegas la retrospectiva del Sprint en la pared de la sala del equipo (cosa que nosotros siempre olvidamos hacer, ¡vergüenza debería darnos!). Cada cambio que introduces tiene algún tipo de coste, así que antes de hacer nuevos cambios considera no hacer nada y esperar a que el problema desaparezca (o se haga más pequeño) automáticamente.

Si introduces un nuevo cambio cada vez que alguien se queja de algo, la gente se resistirá a revelar pequeñas áreas problemáticas, lo cual sería terrible.

Unidad 11: Descansos entre Sprints

Además del propio descanso, hay otra buena razón para dejar algo de espacio entre Sprints. Después de la demo y la retrospectiva tanto el Dueño de Producto como el Equipo estarán llenos de información y de ideas por digerir. Si se ponen de inmediato a planificar el próximo Sprint, es probable que nadie tenga la oportunidad de digerir ninguna información o lección aprendida, el Dueño de Producto no tendrá tiempo de ajustar sus prioridades después de la demo, etc.

Como mínimo, intentamos que la retrospectiva y la subsiguiente reunión de planificación de Sprint no ocurran el mismo día. Todo el mundo debería tener al menos una buena noche de sueño sin Sprint antes de comenzar el siguiente Sprint.

Una forma de hacer esto son los “días de laboratorio” (o como quieras llamarlos). Es decir, días en los que a los desarrolladores se les permite hacer esencialmente cualquier cosa que deseen.

Nuestro objetivo es tener un día de laboratorio entre cada Sprint. De esta forma obtenemos un descanso natural entre Sprints, y tendrás un equipo de desarrollo que tendrá una oportunidad realista de mantener su conocimiento al día. Además, es un estupendo beneficio laboral.