

3

Versión gratuita on-line.

Apoya este trabajo, compra la copia impresa:

<http://infoq.com/minibooks/scrum-xp-from-the-trenches>

Como nos preparamos para la planificación de Sprint

OK, el día de la planificación de Sprint se aproxima rápidamente. Una lección que hemos aprendido una y otra vez es:

Lección: Asegúrate de que la Pila de Producto está perfectamente lista antes de la reunión de planificación de Sprint.

¿Y esto qué significa? ¿Que todas las historias deban estar perfectamente bien definidas? ¿Qué las estimaciones sean correctas? ¿Qué todas las prioridades hayan sido fijadas? ¡No, no y no! Lo que significa realmente es:

- ¡La pila de producto debe existir! (¿Podéis imaginároslo?)
- Debería haber *una* Pila de Producto y *un* dueño de producto (por producto, claro).
- Todos los elementos importantes deberían tener ratios de importancia asignados, *diferentes* ratios de importancia. En realidad, da igual si los elementos menos importantes tienen todos el mismo valor, ya que probablemente no se discutirán durante la planificación de Sprint en cualquier caso. Cualquier historia sobre la que el Dueño de Producto piense que tiene una remota posibilidad de incluirse en el Sprint debería tener un nivel de importancia único definido. El ratio de importancia se emplea solo para ordenar los elementos por relevancia. Así que si el elemento A tiene una importancia de 20 y el elemento B una importancia de 100, simplemente significa que B es más importante que A. No significa que B sea cinco veces más importante que A. Si B tuviera una importancia de 21, ¡aun significaría lo mismo! Es útil dejar espacio entre la secuencia de números por si aparece un elemento C que es más importante que A pero menos importante que B. Por supuesto, le podríamos dar un ratio de importancia de 20,5 a C, pero queda mal, así que en vez de ello dejamos espacio entre números.
- El dueño de producto debe comprender cada historia (normalmente él es el autor, pero en algunos casos otras personas añaden solicitudes, que el Dueño de Producto puede priorizar). No necesita saber cómo exactamente debe implementarse, pero debería entender por qué la historia está ahí.

Nota: Otras personas aparte del Dueño de Producto pueden añadir sus historias a la Pila de Producto. Pero no pueden asignarles niveles de importancia, ese es

un cometido exclusivo del Dueño de Producto. Tampoco pueden establecer estimaciones, ese es un cometido exclusivo del equipo.

Otras aproximaciones que hemos intentado o evaluado:

- Usar Jira (nuestro sistema de seguimiento de errores) para almacenar la Pila de Producto. La mayoría de nuestros Dueños de Producto encuentran que deben hacer demasiados “clicks” en ella. Excel está bien, y es fácil de manejar directamente. Puedes añadir códigos de color fácilmente, reordenar los elementos, añadir nuevas columnas, notas, importar y exportar datos, etc. Todo ello “ad-hoc”.
- Utilizar una herramienta de soporte a procesos ágiles, como VersionOne, ScrumWorks, XPlanner, etc. Todavía no hemos conseguido probar ninguna de ellas, pero probablemente lo haremos.

4

Versión gratuita on-line.

Apoya este trabajo, compra la copia impresa:

<http://infoq.com/minibooks/scrum-xp-from-the-trenches>

Como hacemos la planificación de Sprint

La planificación de Sprint es una reunión crítica, probablemente la más importante de Scrum (en mi subjetiva opinión, por supuesto). Una planificación de Sprint mal ejecutada puede arruinar por completo todo el Sprint.

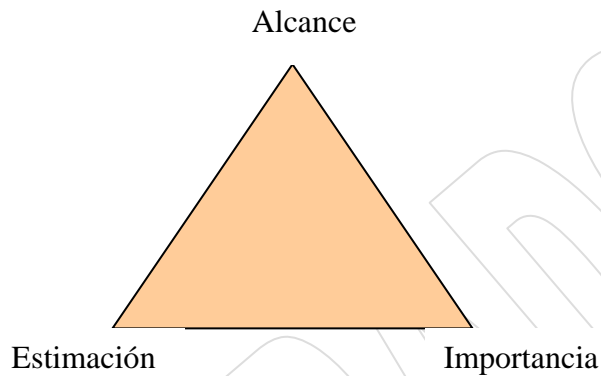
El propósito de la planificación de Sprint es proporcionar al equipo suficiente información como para que puedan trabajar en paz y sin interrupciones durante unas pocas semanas, y para ofrecer al Dueño de Producto suficiente confianza como para permitirse.

Vale, ha quedado un poco difuso. Una planificación de Sprint produce, concretamente:

- Una meta de Sprint.
- Una lista de miembros (y su nivel de dedicación, si no es del 100%)
- Una Pila de Sprint (lista de historias incluidas en el Sprint)
- Una fecha concreta para la Demo del Sprint.
- Un lugar y momento definidos para el Scrum Diario.

Por qué debe asistir el Dueño de Producto

A veces los Dueños de Producto se resisten a pasar horas con el equipo preparando la planificación de Sprint. “Mirad, tíos, ya os he listado lo que quiero. No tengo tiempo para estar en vuestra reunión de planificación”. Es un problema bastante serio. La razón por la que el equipo y el Dueño de Producto 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 Dueño de Producto. La estimación la 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 Dueño de Producto.

Normalmente, el Dueño de Producto 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. Conforme van haciéndolo, aparecerán dudas importantes respecto al alcance: “esta historia sobre ‘borrar usuario’, ¿incluye repasar todas las transacciones pendientes del usuario y cancelarlas?”. En algunos casos, las respuestas sorprenderán al equipo y les obligarán a cambiar sus estimaciones.

En algunos casos, la estimación para una historia no será la que el Dueño de Producto esperaba. Esto puede forzarle a cambiar la importancia de la historia o su alcance, lo que obligará al equipo a re-estimarla, etc., etc.

Este tipo de colaboración directa es fundamental en Scrum. De hecho, en todo el Desarrollo Ágil de software. ¿Qué ocurre si el Dueño de Producto 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 Dueño de Producto comprenda por qué su participación es crucial y esperar que cambie de parecer.
- Intentar que alguien del equipo se presente voluntario como delegado del Dueño de Producto durante la reunión. A continuación, decirle al Dueño de Producto “ya que no puedes venir a nuestra reunión vamos a dejar que Jeff te represente. Tendrá poder de decisión pleno para cambiar las prioridades y el alcance de las historias por ti durante la reunión. Te sugiero que te sincronices con él lo máximo posible antes de la reunión. Si no te gusta Jeff como delegado, por favor, sugiere a otro, siempre que ese otro pueda asistir a toda la reunión”.

- Tratar de convencer a la gerencia de que asigne un nuevo Dueño de Producto.
- Posponer el lanzamiento del Sprint hasta que el Dueño de Producto encuentre el tiempo para asistir a la reunión. Mientras tanto, no comprometerse a ninguna entrega y permitir que el equipo pase el día haciendo cualquier cosa que les parezca importante hacer.

Por qué la calidad no es negociable

En el triángulo mencionado anteriormente he dejado intencionadamente fuera una cuarta variable: la calidad. Tiendo a distinguir entre *calidad interna* y *calidad externa*.

- *Calidad externa*: es lo que perciben los usuarios del sistema. Un interfaz de usuario lento y poco intuitivo es un ejemplo de baja calidad externa.
- *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. Cosas como consistencia del diseño del sistema, cobertura de pruebas, legibilidad del código, refactorización, etc.

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 buena calidad externa. Es difícil construir algo sobre unos cimientos podridos.

Yo trato la calidad externa como parte del alcance. En algunos casos puede tener sentido, desde el punto de vista de negocio, liberar una versión del producto que tenga un internaza de usuario torpe y lento, y más tarde liberar una versión mejorada. Dejo esa decisión al Dueño de Producto, ya que él es el responsable de definir el alcance.

Sin embargo, 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. Nunca. (Vale, OK, *casí* nunca).

¿Y como distinguimos la diferencia entre aspectos de calidad externa y aspectos de calidad interna? Supongamos que el Dueño de Producto dice “OK tíos, respeto vuestras estimaciones de esfuerzo de 6 puntos de historia, pero estoy seguro de que podríamos usar algún apaño rápido para hacerlo en la mitad de tiempo si le dais una pensada”.

¡Ajá! Está intentando usar la calidad interna como una variable. ¿Cómo lo se? Porque quiere que reduzcamos la estimación de la historia sin “pagar el precio” de reducir el alcance. La palabra “apaño” debería disparar una alarma en vuestras mentes...

¿Y por qué no permitimos esto? Mi experiencia es que sacrificar la calidad interna es, prácticamente siempre, una idea terrible. 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.

En lugar de eso intento reconducir la discusión hacia el alcance. “Ya que es importante para ti que entreguemos esta historia pronto, ¿podemos reducir su alcance de forma que sea más rápida de implementar? Quizás podríamos

simplificar el manejo de errores y convertir “Manejo Avanzado de Errores” en una historia separada que reservemos para el futuro. O podríamos reducir la prioridad de otras historias de forma que podamos centrarnos en esta”.

Una vez que el Dueño de Producto aprende que la calidad interna no es negociable, normalmente se hace muy bueno en manipular las otras variables.

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

Lo más difícil de las Planificaciones de Sprint es que:

- 1) La gente no piensa que vayan a durar tanto...
- 2) ... ¡Pero lo hacen!

Todo en Scrum tiene una duración determinada (*time-boxed*). Me encanta esa única, simple y consistente regla. Intentamos cumplirla al cien por cien. Así que ¿qué hacemos cuando la reunión de planificación de Sprint está llegando al final y no hay señales de una meta de Sprint o Pila de Sprint? ¿Limitamos la reunión al tiempo establecido? ¿O la extendemos una hora? ¿O terminamos por hoy y continuamos al día siguiente?

Esto nos pasa una y otra vez, especialmente con los equipos nuevos. Así que, ¿qué hacer? No lo sé. Pero, ¿qué hacemos nosotros? Oh, um, vale, usualmente suelo cerrar la reunión brutalmente a la hora en punto. Acabarla. Dejar que el Sprint sufra. Más específicamente, le digo al equipo y al Dueño de Producto “bueno, esta reunión acaba en 10 minutos. No tenemos lo que se dice un Plan de Sprint, realmente. ¿Trabajamos con lo que tenemos o planifico otra reunión de cuatro horas mañana a las ocho de la mañana?”. Podéis imaginar cual es la respuesta... :o)

He intentado dejar que las reuniones se alarguen. Eso normalmente no vale para nada, porque la gente está cansada. Si no han producido un Plan de Sprint decente en 2 – 8 horas (o lo que sea que duren vuestras reuniones de planificación), probablemente no lo lograrán en otra hora más. La siguiente opción no está tan mal, planificar otra reunión al día siguiente. Excepto que la gente normalmente está impaciente y quieren comenzar con el Sprint, y no emplear otro montón de horas en planificación.

Así que acabo con la reunión. Y 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.

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

Agenda de la reunión de planificación de Sprint

Tener algún tipo de agenda u orden del día de la reunión de planificación de Sprint reducirá el riesgo de sobrepasar la duración determinada. Este es un ejemplo típico de nuestras agendas:

Reunión de planificación de Sprint: 13:00 – 17:00 (10 minutos de descanso cada hora)

- **13:00 – 13:30.** El Dueño de Producto comenta la meta del Sprint y resume la Pila de Producto. Se establece un lugar, fecha y hora para la Demo.
- **13:30 – 15:00.** El equipo da estimaciones de tiempo, y divide los elementos tanto como sea necesario. El Dueño de Producto actualiza los ratios de importancia. Se clarifican los elementos. Para todos los elementos de alta importancia se establece el “Como probarlo”.
- **15:00 – 16:00.** El equipo selecciona las historias que se incluirán en el Sprint. Se realizan cálculos de velocidad para chequear si es factible.
- **16:00 – 17:00.** Se selecciona un lugar y hora para el Scrum Diario (si es que cambio respecto al último Sprint). Se continúa dividiendo las historias en tareas.

La agenda no es en absoluto inamovible. El Scrum Master puede ampliar o acortar los periodos según sea necesario conforme progresa la reunión.

Definiendo la duración del Sprint

Uno de los resultados de la planificación de Sprint es una fecha de Demo de Sprint definida. Eso significa que debes determinar una duración del Sprint.

¿Y cuanto debería durar un 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.

Generalmente, los Dueños de Producto prefieren los Sprints cortos y a los desarrolladores les gustan los Sprints largos. Así que la duración del Sprint es un valor de compromiso. Hemos experimentados con varias duraciones y al final encontramos nuestra duración favorita: 3 semanas. La mayoría de nuestros equipos (pero no todos) hacen Sprints de 3 semanas. Suficientemente cortos para proporcionarnos agilidad corporativa, suficientemente largos para lograr *flujo* y recuperarse de los problemas que aparezcan durante el Sprint.

Una conclusión a la que hemos llegado es: *experimentad* con las duraciones al principio. No perdáis mucho tiempo *analizando*, simplemente seleccionad una duración decente y dadle un Sprint o dos, y entonces cambiad la duración. *En cualquier caso*, una vez que hayáis establecido una duración, *mantenedla* durante un buen periodo de tiempo. Después de unos pocos meses de experimentación, nosotros encontramos que 3 semanas estaba bien. Así que hacemos Sprints de 3 semanas, punto. A veces puede parecer un poco corto, a veces un poco largo. Pero manteniendo la misma duración conseguimos un

latido corporativo al que todo el mundo se acostumbra confortablemente. No hay discusiones sobre las fechas de entrega ya que todo el mundo sabe que cada 3 semanas hay una entrega. Punto.

Definiendo la meta del Sprint

Pasa todo el tiempo. En algún momento durante la reunión de planificación de Sprint, pregunto “¿Y cual es la meta del Sprint”, y todo el mundo me mira anonadado mientras el Dueño de Producto arruga la frente y se rasca la barbilla.

Por alguna razón, es *difícil* conseguir una meta de Sprint. Pero aun así he descubierto que realmente merece la pena exprimir una. Mejor una medio-meta roñosa que no tener ninguna meta. La meta podría ser “ganar más dinero” o “impresionar al Director General” o “hacer el sistema lo suficientemente bueno como para entregarlo a un grupo de usuarios beta reales” o “añadir soporte de backoffice básico” o lo que sea. Lo importante es que esté descrito en términos de negocio. Eso significa en términos en los que la gente de fuera del equipo lo pueda entender.

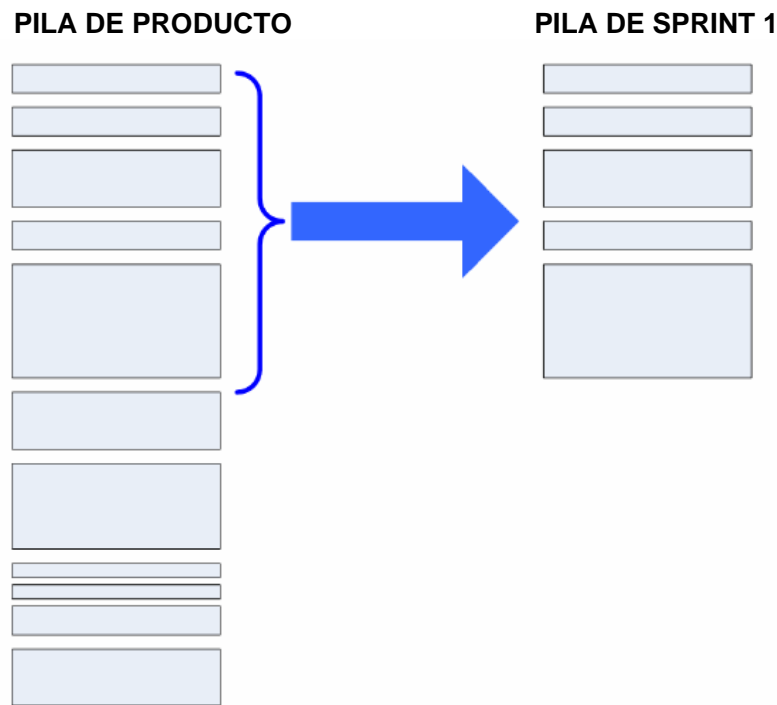
La meta de Sprint debería responder a la pregunta fundamental “¿*Por qué* hacemos este Sprint en vez de irnos todos de vacaciones?”. De hecho, una forma de obtener la meta del Dueño de Producto es precisamente hacerle esa misma pregunta.

La meta debería ser algo que no se haya logrado aun. “Impresionar al Director General” puede ser una meta aceptable, pero sólo si no está impresionado aun por el sistema tal y como se encuentra ahora mismo. Si fuera así, todo el mundo podría irse a casa y aun así lograr la meta del Sprint.

La meta del Sprint puede parece 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. Si tienes bastante equipos Scrum (como nosotros) trabajando en diferentes productos, es muy útil poder listar todas las metas de Sprint en una página Wiki (o donde sea) y colocarla en un sitio prominente donde todo el mundo en la empresa (no solo la alta dirección) pueda saber qué está haciendo la compañía - ¡y por qué!

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 Pila de Producto copiar en la Pila de Sprint.



Observa la imagen anterior. Cada rectángulo representa una historia, ordenadas por importancia. La historia más importante está al principio de la lista. El tamaño de cada rectángulo representa el tamaño de la historia (es decir, el tiempo estimado en puntos de historia). La altura del corchete azul representa la *velocidad estimada* del equipo, es decir, cuántos puntos de historia cree el equipo que puede completar durante el próximo Sprint.

La Pila de Sprint de la derecha es una instantánea de las historias la Pila de Producto. Representa las historias a las que el equipo se compromete durante este Sprint. *El equipo* decide cuántas historias incluirá en el Sprint. No el Dueño de Producto ni nadie más.

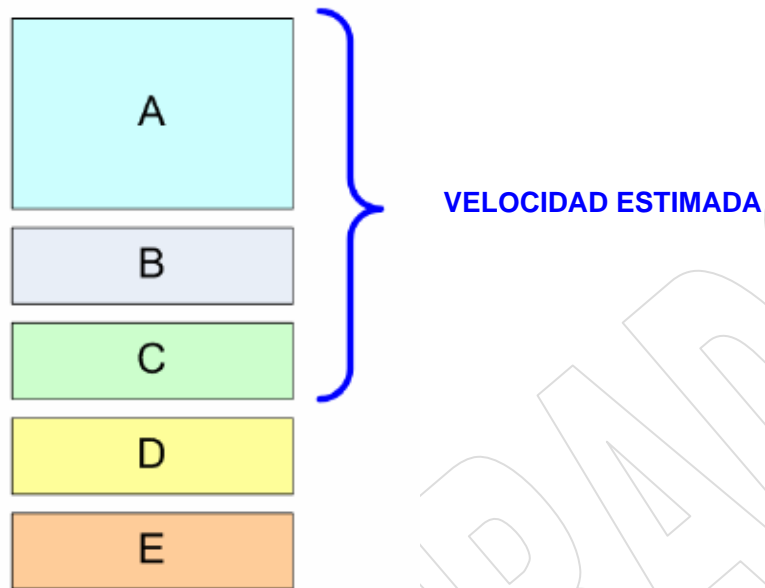
Esto plantea dos cuestiones:

1. ¿Cómo decide el equipo qué historias incluir en el Sprint?
2. ¿Cómo puede el Dueño de Producto alterar la decisión del equipo?

Empezaré con la segunda cuestión.

¿Cómo puede el Dueño de Producto alterar las historias que se incluyen en el Sprint?

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

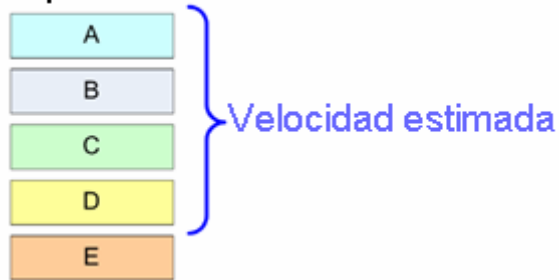
PILA DE PRODUCTO

Al Dueño de Producto 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?

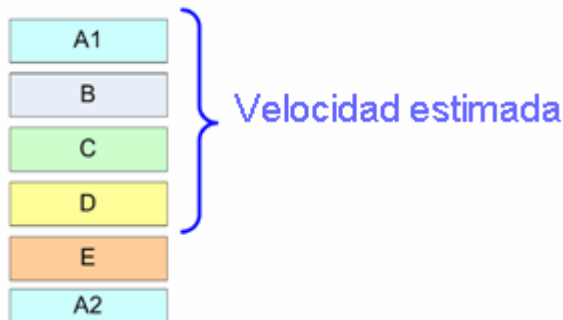
Una es re-priorizar. Si le da al elemento D la mayor importancia, el equipo se verá obligado a añadirlo al Sprint en primer lugar (descartando en ese caso la historia C).

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 sería dividir una historia. El Dueño de Producto podría decidir que hay algunos aspectos de la historia A que no son tan importantes, así que dividiría la historia A en dos historias A1 y A2 con diferentes niveles de importancia.

Opción 3:

Como ves, aunque el Dueño de Producto no puede controlar normalmente la velocidad estimada, hay varias formas en las que puede influenciar qué historias entran en cada Sprint.

¿Cómo decide el equipo qué historias incluir en el Sprint?

Utilizamos dos técnicas para esto:

1. A ojo de buen cubero
2. Cálculos de velocidad

Estimando a ojo de buen cubero

- **ScrumMaster:** “ A ver chicos, ¿podemos terminar la historia A en este Sprint?” (señala el elemento más importante de la Pila de Producto).
- **Lisa:** “Bah. Por supuesto que podemos. Tenemos tres semanas, y es una funcionalidad bastante trivial.”

- **ScrumMaster:** “Ok, ¿y si le añadimos la historia B?” (señala el segundo elemento más importante)
- **Tom&Lisa al unísono:** “Sigue siendo muy fácil.”
- **ScrumMaster:** “OK, ¿qué tal las historias A, B y C entonces?”
- **Sam (al Dueño de Producto):** “¿La historia C incluye manejo avanzado de errores?”
- **Dueño de Producto:** “No, no es necesario por ahora, podéis implementarlo con un manejo básico de errores.”
- **Sam:** “Entonces C podría entrar también.”
- **ScrumMaster:** “OK, ¿y si añadimos la historia D?”
- **Lisa:** “Hmmm...”
- **Tom:** “Creo que podríamos hacerlo.”
- **ScrumMaster:** “¿Seguro al 90%? ¿Al 50%?”
- **Lisa y Tom:** “Más bien al 90%.”
- **ScrumMaster:** “OK, D entra entonces. ¿Y si añadimos la historia E?”
- **Sam:** “Quizás.”
- **ScrumMaster:** “¿90%? ¿50%?”
- **Sam:** “Yo diría que más cerca del 50%”.
- **Lisa:** “Tengo mis dudas.”
- **ScrumMaster:** “OK, entonces lo dejamos fuera. Nos comprometemos a A, B, C y D. Por supuesto, terminaremos E si podemos, pero nadie debería contar con ello, así que lo dejamos fuera del plan de Sprint. ¿Qué tal así?”.
- **Todo el mundo:** “OK!”

El ojo de buen cubero funciona bastante bien para equipos pequeños y Sprints cortos.

Estimando usando cálculos de velocidad

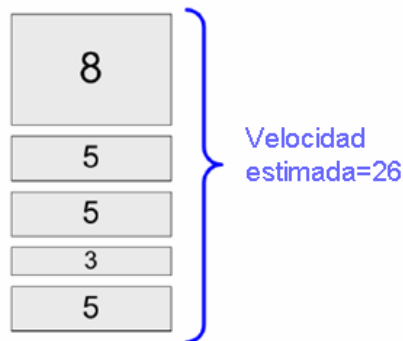
Esta técnica consta de dos pasos

1. Decidir la velocidad estimada
2. Calcular cuántas historias se pueden añadir sin sobrepasar la velocidad estimada

La velocidad es una medida de “cantidad de trabajo realizado”, donde cada elemento se evalúa en función de su estimación inicial.

El siguiente gráfico muestra un ejemplo de velocidad estimada al principio de un Sprint y la velocidad real al final de dicho Sprint. Cada rectángulo es una historia, y el número interior es la estimación inicial de dicha historia.

Comienzo de Sprint:



Final de Sprint:



Nota que la velocidad real esta basada en las estimaciones iniciales de cada historia. Cualquier actualización a la estimación de la historia realizada durante el Sprint es ignorada.

Ya puedo escuchar las objeciones: “¿Cómo va a ser esto útil? ¡Una velocidad alta o baja dependerá de un montón de factores! Programadores con poco sentido común, estimaciones iniciales incorrectas, cambios en el alcance, distracciones imprevistas durante el Sprint, etc.

Estoy de acuerdo en que se trata de un número bastante aproximado. Pero aun así es bastante útil, especialmente si lo comparamos con “nada en absoluto”. Te proporciona algunas verdades crudas: “sean cuales sean las razones, aquí está la diferencia aproximada entre cuánto pensábamos que podríamos hacer y cuánto hicimos en realidad”.

¿Qué pasa con las historias que casi se consiguieron durante el Sprint? ¿Por qué no conseguimos algunos puntos por ellas en la velocidad real? Bueno, esto es así para reforzar el hecho de que Scrum (y de hecho todo el Desarrollo Ágil de Software y *Lean Manufacturing* en general) gira en torno al concepto de conseguir que las cosas se hagan completamente, hasta un estado de “potencialmente entregable”. El valor de las cosas medio hechas es cero (podría de hecho ser negativo). Coged el libro de Donald Reinertsen “*Managing the Design Factory*” o uno de los libros de Poppendieck para aprender más sobre este concepto.

Así que, ¿mediante qué tipo de magia arcana estimamos la velocidad?

Una manera muy fácil de estimar la velocidad es revisar la historia del equipo. ¿Cuál fue su velocidad durante los últimos Sprints? Y entonces asumir que la velocidad será más o menos la misma en el próximo Sprint.

Esta técnica se conoce como *el tiempo que hizo ayer*. Solo es factible para equipos que ya han hecho algunos Sprints (de forma que haya estadísticas disponibles) y que harán el próximo Sprint más o menos de la misma manera, con el mismo tamaño de equipo, las mismas condiciones de trabajo, etc. Este, claro está, no siempre es el caso.

Una forma más sofisticada de hacerlo es realizar un simple cálculo de recursos. Digamos que estamos planificando un Sprint de 3 semanas (15 días laborables) con un equipo de 4 personas. Lisa estará de vacaciones 2 días. Dave sólo estará disponible al 50% y estará un día de vacaciones. Poniéndolo todo junto...

DÍAS DISPONIBLES

TOM	15
LISA	13
SAM	15
DAVE	7

50 DÍAS-HOMBRE DISPONIBLES

...Tenemos 50 días-hombre disponibles en este Sprint.

¿Es esta nuestra velocidad estimada? ¡No! Porque nuestra unidad de estimación son *puntos de historia* lo que, en nuestro caso, corresponde más o menos a “días-hombre ideales”. Un día-hombre ideal es un día perfectamente efectivo, sin distracciones, lo cuál es raro. Lo que es más, debemos tener en cuenta cosas como trabajo no planificado que se añade al Sprint, gente que se pone enferma, etc.

Así que nuestra velocidad estimada será sin duda menor de 50. ¿Pero cuanto menor? Para esto usamos el “factor de dedicación”.

VELOCIDAD ESTIMADA DE ESTE SPRINT

(DÍAS-HOMBRE DISPONIBLES) X (FACTOR DE DEDICACIÓN) = VELOCIDAD ESTIMADA

El factor de dedicación es una estimación de cómo de centrado va a estar el equipo. Un factor de dedicación bajo puede significar que el equipo espera encontrar muchas distracciones e impedimentos o que considera que sus propias estimaciones son optimistas.

La mejor manera de determinar un factor de dedicación razonable es estudiar el último Sprint (o incluso mejor, la media de los últimos Sprints).

FACTOR DE DEDICACIÓN DEL ÚLTIMO SPRINT

(FACTOR DE DEDICACIÓN) = $\frac{\text{(VELOCIDAD REAL)}}{\text{(DÍAS-HOMBRE DISPONIBLES)}}$

La *velocidad real* es la suma de las estimaciones iniciales que se completaron en el último Sprint.

Digamos que en el último Sprint se completaron 18 puntos de historia utilizando un equipo de 3 personas formado por Tom, Lisa y Sam trabajando las tres

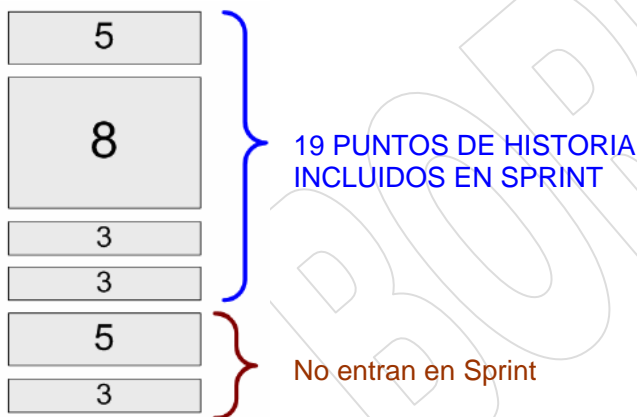
semanas hasta un total de 45 días-hombre. Y ahora estamos intentando calcular la velocidad del próximo Sprint. Para complicar las cosas, un nuevo tipo, Dave, se une al equipo para este Sprint. Teniendo en cuenta las vacaciones y demás asuntos tenemos 50 días-hombre ideales este Sprint.

FACTOR DE DEDICACIÓN DEL ÚLTIMO SPRINT

$$\begin{array}{l} \text{(18 PUNTOS HISTORIA)} \\ (40\%) = \frac{\text{HISTORIA}}{(45 \text{ DÍAS-HOMBRE})} \end{array} \quad 50 \text{ DÍAS-HOMBRE} \times 40\% = 20 \text{ PUNTOS-HISTORIA}$$

Así que nuestra velocidad estimada para el próximo Sprint es de 20 puntos de historia. Eso significa que el equipo debe añadir historias al Sprint hasta que sume aproximadamente 20.

COMIENZO DEL SPRINT



En este caso, el equipo puede escoger las 4 historias más importantes hasta un total de 19 puntos de historia, o las 5 historias más importantes hasta 24 puntos de historia. Digamos que escogen 4 historias, ya que se aproximan más a los 20 puntos de historia. Siempre que haya dudas, escoge añadir menos historias.

Dado que estas 4 historias suman 19 puntos, la velocidad estimada final para este Sprint es de 19.

“El tiempo que hizo ayer” es una técnica sencilla, pero úsala con cierta dosis de sentido común. Si el último Sprint fue especialmente malo porque la mayoría del equipo estuvo enfermo una semana, entonces podría ser adecuado asumir que no volverás a tener tan mala suerte y podrías estimar un factor de dedicación mayor el próximo Sprint. Si el equipo ha instalado recientemente un sistema super-rápido de compilación continua probablemente también podrás incrementar el factor de dedicación gracias a ello. Si una nueva persona se une a este Sprint deberías reducir su factor de dedicación para tener en cuenta su formación. Etc.

Siempre que sea posible, ten en cuenta varios Sprints y saca medias para conseguir estimaciones más acertadas.

¿Qué ocurre si el equipo es completamente Nuevo y no tienes ninguna estadística? Mira al factor de dedicación de otros equipos en circunstancias similares. ¿Qué pasa si no tienes otros equipos en los que fijarte? Adivina un factor de dedicación. Las buenas noticias son que sólo deberás adivinar para el primer Sprint. Después, tendrás estadísticas que podrás ir midiendo continuamente para aproximar mejor el factor de dedicación y la velocidad estimada.

El factor de dedicación “por defecto” que usamos para equipos nuevos es habitualmente el 70%, dado que es ahí donde terminan la mayoría de nuestros otros equipos con el tiempo.

¿Qué técnicas de estimación usamos?

He mencionado varias técnicas anteriormente – ojo de buen cubero, cálculo de velocidad basado en el tiempo que hizo ayer y cálculo de velocidad basado en días-hombre disponibles y factor de dedicación.

Así que, ¿qué técnica usamos?

Usualmente combinamos todas estas técnicas hasta cierto punto. Realmente no conlleva mucho esfuerzo.

Miramos el factor de dedicación y la velocidad real del último Sprint. Miramos al total de recursos disponibles para este Sprint y estimamos el factor de dedicación. Discutimos cualquier diferencia entre estos dos factores de dedicación y hacemos los ajustes que sean necesarios.

Una vez que tenemos la lista preliminar de historias a incluir en el Sprint hacemos un chequeo a “ojo de buen cubero”. Le pido al equipo que ignore los números por un momento y que piensen sobre si les *parece* un bocado realista como para zampárselo en un Sprint.

Si parece demasiado, quitamos una historia o dos. Y viceversa. Al final del día, el objetivo es simplemente decidir qué historias se incluyen en el Sprint. Factores de dedicación, disponibilidad de recursos y velocidad estimada son sólo medios para conseguir dicho objetivo.

Por qué usamos tarjetas

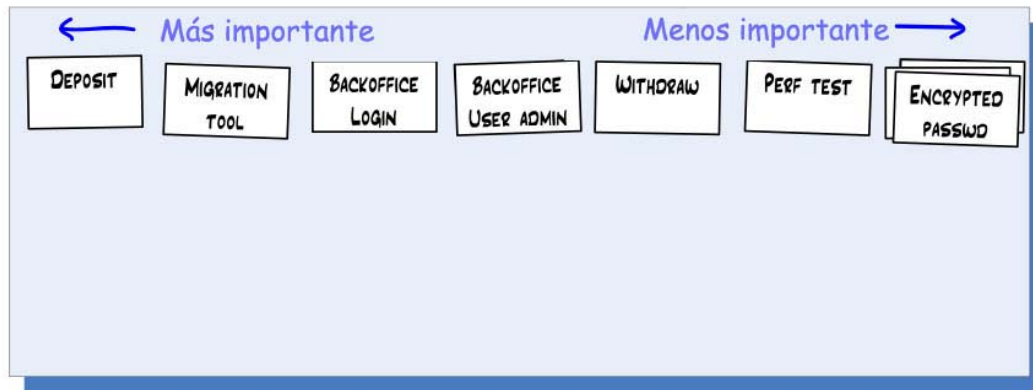
La mayor parte de la reunión de planificación de Sprint se dedica a las historias de la Pila de Producto. Estimándolas, repriorizándolas, clarificándolas, dividiéndolas, etc.

¿Cómo hacemos esto en la práctica?

Bueno, por defecto, los equipos solían conectar el proyector, mostrar la Pila basada en Excel y alguien (típicamente el Dueño de Producto o el ScrumMaster) toma el teclado, murmura algo sobre cada historia e invita a la discusión. Conforme el equipo y el Dueño de Producto discuten las prioridades y los detalles, la persona en el teclado actualiza cada historia directamente en Excel.

¿Suenan bien? Bueno, pues no. De hecho apesta. Y lo que es peor, el equipo *no se da cuenta* de que apesta hasta que llega el final de la reunión y se dan cuenta de que no han conseguido revisar toda la lista de historias.

Una solución que funciona mucho mejor es crear tarjetas y ponerlas en la pared (o en una mesa grande).



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 despiertos y alerta más tiempo
- Todo el mundo se siente personalmente más involucrado (y no solamente el tipo del teclado)
- 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 (ver “cómo hacemos la Pila de Sprint”).

Puedes escribirlas a mano o (como hacemos nosotros) utilizar un simple programa para generar las tarjetas directamente desde la Pila de Producto (PD – el programa está disponible en mi blog, en <http://blog.crisp.se/henrikkniberg>).

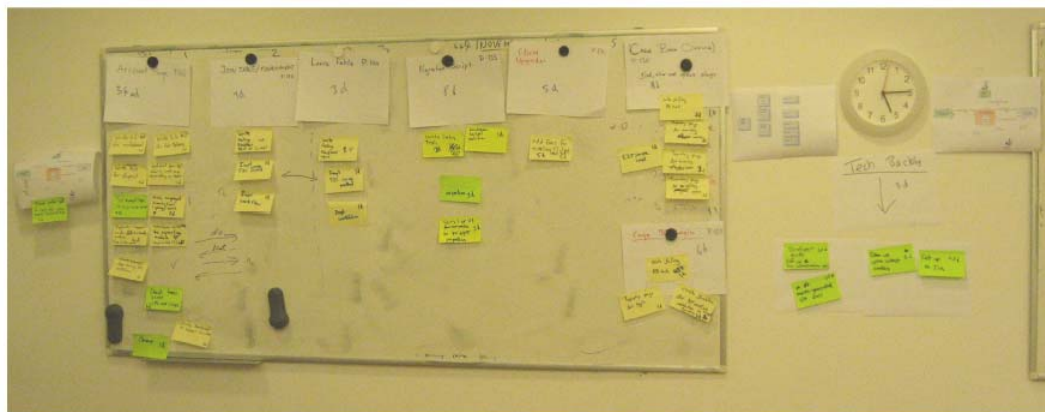
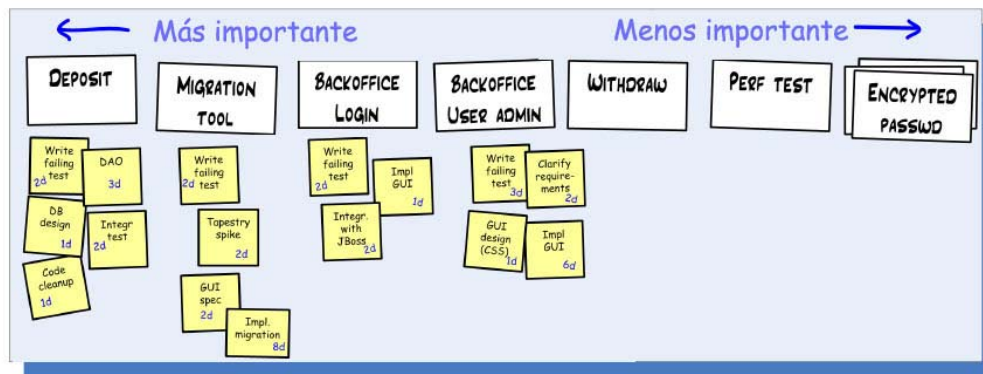
Elemento de Pila	
Depósito	
Notas	Importancia
Necesita un diagrama secuencial UML. No hay que preocuparse aún por el cifrado.	30
Como probarlo	Estimación
Entrar, abrir la página de depósito, depositar 10€, ir a la página de balance y comprobar que se ha incrementado en 10€	

Importante: Tras la reunión de planificación de Sprint, nuestro ScrumMaster actualiza manualmente la Pila de Producto en Excel respecto a cualquier cambio que se haya realizado sobre las tarjetas de historia físicas. Sí, es una pequeña molestia administrativa, pero lo encontramos perfectamente asumible considerando cuánto más eficiente es la reunión de planificación de Sprint utilizando tarjetas físicas.

Una nota sobre el campo “importancia”: se trata de la importancia tal y como figura en la Pila de Producto en Excel en el momento de la impresión. Tenerla en las tarjetas hace que sea más sencillo ordenarlas físicamente por importancia (normalmente, ponemos las más importantes a la izquierda y las menos importantes a la derecha). En cualquier caso, una vez que las tarjetas están en la pared puedes ignorar el ratio de importancia y en lugar de ello usar el orden físico para indicar la importancia relativa. Si el dueño de producto cambia dos historias de sitio no pierdas el tiempo actualizando los ratios de importancia en la tarjeta. Simplemente asegúrate de actualizarlas en la hoja Excel después de la reunión.

Las estimaciones de tiempo son normalmente más fáciles de hacer (y más exactas) si una historia se subdivide en tareas. De hecho, utilizamos la palabra “actividades” porque “tareas” significa algo *completamente* diferente en Sueco :o)

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 post-it representa una tarea dentro de dicha historia.



No actualizamos la Pila de Producto en Excel respecto a nuestra división en 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 Pila de Producto en Excel sincronizada.
- De todas formas, el Dueño de Producto no necesita estar involucrado a ese nivel de detalle.

Al igual que con las tarjetas de historia, los post-it de tareas pueden reutilizarse directamente en la Pila de Sprint (ver “cómo hacemos Pilas de Sprint”).

Definición de “terminado”

Es importante que el dueño de producto y el equipo estén de acuerdo en una definición clara de “terminado”. ¿Está terminada una historia cuando todo el código ha sido chequeado? ¿O está terminada cuando se ha instalado en un entorno de pre-producción y ha sido verificada por un equipo de pruebas de integración? Siempre que es posible, utilizamos la definición “lista para pasar a producción”, aunque a veces debemos conformarnos con “instalado en pre-producción y lista para las pruebas de aceptación”.

Al principio solíamos tener listas de comprobación detalladas para esto. Ahora usualmente decimos “una historia está terminada cuando el encargado de pruebas del equipo Scrum lo dice”. Así que es labor del encargado de pruebas asegurarse de que la intención del Dueño de Producto es correctamente

entendida por el equipo, y que el elemento esté lo suficientemente “terminado” como para cumplir con la definición de terminado.

Nosotros hemos llegado a la conclusión de que todas las historias no se pueden tratar igual. Una historia llamada “formulario de consulta de usuarios” se tratará de una forma muy diferente a otra llamada “manual de operaciones”. En el último caso, la definición de terminado puede significar simplemente “aceptada por el equipo de operaciones”. Es por eso que el sentido común es, a menudo, mejor que las listas de comprobación formales.

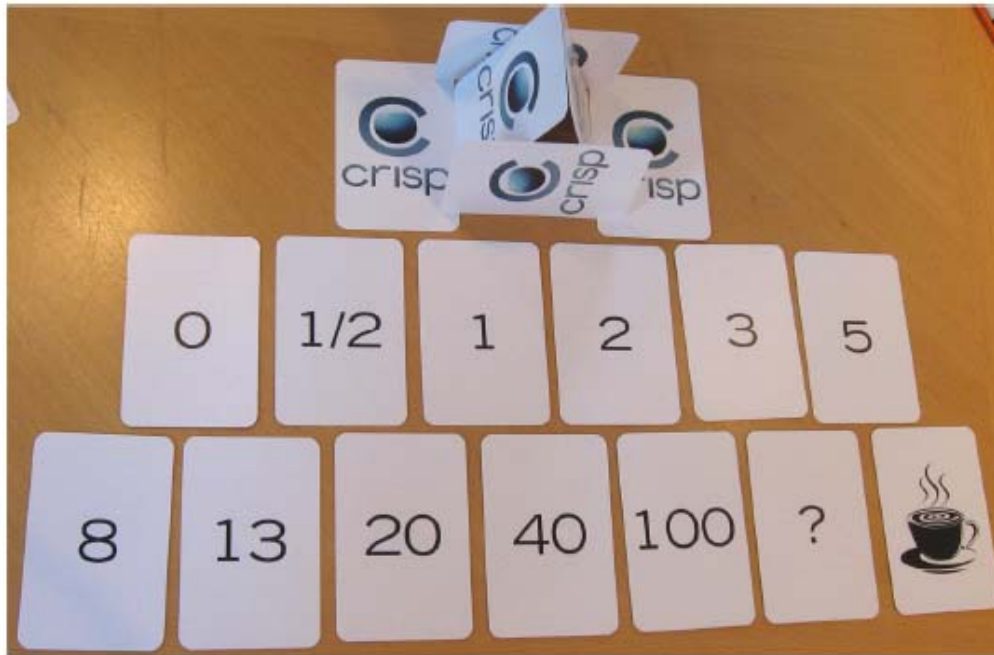
Si os sentís confusos con frecuencia acerca de la definición de terminado (cosa que nos ocurría al principio) probablemente deberíais tener un campo “definición de terminado” en cada historia individual.

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 (diseño de interfaz de usuario, programación, pruebas, etc.).
- 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 de que cada miembro del equipo comprende de qué trata cada elemento. Esto incrementa las posibilidades de que unos miembros del equipo ayuden a otros durante el Sprint. También mejora las posibilidades de que aparezcan pronto las preguntas importantes sobre cada historia.
- Cuando pedimos a todo el mundo que de estimaciones muchas veces encontramos discrepancias en las que dos miembros del equipo tienen estimaciones tremendamente distintas sobre la misma historia. Es mejor descubrir tales cosas al principio.

Si pides al equipo que proporcione una estimación, normalmente la persona que entiende mejor la historia será el primero en soltar una. Desafortunadamente, esto afectará severamente a las estimaciones de los demás. Hay una técnica excelente para evitar esto: se llama planning poker (creo que el término fue acuñado por Mike Cohn).



Cada miembro del equipo cuenta con una baraja de 13 cartas, como las que se muestran en la imagen. Cada vez que hay que estimar una historia, cada miembro del equipo selecciona una carta que representa su estimación de tiempo (en puntos de historia) y la coloca bocabajo en la mesa.

Cuando todos los miembros del equipo han preparado sus cartas, se les da la vuelta al mismo tiempo. Así obligamos a cada miembro del equipo a pensar por sí mismo en lugar de seguir la estimación de otro.

Si hay mucha discrepancia entre dos estimaciones, el equipo discute las diferencias y trata de construir una imagen común del trabajo necesario para la historia. Pueden hacer algún tipo de división en tareas. Después, el equipo estima de nuevo. Este bucle se repite hasta que la estimación de tiempo converge, es decir, que todas las estimaciones sean *aproximadamente* las mismas para esa historia.

Es importante que los miembros del equipo recuerden que deben estimar el total de tiempo necesario para la historia. No solamente “su” parte del trabajo. El encargado de pruebas no debería estimar solo la cantidad de trabajo de pruebas.

Date cuenta de que la secuencia de números no es lineal. Por ejemplo, no hay nada entre 40 y 100. ¿Por qué?

Esto es para evitar una falsa sensación de exactitud para las estimaciones de tiempo más grandes. Si una historia se estima aproximadamente en 20 puntos, no es relevante discutir si deberían ser 20, 18 o 21. Todo lo que sabemos es que es una historia grande y es difícil de estimar. Así que 20 es nuestra idea aproximada.

¿Quieres estimaciones más detalladas? Divide la historia en historias más pequeñas y trata de estimar las historias pequeñas.

Y no, no se puede hacer trampas combinando un 5 y un 2 para hacer un 7. Tienes que escoger entre 5 y 8, no hay 7.

Algunas cartas especiales sobre las que hablar:

- 0 = “esta historia ya está hecha” o “esta historia es prácticamente nada, apenas unos minutos de trabajo”.
- ¿ = “no tengo ni la más remota de las ideas. Nada”.
- Taza de café = “estoy demasiado cansado para pensar. Tomemos un descanso”.

Clarificando historias

Lo peor ocurre cuando el equipo demuestra orgulloso una nueva funcionalidad en la demo de Sprint y el Dueño de Producto frunce el ceño y dice “vale, muy bonito, pero ¡eso no es lo que yo pedí!”.

¿Cómo te aseguras de que el concepto que tiene el Dueño de Producto sobre una historia coincida con el concepto del equipo? ¿O de que cada miembro del equipo tenga el mismo concepto sobre la historia? Bueno, no se puede. Pero hay algunas técnicas simples para identificar los malentendidos más flagrantes. La técnica más simple es simplemente asegurarse de que todos los campos están rellenos para cada historia (o más específicamente, para cada historia con suficiente importancia como para ser considerada en este Sprint).

Ejemplo 1:

El equipo y el Dueño de Producto están contentos con el plan de Sprint y están listos para terminar con la reunión. El Scrum Master dice “esperad un segundo, esta historia llamada “añadir usuario”, no tiene estimación. Hagamos una estimación.” Tras un par de rondas de planning poker el equipo acuerda 20 puntos de historia, haciendo que el Dueño de Producto se levante preso de un ataque de rabia, “¡¿Cómo?!”. Tras unos minutos de discusión acalorada, resulta que el equipo entendió mal el alcance de “añadir usuario” y pensaban que significaba “un bonito interfaz Web para añadir, eliminar, buscar usuarios”, cuando el Dueño de Producto quería decir “añadir usuarios a mano haciendo sentencias SQL contra la base de datos”. Estiman de nuevo y acaban con 5 puntos de historia.

Ejemplo 2:

El equipo y el Dueño de Producto está contentos con el plan de Sprint y están listos para acabar la reunión. El Scrum Master dice “esperad un segundo, esta historia que se llama “añadir usuario”, ¿cómo debería demostrarse?”. Se producen algunos murmullos y tras un minuto alguien dice “bueno, primero nos autenticamos en la Web y luego...” y el Dueño de Producto interrumpe “¿autenticarse en la Web? No, no, no, esta funcionalidad no debería de ninguna forma ser parte del sitio Web, debería ser sólo un script muy sencillo para los administradores técnicos”.

La descripción de “como probarlo” de cada historia puede (y debe) ser *muy breve*. De otra forma no terminarás con la planificación de Sprint a tiempo. Es básicamente una descripción básica a alto nivel, en castellano simple, de cómo ejecutar el escenario de pruebas más común manualmente. “Haz esto, entonces haz lo otro y verifica aquello”. He descubierto que estas descripciones simples a

menudo descubren malentendidos importantes sobre el alcance de una historia. Menos mal que las descubrimos pronto, ¿verdad?

Dividiendo historias en historias más pequeñas

Las historias no deberían ser demasiado pequeñas ni demasiado grandes (en términos de estimación). Si tienes un montón de historias de 0.5 puntos probablemente vas a ser una víctima de la microgestión. Por otra parte, una historia de 40 puntos corre un importante riesgo de acabar *parcialmente* completa, lo que no aporta valor a tu empresa y simplemente incrementa la administración. Lo que es más, si tu velocidad estimada es 70 y tus dos historias más prioritarias pesan 40 puntos cada una, la planificación se vuelve difícil. Tienes que elegir entre comprometerte a más de lo que deberías (escoger las dos historias) o a menos (elegir sólo una).

Creo que casi siempre es posible dividir una historia grande en historias más pequeñas. Simplemente asegúrate de que las historias pequeñas siguen representando entregables con valor de negocio.

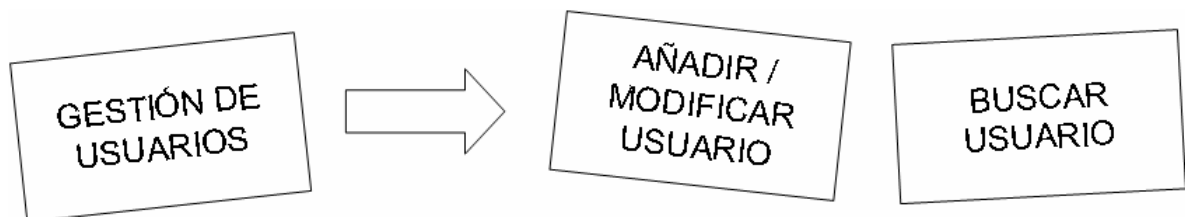
Normalmente buscamos historias con estimaciones de 2 a 8 días/hombre. Nuestra velocidad es usualmente 40-60 para un equipo típico, así que eso nos da para unas 10 historias por Sprint. A veces sólo 5, y a veces hasta 15. Es un número de tarjetas gestionable con el que trabajar.

Dividiendo las historias en tareas.

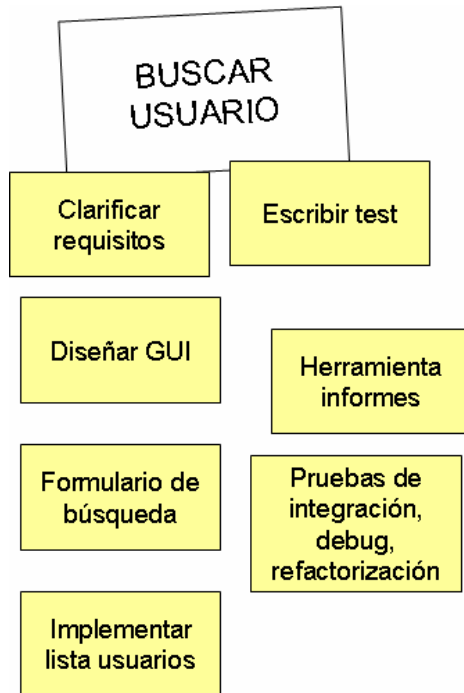
Un momento, ¿Cuál es la diferencia entre historias y tareas? Una pregunta muy válida.

La diferencia es muy simple. Las historias son entregables de los que el Dueño de Producto se preocupa. Las tareas son no-entregables, o aspectos de los que el Dueño de Producto no se preocupa.

Ejemplo de división de una historia en historias más pequeñas:



Ejemplo de división de una historia en tareas:



Algunas observaciones interesantes:

- Los equipos que están empezando con Scrum son reticentes a perder tiempo dividiendo un montón de historias en tareas desde el principio. Algunos piensan que es una aproximación tipo “cascada”.
- Para historias que se entienden bien, es tan fácil hacer esto desde el principio como hacerlo más tarde,
- Este tipo de división frecuentemente revela trabajo adicional que hace que las estimaciones suban, con lo que se consigue un plan de Sprint más realista.
- Este tipo de división desde el principio hace que los Scrum diarios sean notablemente más eficientes.
- Incluso si la división es inexacta y cambia una vez que empezamos, todas las ventajas anteriormente mencionadas siguen siendo válidas.

Así que tratamos que el tiempo disponible para la planificación de Sprint sea suficientemente largo como para que de tiempo a hacer esto, pero si el tiempo se acaba lo dejamos como esté (ver “Dónde trazar la línea” más adelante).

Nota – nosotros practicamos TDD (test driven development, desarrollo orientado a test) lo que en la práctica significa que la primera tarea para casi todas las historias es “escribir una prueba” y la última es “refactorizar” (es decir, mejorar la legibilidad del código y eliminar la duplicación).

Definiendo el sitio y la hora para el Scrum diario

Uno de los productos frecuentemente olvidados de 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.

Yo prefiero las reuniones por la mañana. Pero debo admitir que realmente no hemos probado a hacer Scrum diario por las tardes o al mediodía.

Desventaja de Scrums por las tardes: cuando llegas al trabajo por la mañana, tienes que acordarte de qué le dijiste a la gente sobre lo que deberían hacer hoy.

Desventaja de los Scrums por las mañanas: cuando llegas al trabajo por la mañana, debes acordarte de qué hiciste ayer para informar sobre ello hoy.

En mi opinión, la primera desventaja es peor, ya que lo más importante es *lo que vas a hacer*, no *lo que hiciste ayer*.

Nuestro procedimiento por defecto es seleccionar la hora más temprana a la que ningún miembro del equipo vaya a quejarse. Usualmente las 9:00, 9:30 o 10:00. Lo más importante es que sea a una hora a la que todo el equipo acepte con total convencimiento.

Dónde trazar la línea

OK, el tiempo se está agotando. De todos los asuntos que queremos resolver durante la planificación de Sprint, ¿qué abandonamos si nos quedamos sin tiempo?

Bueno, normalmente uso la siguiente lista de prioridades:

Prioridad 1: Una meta de Sprint y una fecha para la demo. Esto es lo mínimo que necesitas para comenzar un Sprint. El equipo tiene una meta y una fecha de finalización, y pueden trabajar directamente con la Pila de Producto. Apesta, sí, y deberías considerar seriamente organizar una nueva reunión de planificación de Sprint mañana mismo, pero si realmente necesitas que el Sprint comience entonces probablemente puedas hacerlo con esto. Para ser honesto, yo nunca he empezado un Sprint con tan poca información.

Prioridad 2: Lista de qué historias ha aceptado terminar el equipo en este Sprint.

Prioridad 3: Una estimación para cada historia del Sprint.

Prioridad 4: “Como probarlo”, relleno para cada historia del Sprint.

Prioridad 5: Cálculos de velocidad y recursos, como chequeo de la planificación del Sprint. Incluyendo una lista de los miembros del equipo y sus compromisos (de otra forma, no podrías calcular la velocidad).

Prioridad 6: Un sitio y hora específicos para la realización del Scrum diario. Solo necesitas un momento para decidirlo, pero si te quedas sin tiempo el Scrum Master puede simplemente decidir esto después de la reunión y mandar un correo a todo el mundo.

Prioridad 7: Historias divididas en tareas. Esta división puede sin embargo hacerse diariamente durante los Scrum diarios, pero interferirá levemente el flujo del Sprint.

Historias técnicas

He aquí un asunto complejo: historias técnicas. O elementos no-funcionales, o como quieras llamarlos.

Me refiero a 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 Dueño de Producto.

Las llamamos “historias técnicas”.

Por ejemplo:

- **Instalar un servidor de compilación continua:**
 - Por qué debe hacerse: porque ahorra cantidades inmensas de tiempo a los desarrolladores y reduce el riesgo de problemas explosivos de integración al final de la iteración.
- **Escribir una descripción general del diseño:**
 - Por qué debe hacerse: porque los desarrolladores olvidan constantemente el diseño general, y entonces escriben código inconsistente. Necesitan una visión global documentada para mantener a todo el mundo en la misma línea de diseño.
- **Refactorizar la capa de acceso a datos:**
 - Por qué debe hacerse: porque la capa de acceso a datos se ha vuelto realmente desordenada y le está costando tiempo a todo el mundo debido a la confusión y los errores innecesarios. Limpiar el código ahorraría tiempo a todo el mundo y mejoraría la robustez del sistema.
- **Actualizar Jira (seguimiento de errores)**
 - Por qué debe hacerse: la actual versión es demasiado lenta y falla demasiado. Actualizar a una nueva versión ahorrará tiempo a todo el mundo.

¿Son historias en el sentido normal? ¿O son tareas que no están conectadas a ninguna historia específica? ¿Quién las prioriza? ¿Debería involucrarse el Dueño de Producto en estos asuntos?

Hemos experimentado mucho con diferentes maneras de manejar las historias técnicas. Hemos intentado tratarlas como historias de primera clase, como todas las demás. Esto no funcionó bien, ya que cuando el Dueño de Producto priorizaba la Pila de Producto era como comparar peras con manzanas. De hecho, por razones obvias, las historias técnicas obtenían siempre mínima prioridad con razonamientos del tipo “sí, chavales, seguro que la compilación continua es muy importante y todo eso, pero construyamos primero algo que nos permita facturar, ¿vale? Entonces podréis añadir vuestras chucherías técnicas más adelante, ¿OK?”

En algunos casos el Dueño de Producto tiene razón, pero a menudo no es así. Hemos concluido que el Dueño de Producto no siempre está cualificado para manejar estos compromisos. Así que esto es lo que hacemos:

1) Intentamos evitar las historias técnicas. Busca efusivamente formas de transformar las historias técnicas en historias normales con valor de negocio mensurable. Así el Dueño de Producto tendrá mejores oportunidades para realizar decisiones correctas entre los pros y los contras.

2) Si no podemos transformar una historia técnica en una historia normal, intentamos ver si es posible hacerla como una tarea dentro de otra historia. Por ejemplo, “refactorizar la capa de acceso a datos” podría ser una tarea dentro de la historia “editar usuario”, ya que esto involucra utilizar la capa de acceso a datos.

3) Si lo anterior falla, definirla como historia técnica y mantener una lista separada con dichas historias. Permitimos al Dueño de Producto que vea dicha lista, pero no que la modifique. Usamos los factores de dedicación y la velocidad estimada para negociar con el Dueño de Producto y sacar algo de tiempo del Sprint para implementar estas historias.

Ejemplo (un diálogo muy similar a este ocurrió durante una de nuestras reuniones de planificación de Sprint):

- **Equipo:** “Tenemos algunos asuntos técnicos internos que deben hacerse. Nos gustaría presupuestar un 10% de nuestro tiempo para ello, es decir, reducir el factor de dedicación del 75% al 65%. ¿Os parece bien?”
- **Dueño de producto:** “¡Y una leche! ¡No tenemos tiempo!”
- **Equipo:** “Bueno, mira el ultimo Sprint (todas las cabezas se giran hacia la velocidad que está apuntada en la pizarra). Nuestra velocidad estimada fue 80, y la velocidad real fue de 30, ¿correcto?”
- **Dueño de Producto:** “¡Exacto! ¡Así que no tenemos tiempo para asuntos técnicos internos! ¡Tenemos que añadir nuevas funcionalidades!”
- **Equipo:** “Bueno, la razón por la que nuestra velocidad fue tan mala fue que pasamos demasiado tiempo intentando conseguir versiones consistentes para probar”.
- **Dueño de Producto:** “Sí, ¿Y?”
- **Equipo:** “Así que proponemos dedicar un 10% de nuestro tiempo del Sprint para implementar un servidor de compilación continua y otras cosas que nos ayudarán a hacer la integración menos penosa. Esto incrementará nuestra velocidad *al menos un 20%* para los próximos Sprints, ¡para siempre!”.
- **Dueño de Producto:** “¿En serio? ¿Y por qué no lo hicimos en el último Sprint entonces?”.
- **Equipo:** “Eh...porque no querías que lo hiciéramos...”
- **Dueño de Producto:** “Oh, um, bueno, vale, parece una buena idea hacerlo ahora entonces...”

Por supuesto, la otra opción es simplemente mantener al Dueño de Producto fuera del ciclo o darle un factor de dedicación no negociable. Pero no hay excusa para no *intentar* llegar a un consenso primero.

Si el Dueño de Producto es un tipo competente y razonable (y nosotros hemos tenido suerte en eso) yo sugiero mantenerle lo más informado posible y permitirle establecer las prioridades generales. La transparencia es uno de los valores principales de Scrum, ¿no es cierto?

Sistema de seguimiento de errores vs. Pila de Producto

He aquí un asunto complejo. Excel es un formato estupendo para la Pila de Producto. Pero aun así necesitas un sistema de seguimiento de errores, y Excel probablemente no funcione bien. Nosotros usamos Jira.

Así que, ¿cómo incluimos los elementos de Jira en la reunión de Planificación de Sprint? Quiero decir que no sirve simplemente ignorarlos y concentrarse solo en las historias.

Hemos intentado varias estrategias:

- 1) El Dueño de Producto imprime los elementos de Jira más importantes, los trae a la reunión de planificación de Sprint y los coloca en la pared junto al resto de historias (especificando así implícitamente la prioridad de estos elementos comparados con las otras historias).
- 2) El Dueño de Producto crea historias que se refieren a los elementos de Jira. Por ejemplo, "Arreglar los errores más críticos de informes de backoffice, Jira-124, Jira-126 y Jira-180".
- 3) La corrección de errores se considera algo fuera del Sprint. El equipo mantiene un factor de dedicación suficientemente bajo (por ejemplo, el 50%) para asegurar que tienen tiempo suficiente para arreglar errores. Entonces simplemente se asume que el equipo pasará parte de su tiempo arreglando los errores reportados en Jira.
- 4) Hacer la Pila de Producto en Jira (es decir, abandonar Excel). Tratar los errores como cualquier otra historia.

No hemos acabado de determinar qué estrategia es mejor para nosotros. De hecho, varía de equipo a equipo y de Sprint a Sprint. Tiendo no obstante a favorecer la primera estrategia. Es sencilla y agradable.

¡Por fin acabó la reunión de planificación de Sprint!

¡Wow, nunca creí que este capítulo sobre las reuniones de planificación de Sprint sería tan largo! Supongo que eso refleja mi opinión de que la reunión de planificación de Sprint es lo más importante que se hace en Scrum. Emplea una buena cantidad de esfuerzo en hacerla bien, y el resto será mucho más fácil.

La reunión de planificación de Sprint tiene éxito si todo el mundo (todos los miembros del equipo y el Dueño de Producto) sale de la reunión con una sonrisa y se levanta a la mañana siguiente con una sonrisa, y hacen su primer Scrum diario con una sonrisa. A partir de ahí, por supuesto, toda clase de cosas pueden ir horriblemente mal, pero al menos no puedes echarle la culpa al plan de Sprint :o)

5

Versión gratuita on-line.

Apoya este trabajo, compra la copia impresa:

<http://infoq.com/minibooks/scrum-xp-from-the-trenches>

Como comunicamos los Sprints

Es importante mantener a toda la compañía informada sobre lo que está ocurriendo. De otra forma, la gente se queja constantemente o, incluso peor, hacen falsas presunciones sobre lo que está ocurriendo.

Utilizamos una “página de información de Sprint” para esto.

Equipo Jackass, Sprint 15
Objetivo de Sprint:

- ¡Versión beta lista!

Pila de Sprint:

- Deposito (3)
- Herramienta de migración (8)
- Login en backoffice (5)
- Administración de usuarios de backoffice (5)

Velocidad estimada: 21
Calendario:

- Periodo de Sprint: 6/11/2006 a 24/11/2006
- Scrum diario: 9:30-9:45 en la sala del equipo
- Demo de Sprint: 24/11/2006, 13:00 en la cafetería

Equipo:

- Jim
- Erica (Scrum Master)
- Tom (75%)
- Eva
- John

A veces incluimos también información sobre como se demostrará cada historia.

Tan pronto como sea posible tras la reunión de planificación de Sprint, el Scrum Master crea esta página, la pone en el Wiki y manda un *spam* a toda la compañía:

Asunto: Comienzo del Sprint 15 de Jackass

¡Hola a todos! El equipo Jackass ha comenzado con el Sprint 15. Nuestro objetivo es demostrar la versión el 24 de noviembre.

Podéis ver la página de información del Sprint en:

<http://wiki.miempresa.com/jackass/sprint15>

También tenemos un “panel de control” en el Wiki, que enlaza con todos los Sprints que estén teniendo lugar:

Panel de control corporativo:

Sprints en curso:

- [Equipo X – Sprint 11](#)
- [Equipo Y – Sprint 12](#)
- [Equipo Z – Sprint 1](#)

Adicionalmente, el Scrum Master imprime la página de información de Sprint y la pega en la pared fuera de la sala del equipo. Así, todo el que pasa cerca puede ver la página de información de Sprint y averiguar qué está haciendo el equipo. Ya que esto incluye el sitio y la hora a la que se hace el Scrum diario y la demo de Sprint, ya sabe a donde ir para averiguar más.

Cuando el Sprint se acerca a su final, el Scrum Master recuerda a todo el mundo que se acerca la demo;

Asunto: Demo del Sprint 15 de Jackass mañana en la cafetería

¡Hola a todos! Os invitamos a asistir a la demo de Sprint que tendrá lugar mañana a las 13:00 en la cafetería.

Demostraremos una versión beta lista para liberar.

Podéis ver más en la página de Sprint:

<http://wiki.miempresa.com/jackass/sprint15>

Con todo esto, nadie tiene excusa para no saber lo que realmente está ocurriendo.