



Scrum Manager I

Las reglas de scrum

v. 2.5.1



Scrum Manager I

Las reglas de scrum

Versión. 2.5.1 – Abril 2015

Diseño de cubierta: Scrum Manager.

Imagen derivada de la original: The Albert Bridge 04 – Belfast de William Murphy (<http://www.streetsofdublin.com/>)

© 2015 Juan Palacio

© De la edición: Scrum Manager®

Información de derechos y licencia de uso:



<http://www.safecreative.org/work/1404240651012>

Contenido

<i>Contenido</i>	5
<i>Formación Scrum Manager</i>	7
Servicios de formación y asesoría Scrum Manager	7
<i>Mejora continua y control de calidad Scrum Manager</i>	9
PRIMERA PARTE	11
<i>Agilidad</i>	12
<i>El Manifiesto Ágil</i>	12
Los 12 principios del manifiesto ágil	14
<i>Origen de scrum.</i>	14
<i>Adopciones de scrum: técnica y pragmática</i>	16
<i>Introducción al modelo</i>	17
<i>Gestión de la evolución del proyecto</i>	17
Revisión de las Iteraciones	17
Desarrollo incremental	17
Autoorganización	18
Colaboración	18
Scrum técnico	21
<i>Scrum técnico</i>	22
<i>Artefactos</i>	22
Pila del producto y pila del sprint: los requisitos en desarrollo ágil.	23
Pila del producto: los requisitos del cliente	24
Pila del Sprint	26
El Incremento	27
<i>Eventos</i>	27
Planificación del sprint	28
Scrum diario	30
Revisión del sprint	31
Retrospectiva	32
<i>Roles</i>	32
Propietario del producto	33
Equipo de desarrollo	34
Scrum Master	34
<i>Cultura y Valores</i>	35
Medición y estimación ágil	37
<i>¿Por qué medir?</i>	38

<i>Flexibilidad y sentido común</i>	38
<i>Criterios para el diseño y aplicación de métricas</i>	39
Cuantas menos, mejor	39
¿El indicador es apropiado para el fin que se debe conseguir?	39
<i>Velocidad, trabajo y tiempo</i>	41
Tiempo	41
Trabajo	42
Trabajo ya realizado	42
Trabajo pendiente de realizar	42
Unidades de trabajo	44
Velocidad	44
<i>Medición: usos y herramientas</i>	45
Gráfico de producto.	45
Gráfico de avance: monitorización del sprint	48
Estimación de póquer	51
<i>Bibliografía</i>	55
<i>Tabla de ilustraciones</i>	57
<i>Índice</i>	58

Formación Scrum Manager

Este libro cubre los niveles I y II del conocimiento troncal de formación Scrum Manager®

Formación
Scrum Manager

Nivel I: Scrum técnico

Para aprender las reglas de scrum

Nivel II: Scrum pragmático

Para aprender a romper las reglas de scrum

Los contenidos de formación se mantienen regularmente actualizados. Puede descargar la última versión, o consultarla en línea en la dirección: <http://www.scrummanager.net/bok>

Este documento es un **recurso educativo abierto (OER)** y puede emplearse gratuitamente para consulta y autoformación.

No está permitido su uso para actividades comerciales.



Los recursos educativos abiertos de Scrum Manager son posibles gracias al soporte de los:

Servicios de formación y asesoría Scrum Manager

- **Cursos** en convocatorias presenciales, o a medida para empresas o grupos.
 - Puede consultar el calendario de convocatorias en diferentes ciudades en la página de cursos de <http://www.scrummanager.net>:
 - Para información de cursos a medida: contacte con un Centro Oficial de Formación Scrum Manager (<http://scrummanager.net/centros>) o en la dirección admin@scrummanager.net
- **Exámenes presenciales de certificación.** con supervisión de la ejecución de la prueba, e identificación del alumno
 - Incluidos en los cursos presenciales.
 - Examen de certificación sin curso: en [centros examinadores](#).

Más información:

<http://www.scrummanager.net> – (preguntas frecuentes)
<http://www.scrummanager.net/oks>
admin@scrummanager.net

Mejora continua y control de calidad Scrum Manager

Gracias por elegir los servicios de formación de Scrum Manager

Su valoración es el criterio del control de calidad de Scrum Manager, y decide la validez o no de los servicios de formación, y en su caso la continuidad de los cursos, centros y profesores.

Si ha participado en una actividad de formación auditada por Scrum Manager, le rogamos y agradecemos que valore la calidad del material, profesor, temario, etc. así como tus comentarios y sugerencias.

Scrum Manager anonimiza la información recibida, de forma que comparte con los profesores, y centros autorizados las valoraciones y aspectos de mejora, pero en ningún caso los nombres de los alumnos que las han realizado.

Puede realizar la valoración en la página accesible a miembros de Scrum Manager:

<http://scrummanager.net/qa>.

PRIMERA PARTE

Las reglas de scrum.

Agilidad

El entorno de trabajo de las empresas del conocimiento se parece muy poco al que originó la gestión de proyectos predictiva. Ahora se necesitan estrategias para el lanzamiento de productos orientadas a la entrega temprana de resultados tangibles, y a la respuesta ágil y flexible, necesaria para trabajar en mercados de evolución rápida.

Ahora se construye el producto mientras se modifican y aparecen nuevos requisitos. El cliente parte de una visión medianamente clara, pero el nivel de innovación que requiere, y la velocidad a la que se mueve su sector de negocio, no le permiten predecir con detalle cómo será el resultado final.

Quizá ya no hay “productos finales”, sino productos en continua evolución y mejora.

La gestión de proyectos ágil no se formula sobre la necesidad de anticipación, sino sobre la de adaptación continua.

¿La gestión de proyectos predictiva es la única posible? ¿Los criterios para determinar el éxito son siempre el cumplimiento de fechas y costos? ¿Puede haber proyectos cuya gestión no busque realizar un trabajo previamente planificado, con un presupuesto y en un tiempo previamente calculado?

Hoy hay directores de producto que no necesitan conocer cuáles van a ser las 200 funcionalidades que tendrá el producto final, ni si este estará terminado en 12 o en 16 meses.

Hay clientes que necesitan disponer de una primera versión con funcionalidades básicas en cuestión de semanas, y no un producto completo dentro de uno o dos años. Clientes cuyo interés es poner en el mercado rápidamente un concepto nuevo, y desarrollar de forma continua su valor.

Hay proyectos que no necesitan gestionar el seguimiento de un plan, y que fracasan por haber empleado un modelo de gestión inapropiado.

La mayoría de los fracasos se producen por aplicar ingeniería secuencial y gestión predictiva tanto en el proceso de adquisición (requisitos, contratación, seguimiento y entrega) como en la gestión del proyecto, en productos que no necesitan tanto garantías de previsibilidad en la ejecución, como respuesta rápida y flexibilidad para funcionar en entornos de negocio que cambian y evolucionan rápidamente.

El Manifiesto Ágil

En marzo de 2001, 17 críticos de los modelos de producción basados en procesos, convocados por Kent Beck, que había publicado un par de años antes el libro en el que explicaba la nueva metodología Extreme Programming (Beck, 2000) se reunieron en Salt Lake City para discutir sobre el desarrollo de software. En la reunión se acuñó el término “Métodos Ágiles” para definir a aquellos que estaban surgiendo como alternativa a las metodologías formales: CMM-SW, (precursor de CMMI) PMI, SPICE (proyecto inicial de ISO 15504), a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas, previas al desarrollo.

Los integrantes de la reunión resumieron en cuatro postulados lo que ha quedado denominado como “Manifiesto Ágil”, que son los valores sobre los que se asientan estos métodos.

Hasta 2005, entre los defensores de los modelos de procesos y los de modelos ágiles fueron frecuentes las posturas radicales, más ocupadas en descalificar al otro, que en estudiar sus métodos y conocerlos para mejorar los propios.

Manifiesto Ágil

Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan.

Con este trabajo hemos llegado a valorar:

- *A los individuos y su interacción, por encima de los procesos y las herramientas.*
- *El software que funciona, por encima de la documentación exhaustiva.*
- *La colaboración con el cliente, por encima de la negociación contractual.*
- *La respuesta al cambio, por encima del seguimiento de un plan.*

Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.

Valoramos más a los individuos y su interacción que a los procesos y las herramientas.

Este es el valor más importante del manifiesto.

Por supuesto que los procesos ayudan al trabajo. Son una guía de operación. Las herramientas mejoran la eficiencia, pero hay tareas que requieren talento y necesitan personas que lo aporten y trabajen con una actitud adecuada.

La producción basada en procesos persigue que la calidad del resultado sea consecuencia del *know-how* "explicitado" en los procesos, más que en el conocimiento aportado por las personas que los ejecutan.

Sin embargo en desarrollo ágil los procesos son una ayuda. Un soporte para guiar el trabajo. La defensa a ultranza de los procesos lleva a afirmar que con ellos se pueden conseguir resultados extraordinarios con personas mediocres, y lo cierto es que este principio no es cierto cuando se necesita creatividad e innovación.

Valoramos más el software que funciona que la documentación exhaustiva.

Poder anticipar cómo será el funcionamiento del producto final, observando prototipos previos, o partes ya elaboradas ofrece un "*feedback*" estimulante y enriquecedor, que genera ideas imposibles de concebir en un primer momento, y difícilmente se podrían incluir al redactar un documento de requisitos detallado en el comienzo del proyecto.

El manifiesto ágil no da por inútil la documentación, sólo la de la documentación innecesaria. Los documentos son soporte de hechos, permiten la transferencia del conocimiento, registran información histórica, y en muchas cuestiones legales o normativas son obligatorios, pero su relevancia debe ser mucho menor que el producto final.

La comunicación a través de documentos no ofrece la riqueza y generación de valor que logra la comunicación directa entre las personas, y a través de la interacción con prototipos del producto.

Por eso, siempre que sea posible debe preferirse reducir al mínimo indispensable el uso de documentación, que requiere trabajo sin aportar un valor directo al producto.

Si la organización y los equipos se comunican a través de documentos, además de ocultar la riqueza de la interacción con el producto, forman barreras de burocracia entre departamentos o entre personas.

Valoramos más la colaboración con el cliente que la negociación contractual.

Las prácticas ágiles están indicadas para productos cuyo detalle resulta difícil prever al principio del proyecto; y si se detallara al comenzar, el resultado final tendría menos valor que si se mejoran y precisan con retroinformación continua durante el.

También son apropiadas cuando se prevén requisitos inestables por la velocidad de cambio en el entorno de negocio del cliente.

El objetivo de un proyecto ágil no es controlar la ejecución conforme a procesos y cumplimiento de planes, sino proporcionar el mayor valor posible al producto.

Resulta por tanto más adecuada una relación de implicación y colaboración continua con el cliente, más que una contractual de delimitación de responsabilidades.

Valoramos más la respuesta al cambio que el seguimiento de un plan

Para desarrollar productos de requisitos inestables, que tienen como factor inherente el cambio y la evolución rápida y continua, resulta mucho más valiosa la capacidad de respuesta que la de seguimiento y aseguramiento de planes. Los principales valores de la gestión ágil son la anticipación y la adaptación, diferentes a los de la gestión de proyectos ortodoxa: planificación y control que evite desviaciones del plan.

Los 12 principios del manifiesto ágil

El manifiesto ágil, tras los postulados de estos cuatro valores en los que se fundamenta, establece estos 12 principios:

1. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se doblan al cambio como ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la principal medida del progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo que se hace, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se autoorganizan.
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

Origen de scrum.

Scrum es un modelo de desarrollo ágil caracterizado por:

- **Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.**
- **Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.**
- **Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada.**

Este modelo fue identificado y definido por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80, al analizar cómo desarrollaban los nuevos productos las principales empresas de manufactura tecnológica: Fuji-Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard (Nonaka & Takeuchi, The New New Product Development Game, 1986)

En su estudio, Nonaka y Takeuchi compararon la nueva forma de trabajo en equipo, con el avance en formación de scrum de los jugadores de Rugby, a raíz de lo cual quedó acuñado el término “scrum” para referirse a ella.

Aunque esta forma de trabajo surgió en empresas de productos tecnológicos, es apropiada para proyectos con requisitos inestables y para los que requieren rapidez y flexibilidad, situaciones frecuentes en el desarrollo de determinados sistemas de software.

En 1995 Ken Schwaber presentó “Scrum Development Process” en OOPSLA 95 (Object-Oriented Programming Systems & Applications conference) (SCRUM Development Process), un marco de reglas para desarrollo de software, basado en los principios de scrum, y que él había empleado en el desarrollo de Delphi, y Jeff Sutherland en su empresa Easel Corporation (compañía que en los macrojuegos de compras y fusiones, se integraría en VMARK, y luego en Informix y finalmente en Ascential Software Corporation).

Adopciones de scrum: técnica y pragmática

Scrum se puede adoptar de forma técnica, aplicando reglas definidas, o pragmática, adoptando los valores originales scrum con reglas personalizadas. ..

Esta primera parte del temario (Scrum Manager I) enseña scrum técnico, basado en la aplicación de reglas concretas en un marco de roles, eventos y artefactos definidos. El aprendizaje de scrum técnico es el primer paso aconsejable para familiarizarse con la agilidad.

Una vez iniciados en agilidad, y con el conocimiento que el propio equipo acumula a través de las retrospectivas, se pueden ir “rompiendo” las reglas y adoptar scrum pragmático, personalizado y más adecuado a las propias circunstancias del propio equipo y proyecto.

Scrum técnico

Reglas



Marco de reglas para desarrollo de software

Autores: Ken Schwaber y Jeff Sutherland
“Scrum Development Process OOPSLA'95” 1995

Aplicación de reglas definidas

Roles

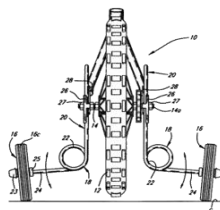
- Dueño de producto
- Equipo de desarrollo
- Scrum Master

Eventos

- El Sprint
- Reunión de planificación
- Scrum diario
- Revisión de sprint
- Retrospectiva de sprint

Artefactos

- Pila de product
- Pila de sprint
- Incremento



Aprender las reglas de Scrum

Scrum pragmático

Valores



Concepto original Scrum

Autores: Hirotaka Takeuchi e Ikujiro Nonaka
“The New New Product Development Game” 1986

Aplicación de valores ágiles

- Personas > procesos
- Resultado > documentación
- Colaboración > negociación
- Cambio > planificación

... “Para avanzar en Scrum”

- Incertidumbre
- Autoorganización
- Fases de desarrollo solapadas
- “Multiaprendizaje”
- Control sutil
- Difusion del conocimiento



Aprender a avanzar en Scrum sin reglas

Introducción al modelo

El marco técnico de scrum, por su sencillez, resulta apropiado para equipos y organizaciones que quieren comenzar a “avanzar en scrum”

Está formado por un conjunto de prácticas y reglas que resultan válidos para dar respuesta a los siguientes principios de desarrollo ágil:

- Gestión evolutiva del avance, en lugar de la tradicional o predictiva.
- Trabajar basando la calidad del resultado en el **conocimiento tácito de las personas**, más que en el explícito de los procesos y la tecnología empleada.
- Estrategia de **desarrollo incremental a través de iteraciones** (sprints) y revisiones.
- Seguir los pasos del desarrollo ágil: desde el concepto o visión general de la necesidad del cliente, construcción del producto de forma incremental a través de iteraciones breves que comprenden fases de especulación – exploración y revisión. Estas iteraciones (en scrum llamadas sprints) se repiten de forma continua hasta que el cliente da por cerrada la evolución del producto.

Se comienza con la visión general de lo que se desea obtener, y a partir de ella se especifica y da detalle a las partes de mayor prioridad, y que se desean tener cuanto antes.

Cada ciclo de desarrollo o iteración (**sprint**) finaliza con la entrega de una parte operativa del producto (**incremento**). La duración de cada sprint puede ser desde una, hasta seis semanas, aunque se recomienda que no excedan de un mes.

En scrum, el equipo monitoriza la evolución de cada sprint en reuniones breves diarias donde se revisa en conjunto el trabajo realizado por cada miembro el día anterior, y el previsto para el día en curso. Esta reunión diaria es de tiempo prefijado de 5 a 15 minutos máximo, se realiza de pie junto a un tablero o pizarra con información de las tareas del sprint, y el trabajo pendiente en cada una. Esta reunión se denomina “reunion de pie” o “scrum diario” y si se emplea la terminología inglesa: “stand-up meeting”, también: “daily scrum” o “morning rollcall”.

Gestión de la evolución del proyecto

Scrum maneja de forma empírica la evolución del proyecto con las siguientes tácticas:

Revisión de las Iteraciones

Al finalizar cada sprint se revisa funcionalmente el resultado, con todos los implicados en el proyecto. Es por tanto la duración del sprint, el período de tiempo máximo para descubrir planteamientos erróneos, mejorables o malinterpretaciones en las funcionalidades del producto

Desarrollo incremental

No se trabaja con diseños o abstracciones durante toda la construcción del producto.

El desarrollo incremental ofrece al final de cada iteración una parte de producto operativa, que se puede usar, inspeccionar y evaluar.

Scrum resulta adecuado en proyectos con requisitos inciertos y, o inestables.

¿Por qué predecir la versión definitiva de algo que va a estar evolucionando de forma continua? scrum considera a la inestabilidad como una premisa, y adopta técnicas de trabajo para facilitar la evolución sin degradar la calidad de la arquitectura y permitir que también evolucione durante el desarrollo.

Durante la construcción se depura el diseño y la arquitectura, y no se cierran en una primera fase del proyecto. Las distintas fases que el desarrollo en cascada realiza de forma secuencial, en scrum se solapan y realizan de forma continua y simultánea.

Autoorganización

Son muchos los factores impredecibles en un proyecto. La gestión predictiva asigna al rol de gestor del proyecto la responsabilidad de su gestión y resolución.

En scrum los equipos son autoorganizados, con un margen de maniobra suficiente para tomar las decisiones que consideren oportunas.

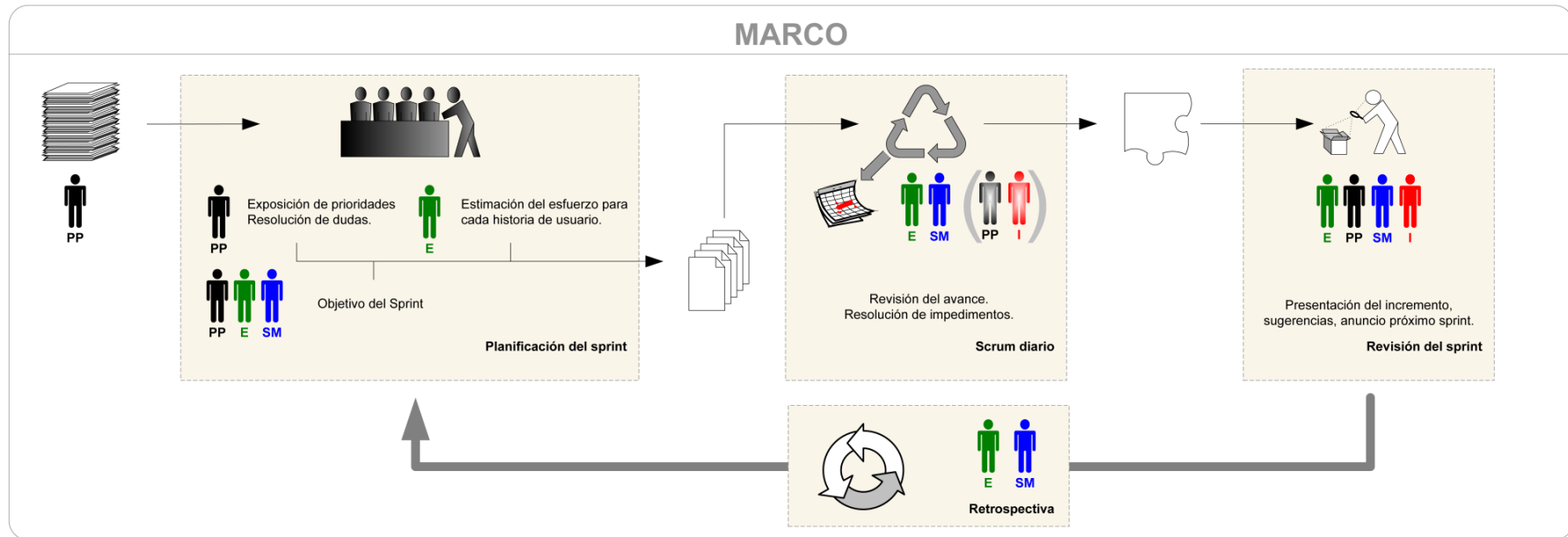
Colaboración

Es un componente importante y necesario para que a través de la autoorganización se pueda gestionar con solvencia la labor que de otra forma realizaría un gestor de proyectos.

Todos los miembros del equipo colaboran de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

LAS REGLAS DE SCRUM

Rev.1.1



ROLES



PROPIETARIO DEL PRODUCTO

Determina las prioridades.
Una sola persona.



EQUIPO DE DESARROLLO

Construye el producto.



SCRUM MASTER

Gestiona y facilita la ejecución de las reglas de Scrum



INTERESADOS

Resto de implicados. Asesoran y observan.

ARTEFACTOS



PILA DEL PRODUCTO

Relación de requisitos del producto, no es necesario excesivo detalle. Priorizados. Lista en evolución y abierta a todos los roles. El propietario del producto es su responsable y quien decide.



PILA DEL SPRINT

Requisitos comprometidos por el equipo para el sprint con nivel de detalle suficiente para su ejecución.



INCREMENTO

Parte del producto desarrollada en un sprint, en condiciones de ser usada (pruebas, codificación limpia y documentada).

EVENTOS



PLANIFICACIÓN DEL SPRINT

1 jornada de trabajo (máx.). El propietario del producto explica las prioridades. El equipo estima el esfuerzo de los requisitos prioritarios y se elabora la pila del sprint. El equipo define en una frase el objetivo del sprint.



SPRINT

Ciclo de desarrollo básico en el marco estándar de scrum, de duración recomendada inferior a un mes y nunca mayor de 6 semanas.



SCRUM DIARIO

15 minutos máximo. Responsabilidad del equipo. Cada miembro expone: Lo que hizo ayer. Lo que va a hacer hoy, si tiene o prevé problemas. Se actualiza la pila del sprint.



REVISIÓN DEL SPRINT

Informativa, máx. 4 horas, presentación del incremento, planteamiento de sugerencias y anuncio del próximo sprint.



RETROSPECTIVA

El equipo autoanaliza la forma de trabajo. Identificación de fortalezas y debilidades. Refuerzo de las primeras, plan de mejora de las segundas.

Ilustración 1: Marco scrum técnico

Scrum técnico

Scrum técnico

El marco técnico de scrum está formado por:

- **Roles:**
 - El equipo scrum.
 - El dueño del producto.
 - El Scrum Master.
- **Artefactos:**
 - Pila del producto.
 - Pila del sprint.
 - incremento.
- **Eventos**
 - Sprint.
 - Reunión de planificación del sprint.
 - Scrum diario.
 - Revisión del sprint.
 - Retrospectiva del sprint.

Y la pieza clave es el **sprint**.

Se denomina sprint a cada ciclo o iteración de trabajo que produce una parte del producto terminada y funcionalmente operativa (**incremento**)

Como se verá más tarde, al abordar scrum pragmático, las implementaciones más flexibles de scrum pueden adoptar dos tácticas diferentes para mantener un avance continuo en el proyecto:

- **Incremento iterativo:** basado en pulsos de tiempo prefijado (timeboxing)
- **Incremento continuo:** basado en el mantenimiento de un flujo continuo, no marcado por pulsos o sprints.

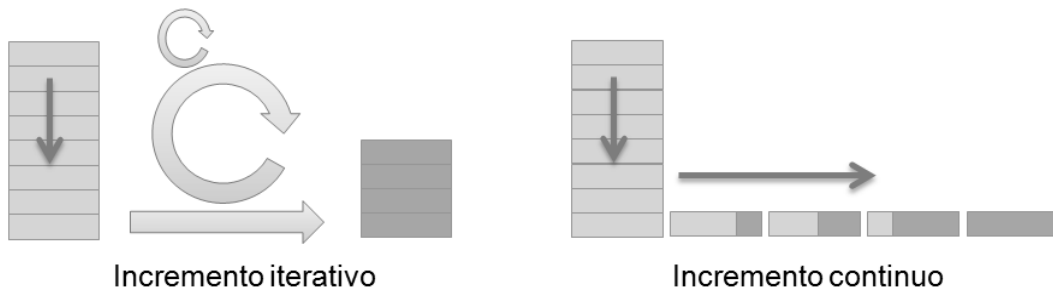


Ilustración 2: Incremento iterativo / continuo

Al usar scrum técnico se trabaja con sprints, y por tanto con incremento iterativo.

Artefactos

- **Pila del producto:** (product backlog) lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.
- **Pila del sprint:** (sprint backlog) lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- **Incremento:** resultado de cada sprint.

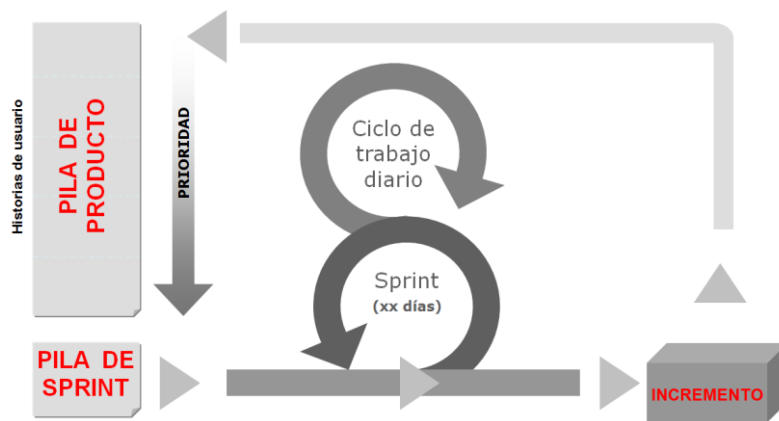


Ilustración 3: Diagrama del ciclo iterativo scrum

Otro artefacto propio del modelo estándar de scrum es el gráfico de avance o gráfico *burn down* que el equipo actualiza a diario para comprobar el avance. Este elemento, junto con la práctica de estimación de póquer y el gráfico de producto o *burn up* se encuentra incluido en el capítulo de Métricas Ágiles.

Pila del producto y pila del sprint: los requisitos en desarrollo ágil.

La ingeniería del software clásica diferencia dos ámbitos de requisitos:

- Requisitos del sistema
- Requisitos del software

Los requisitos del sistema forman parte del proceso de adquisición, y por tanto es responsabilidad del cliente la definición del problema y de las funcionalidades que debe aportar la solución.

No importa si se trata de gestión tradicional o ágil. La pila del producto es responsabilidad del cliente, aunque se aborda de forma diferente en cada caso.

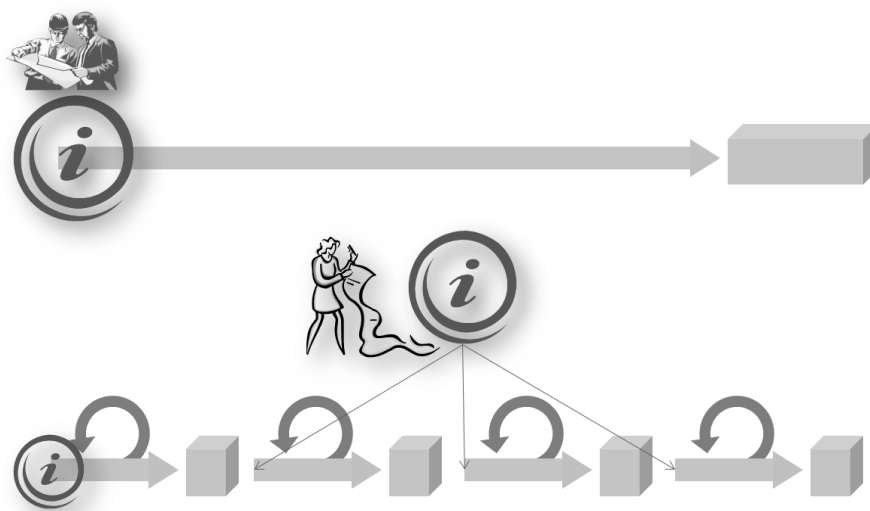


Ilustración 4: Requisitos completos / evolutivos

- En los proyectos predictivos, los requisitos del sistema suelen especificarse en documentos formales; mientras que en los proyectos ágiles toman la forma de pila del producto o lista de historias de usuario.

- Los requisitos del sistema formales se especifican de forma completa y cerrada al inicio del proyecto; sin embargo una pila del producto es un documento vivo, que evoluciona durante el desarrollo.
- Los requisitos del sistema los desarrolla una persona o equipo especializado en ingeniería de requisitos a través del proceso de obtención (elicitación) con el cliente. En scrum la visión del cliente es conocida por todo el equipo (el cliente colabora con el equipo de desarrollo) y la pila del producto se realiza y evoluciona de forma continua con los aportes de todos.

Scrum, aplicado al software, emplea dos formatos para registrar los requisitos:

- Pila del producto (Product Backlog)
- Pila del sprint (Sprint Backlog)

La pila del producto registra los requisitos vistos desde el punto de vista del cliente. Un enfoque similar al de los requisitos del sistema o "ConOps" de la ingeniería tradicional. Está formada por la lista de funcionalidades o "historias de usuario" que desea obtener el cliente, ordenadas por la prioridad que el mismo le otorga a cada una.

La pila del sprint refleja los requisitos vistos desde el punto de vista del equipo de desarrollo. Está formada por la lista de tareas en las que se descomponen las historias de usuario que se van a llevar a cabo en el sprint.

En el desarrollo y mantenimiento de la pila del producto lo relevante no es tanto el formato, sino que:

- Las funcionalidades que incluye den forma a una visión del producto definida y conocida por todo el equipo.
- Las funcionalidades estén individualmente definidas, priorizadas y pre-estimadas.
- Esté realizada y gestionada por el cliente (propietario del producto).

Pila del producto: los requisitos del cliente

La pila del producto es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de los sucesivos sprints.

Representa todo aquello que esperan el cliente, los usuarios, y en general los interesados. Todo lo que suponga un trabajo que debe realizar el equipo debe estar reflejado en esta pila.

Estos son algunos ejemplos de posibles entradas a una pila de producto:

- Permitir a los usuarios la consulta de las obras publicadas por un determinado autor.
- Reducir el tiempo de instalación del programa.
- Mejorar la escalabilidad del sistema.
- Permitir la consulta de una obra a través de un API web.

La pila de requisitos del producto nunca se da por completada; está en continuo crecimiento y evolución. Al comenzar el proyecto incluye los requisitos inicialmente conocidos y mejor entendidos, y conforme avanza el desarrollo, y evoluciona el entorno en el que será usado, se va desarrollando.

En definitiva su continuo dinamismo refleja aquello que el producto necesita incorporar para ser el más adecuado a las circunstancias, en todo momento.

Para comenzar el desarrollo se necesita la visión del objetivo de negocio que se quiere conseguir con el proyecto, comprendida y conocida por todo el equipo, y elementos suficientes en la pila para llevar a cabo el primer sprint.

Habitualmente se comienza a elaborar la pila con el resultado de una reunión de "tormenta de ideas", o "fertilización cruzada", o un proceso de "Exploración" (eXtreme Programming) donde colabora todo el equipo partiendo de la visión del propietario del producto.

El formato de la visión no es relevante. Según los casos, puede ser una presentación informal del responsable del producto, un informe de requisitos del departamento de marketing, u otros.

Sin embargo, sí es importante disponer de una visión real, comprendida y compartida por todo el equipo.

El propietario del producto mantiene la pila ordenada por la prioridad de los elementos, siendo los más prioritarios los que confieren mayor valor al producto, o por alguna razón resultan más necesarios, y determinan las actividades de desarrollo inmediatas.

El detalle de los requisitos en la pila del producto debe ser proporcional a la prioridad: Los elementos de mayor prioridad deben tener mayor nivel de comprensión y detalle que los del resto. De esta forma el equipo de desarrollo puede descomponer un elemento de prioridad alta en tareas con la precisión suficiente para ser hecho en un sprint.

Los elementos de la pila del producto que pueden ser incorporados a un sprint se denominan “preparados” o “accionables” y son los que pueden seleccionarse en la reunión de planificación del sprint.

Preparación de la pila del producto

Se denomina “preparación” (grooming) de la pila del producto a las actividades de priorización, detalle y estimación de los elementos que la componen. Es un proceso que realizan de forma puntual, en cualquier momento, continua y colaborativa el propietario del producto y el equipo de desarrollo. No debe consumir más del 10% de la capacidad de trabajo del equipo.

La responsabilidad de estimar el esfuerzo previsible para cada elemento, es de las personas del equipo que previsiblemente harán el trabajo.

Formato de la pila del producto

Scrum prefiere la comunicación verbal o de visualización directa, a la escrita.

La pila del producto no es un documento de requisitos, sino una herramienta de referencia para el equipo.

Si se emplea formato de lista, es recomendable que al menos incluya la siguiente información para cada elemento:

- Identificador único de la funcionalidad o trabajo.
- Descripción de la funcionalidad/requisito, denominado “historia de usuario”.
- Campo o sistema de priorización.
- Estimación del esfuerzo necesario.

Dependiendo del tipo de proyecto, funcionamiento del equipo y la organización, pueden ser aconsejables otros campos:

- Observaciones.
- Criterio de validación.
- Persona asignada.
- Nº de Sprint en el que se realiza.
- Módulo del sistema al que pertenece.
- Entre otros.

Es preferible no adoptar formatos rígidos. Los resultados de scrum no dependen de las formas, sino de la institucionalización de sus principios y la implementación adecuada a las características de la empresa y del proyecto. He aquí un sencillo ejemplo de pila de producto:

Id	Prioridad	Descripción	Est.	Por
1	Muy alta	Plataforma tecnológica	30	AR
2	Muy Alta	Interfaz de usuario	40	LM
3	Muy Alta	Un usuario se registra en el sistema	40	LM
4	Alta	El operador define el flujo y textos de un expediente	60	AR
5	Alta	xxx	999	CC

Ilustración 5: Ejemplo de pila de producto

Pila del Sprint

La pila del sprint (*sprint Backlog*) es la lista que descompone las funcionalidades de la pila del producto (historias de usuario) en las tareas necesarias para construir un incremento: una parte completa y operativa del producto.

La realiza el equipo durante la reunión de planificación del sprint, autoasignando cada tarea a un miembro del equipo, e indicando en la misma lista cuánto tiempo o esfuerzo se prevé que falta para terminarla.

La pila del sprint descompone el trabajo en unidades de tamaño adecuado para monitorizar el avance a diario, e identificar riesgos y problemas sin necesidad de procesos de gestión complejos.

Es también una herramienta para la comunicación visual directa del equipo.

Condiciones

- Realizada de forma conjunta por todos los miembros del equipo.
- Cubre todas las tareas identificadas por el equipo para conseguir el objetivo del sprint.
- Sólo el equipo la puede modificar durante el sprint.
- Las tareas demasiado grandes deben descomponerse en otras más pequeñas. Se deben considerar “grandes” las tareas que necesitan más de un día para realizarse.
- Es visible para todo el equipo. Idealmente en un tablero o pared en el mismo espacio físico donde trabaja el equipo.

Formato y soporte

Son soportes habituales:

- Tablero físico o pared.
- Hoja de cálculo.
- Herramienta colaborativa o de gestión de proyectos.

Y sobre el más adecuado a las características del proyecto, oficina y equipo, lo apropiado es diseñar el formato más cómodo para todos, teniendo en cuenta los siguientes criterios:

- Incluir la siguiente información: Pila del sprint, persona responsable de cada tarea, estado en el que se encuentra y tiempo de trabajo que queda para completarla.
- Incluir sólo la información estrictamente necesaria.
- Debe servir de medio para registrar en cada reunión diaria del sprint, el tiempo que le queda a cada tarea.
- Facilitar la consulta y la comunicación diaria y directa del equipo.

Ejemplo:

PROYECTO					Sáb 07 Ene	Dom 08 Ene	Lun 09 Ene	Mar 10 Ene	Mié 11 Ene
Inicio	Fin	Jornada							
7-ene-12	1-abr-12	40 hs							
Tareas pendientes					15	15	14	14	11
Horas pendientes					172	162	148	142	124
Fecha de Cierre					12 ene	12 ene	12 ene	13 ene	16 ene

PILA DEL PRODUCTO					OBJETIVO DEL SPRINT			
Categoría	Tarea	Responsable	Estimado en horas	Estado	Crear y publicar versión básica del sitio web público			
Diseño	Crear diseño de base de datos	Juan	24	Completo	24	16	8	4
Diseño	Validar diseño de base de datos	Pedro	4	Completo	4	4	4	4
Desarrollo	Contratar servicio de hosting	Pedro	4	Completo	4	2		
Desarrollo	Crear layout y estilos de sitio web	Maria	16	Activo	8	8	4	2
Desarrollo	Crear página principal	Maria	24	Pendiente	24	24	24	24
Desarrollo	Mostrar resúmenes de noticias por sección	Juan	16	Pendiente	16	16	16	8
Desarrollo	Crear banners de publicidad	Luis	24	Pendiente	24	24	24	24
Desarrollo	Visualizar un Artículo	Luis	8	Pendiente	8	8	8	8
Desarrollo	Imprimir un Artículo	Luis	4	Pendiente	4	4	4	4

Ilustración 6: Ejemplo de pila de sprint con hoja de cálculo

Durante el sprint, el equipo actualiza a diario en ella los tiempos pendientes de cada tarea. Al mismo tiempo, con estos datos traza el gráfico de avance o trabajo consumido (burn-down), que se describe más adelante, en el capítulo de métricas ágiles.

El Incremento

El incremento es la parte de producto producida en un sprint, y tiene como característica el estar completamente terminada y operativa, en condiciones de ser entregada al cliente.

No se deben considerar como Incremento a prototipos, módulos o sub-módulos, ni partes pendientes de pruebas o integración.

Idealmente en scrum:

- Cada elemento de la pila del producto se refiere a funcionalidades entregables, no a trabajos internos del tipo “diseño de la base de datos”.
- Se produce un “incremento” en cada iteración.

Sin embargo es una excepción frecuente el primer sprint. En el que objetivos del tipo “contrastar la plataforma y el diseño” pueden resultar necesarios, e implican trabajos de diseño o desarrollo de prototipos para contrastar las expectativas de la plataforma o tecnología que se va a emplear. Teniendo en cuenta esta excepción habitual:

Incremento es la parte de producto realizada en un sprint potencialmente entregable: **terminada y probada**.

Si el proyecto o el sistema requiere documentación, o procesos de validación y verificación documentados, o con niveles de independencia que implican procesos con terceros, éstos también tienen que estar realizados para considerar que el incremento está “hecho”.

Eventos

- **Sprint:** nombre que recibe cada iteración de desarrollo. Es el núcleo central que genera el pulso de avance por tiempos prefijados (time boxing).
- **Reunión de Planificación del sprint:** reunión de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el objetivo del sprint y las tareas necesarias para conseguirlo.
- **Scrum diario:** breve reunión diaria del equipo, en la que cada miembro responde a tres cuestiones:

- 1.- El trabajo realizado el día anterior.
- 2.- El que tiene previsto realizar.
- 3.- Cosas que puede necesitar o impedimentos que deben eliminarse para poder realizar el trabajo.

Cada persona actualiza en la pila del sprint el tiempo o esfuerzo pendiente de sus tareas, y con esta información se actualiza a su vez el gráfico con el que el equipo monitoriza el avance del sprint (burn-down)

- **Revisión del sprint:** análisis e inspección del incremento generado, y adaptación de la pila del producto si resulta necesario.

Una cuarta reunión se incorporó al marco estándar de scrum en la primera década de 2.000:

- **Retrospectiva del sprint:** revisión de lo sucedido durante el Sprint. Reunión en la que el equipo analiza aspectos operativos de la forma de trabajo y crea un plan de mejoras para aplicar en el próximo sprint.

Planificación del sprint

Descripción

En esta reunión se toman como base las prioridades y necesidades de negocio del cliente, y se determinan cuáles y cómo van a ser las funcionalidades que se incorporarán al producto en el siguiente sprint.

Se trata de una reunión conducida por el responsable del funcionamiento del marco scrum (Scrum Master en scrum técnico, o un miembro del equipo, en scrum pragmático) a la que deben asistir el propietario del producto y el equipo completo, y a la que también pueden asistir otros implicados en el proyecto.

La reunión puede durar una jornada de trabajo completa, cuando se trata de planificar un sprint largo (de un mes de duración) o un tiempo proporcional para planificar un sprint más breve.

Esta reunión debe dar respuesta a dos cuestiones:

- Qué se entregará al terminar el sprint.
- Cuál es el trabajo necesario para realizar el incremento previsto, y cómo lo llevará a cabo el equipo.

La reunión se articula en dos partes de igual duración, para dar respuesta a una de estas cuestiones, en cada una.

Precondiciones

- La organización tiene determinados los recursos disponibles para llevar a cabo el sprint.
- Ya están “preparados” los elementos prioritarios de la pila del producto, de forma que ya tienen un nivel de detalle suficiente y una estimación previa del trabajo que requieren.
- El equipo tiene un conocimiento de las tecnologías empleadas, y del negocio del producto suficiente para realizar estimaciones basadas en juicio de expertos, y para comprender los conceptos del negocio que expone el propietario del producto.

Entradas

- La pila del producto.
- El producto desarrollado hasta la fecha en los incrementos anteriores (excepto si se trata del primer sprint).
- Dato de la velocidad o rendimiento del equipo en el último sprint, que se emplea como criterio para estimar la cantidad de trabajo que es razonable suponer para el próximo sprint.
- Circunstancias de las condiciones de negocio del cliente y del escenario tecnológico empleado.

Resultados

- Pila del sprint.
- Duración del sprint y fecha de la reunión de revisión.
- Objetivo del sprint.

Formato de la reunión

Esta reunión marca el inicio de cada sprint.

Duración máxima: un día.

Asistentes: Propietario del producto, equipo de desarrollo y Scrum Master.

Pueden asistir: todos aquellos que aporten información útil, ya que es una reunión abierta.

Consta de dos partes separadas por una pausa de café o comida, según la duración.

Primera parte: Qué se entregará al terminar el sprint.

El propietario del producto presenta la pila de producto, exponiendo los requisitos de mayor prioridad que necesita y que prevé que se podrán desarrollar en el siguiente sprint. Si la pila del producto ha tenido cambios significativos desde la anterior reunión, explica las causas que los han ocasionado.

El objetivo es que todo el equipo conozca las razones y los detalles con el nivel suficiente para comprender el trabajo del sprint.

Propietario del producto:

- Presenta las funcionalidades de la pila del producto que tienen mayor prioridad y que estima se pueden realizar en el sprint.
- La presentación se hace con un nivel de detalle suficiente para transmitir al equipo toda la información necesaria para construir el incremento.

El equipo

- Realiza las preguntas y solicita las aclaraciones necesarias.
- Propone sugerencias, modificaciones y soluciones alternativas.

Los aportes del equipo pueden suponer modificaciones en la pila.

Esta reunión es un punto caliente de scrum para favorecer la fertilización cruzada de ideas en equipo y añadir valor a la visión del producto.

Tras reordenar y replantear las funcionalidades de la pila del producto, el equipo define el “objetivo del sprint” o frase que sintetiza cuál es el valor que se le va a entregar al cliente.

Exceptuando sprints dedicados exclusivamente a refactorización o a colecciones de tareas desordenadas (que deberían ser los menos), la elaboración de este lema de forma conjunta en la reunión es una garantía de que todo el equipo comprende y comparte la finalidad del trabajo, y durante el sprint sirve de criterio de referencia en las decisiones que autogestiona el equipo.

Segunda parte: Cómo se conseguirá hacer el incremento.

El equipo desglosa cada funcionalidad en tareas, y estima el tiempo para cada una de ellas, componiendo así las tareas que forman la pila del sprint. En este desglose, el equipo tiene en cuenta los elementos de diseño y arquitectura que deberá incorporar el sistema.

Los miembros del equipo establecen cuáles van a ser las tareas para los primeros días del sprint, y se las autoasignan tomando como criterios sus conocimientos, intereses y una distribución homogénea del trabajo.

Esta segunda parte debe considerarse como una “reunión del equipo”, en la que deben estar todos sus miembros, y ser ellos quienes descompongan estimen y asignen el trabajo.

El papel del propietario del producto es atender a dudas y comprobar que el equipo comprende y comparte su objetivo.

El Scrum Master actúa de moderador de la reunión.

Funciones del Scrum Master

El Scrum Master, o el moderador de la reunión es responsable y garante de:

- 1.- Realizar esta reunión antes de cada sprint.
- 2.- Asegurar que se cuenta con una pila de producto adecuadamente preparada por el propietario del producto.
- 3.- Ayudar a mantener el diálogo entre el propietario del producto y el equipo.
- 4.- Asegurar que se llegue a un acuerdo entre el propietario del producto y el equipo respecto de lo que incluirá el incremento.
- 5.- Ayudar al equipo a comprender la visión y necesidades de negocio del cliente.
- 6.- Asegurar que el equipo ha realizado una descomposición y estimación del trabajo realistas, y ha considerado las posibles tareas necesarias de análisis, investigación o apoyo.
- 7.- Asegurar que al final de la reunión están objetivamente determinados:
 - Los elementos de la pila del producto que se van a ejecutar.
 - El objetivo del sprint.
 - La pila del sprint con todas las tareas estimadas.
 - La duración del sprint y la fecha de la reunión de revisión.

El Scrum Master modera la reunión para que no dure más de un día. Debe evitar que el equipo comience a profundizar en trabajos de análisis o arquitectura que son propios del trabajo del sprint.

Ejemplo de tablero operativo para la reunión

Es recomendable, que el propietario del producto emplee una hoja de cálculo o alguna herramienta similar para guardar en formato digital la pila del producto. Aunque no es aconsejable usarla como base para trabajar sobre ella en la reunión.

En la reunión es preferible usar una pizarra o un tablero y fichas o etiquetas removibles.

El tablero facilita la comunicación y el trabajo de la reunión.

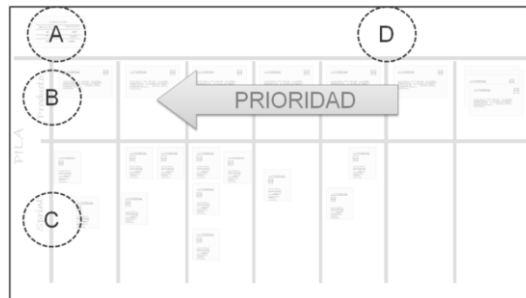


Ilustración 7: Ejemplo de pizarra de trabajo

Algunos soportes habituales:

- Pizarra blanca y notas adhesivas tipo Post-it®.
- Tablero de corcho laminado y chinchetas para sujetar las notas.
- Tablero de acero vitrificado y soportes magnéticos para sujetar notas.

Con cinta adhesiva removible se marcan líneas para delimitar:

- Un área superior donde el equipo anota:
 - (A) las unidades de trabajo que según la velocidad media del equipo se podrían realizar en sprints de 2, 3, 4 y 5 semanas.
 - (D) Duración que finalmente tendrá el sprint, así como el objetivo establecido, duración, hora fijada para las reuniones diarias y fecha prevista para la reunión de revisión del sprint.
- B.- Una franja para ordenar los elementos de la pila del producto de mayor a menor prioridad.
- C.- Una franja paralela para descomponer cada elemento de la pila del producto en las correspondientes tareas de la pila del sprint.

Scrum diario

Descripción

Reunión diaria breve, de no más de 15 minutos, en la que el equipo sincroniza el trabajo y establece el plan para las 24 horas siguientes.

Entradas

- Pila del sprint y gráfico de avance (burn-down) actualizados con la información de la reunión anterior.
- Información del avance de cada miembro del equipo.

Resultados

- Pila del sprint y gráfico de avance (burn-down) actualizados.
- Identificación de posibles necesidades e impedimentos.

Formato de la reunión

Se recomienda realizarla de pie junto a un tablero con la pila del sprint y el gráfico de avance del sprint, para que todos puedan compartir la información y anotar.

En la reunión está presente todo el equipo, y pueden asistir también otras personas relacionadas con el proyecto o la organización, aunque éstas no pueden intervenir.

En esta reunión cada miembro del equipo de desarrollo explica:

- Lo que ha logrado desde el anterior scrum diario.
- Lo que va a hacer hasta el próximo scrum diario.
- Si están teniendo algún problema, o si prevé que puede encontrar algún impedimento.

Y actualiza sobre la pila del sprint el esfuerzo que estima pendiente en las tareas que tiene asignadas, o marca como finalizadas las ya completadas.

Al final de la reunión:

- El equipo refresca el gráfico de avance del sprint, con las estimaciones actualizadas,
- El Scrum Master realiza las gestiones adecuadas para resolver las necesidades o impedimentos identificados.

El equipo es el responsable de esta reunión, no el Scrum Master; y no se trata de una reunión de “inspección” o “control” sino de comunicación entre el equipo para compartir el estado del trabajo, chequear el ritmo de avance y colaborar en posibles dificultades o impedimentos.

Revisión del sprint

Descripción

Reunión realizada al final del sprint para comprobar el incremento. .

No debe durar más de 4 horas, en el caso de revisar sprints largos. Para sprints de una o dos semanas, con una o dos horas de duración debería ser suficiente.

Objetivos:

- El propietario del producto comprueba el progreso del sistema. Esta reunión marca, a intervalos regulares, el ritmo de construcción, y la trayectoria que va tomando la visión del producto.
- El propietario del producto identifica las funcionalidades que se pueden considerar “hechas” y las que no.
- Al ver y probar el incremento, el propietario del producto, y el equipo en general obtienen feedback relevante para revisar la pila del producto.
- Otros ingenieros y programadores de la empresa también pueden asistir para conocer cómo trabaja la tecnología empleada.

Precondiciones

- Se ha concluido el sprint.
- Asiste todo el equipo de desarrollo, el propietario del producto, el Scrum Master y todas las personas implicadas en el proyecto que lo deseen.

Entradas

- Incremento terminado.

Resultados

- Feedback para el propietario del producto: hito de seguimiento de la construcción del sistema, e información para mejorar el valor de la visión del producto.
- Convocatoria de la reunión del siguiente sprint.

Formato de la reunión

Es una reunión informal. El objetivo es ver el incremento realizado. Están prohibidas las presentaciones gráficas y “powerpoints”.

El equipo no debe invertir más de una hora en desarrollar la reunión, y lo que se muestra es el resultado final: terminado, probado y operando en el entorno del cliente (incremento).

Según las características del proyecto puede incluir también documentación de usuario, o técnica.

Es una reunión informativa. **Su misión no es la toma de decisiones ni la crítica del incremento.** Con la información obtenida, posteriormente el propietario del producto tratarán las posibles modificaciones sobre la visión del producto.

Protocolo recomendado:

- 1.- El equipo expone el objetivo del sprint, la lista de funcionalidades que se incluían y las que se han desarrollado.
- 2.- El equipo hace una introducción general del sprint y demuestra el funcionamiento de las partes construidas.
- 3.- Se abre un turno de preguntas y sugerencias. Esta parte genera información valiosa para que el propietario del producto y el equipo en general, puedan mejorar la visión del producto.
- 4.- El Scrum Master, de acuerdo con las agendas del propietario del producto y el equipo, cierra la fecha para la reunión de preparación del siguiente sprint.

Retrospectiva

Reunión que se realiza tras la revisión de cada sprint, y antes de la reunión de planificación del siguiente, con una duración recomendada de una a tres horas, según la duración del sprint terminado.

En ella el equipo realiza autoanálisis de su sobre su forma de trabajar, e identifica fortalezas y puntos débiles. El objetivo es consolidar y afianzar las primeras, y planificar acciones de mejora sobre los segundos.

El hecho de que se realice normalmente al final de cada sprint lleva a veces a considerarlas erróneamente como reuniones de “revisión de sprint”, cuando es aconsejable tratarlas por separado, porque sus objetivos son diferentes.

El objetivo de la revisión del sprint es analizar “QUÉ” se está construyendo, mientras que una reunión retrospectiva se centra en “CÓMO” lo estamos construyendo: “CÓMO” estamos trabajando, con el objetivo de analizar problemas y aspectos mejorables.

Las reuniones "retrospectivas" realizadas de forma periódica por el equipo para mejorar la forma de trabajo, se consideran cada vez más un componente del marco técnico de scrum, si bien no es una reunión para seguimiento de la evolución del producto, sino para mejora del marco de trabajo.

Roles

Todas las personas que intervienen, o tienen relación directa o indirecta con el proyecto, se clasifican en dos grupos: comprometidos e implicados. En círculos de scrum es frecuente llamar a los primeros (sin ninguna connotación peyorativa) “cerdos” y a los segundos “gallinas”.

El origen de estos nombres está en la siguiente metáfora que ilustra de forma gráfica la diferencia entre “compromiso” e “implicación” en el proyecto:

Una gallina y un cerdo paseaban por la carretera. La gallina preguntó al cerdo: “¿Quieres abrir un restaurante conmigo?”.

El cerdo consideró la propuesta y respondió: “Sí, me gustaría. ¿Y cómo lo llamaríamos?”.

La gallina respondió: “huevos con jamón”.

El cerdo se detuvo, hizo una pausa y contestó: “Pensándolo mejor, creo que no voy a abrir un restaurante contigo. Yo estaría realmente comprometido, mientras que tu estarías sólo implicada”.

COMPROMETIDOS (CERDOS)	IMPLICADOS (GALLINAS)
Propietario del producto	Otros interesados (dirección, gerencias, comerciales, marketing, etc.)
Miembros del equipo	

Ilustración 8: Roles estándar de scrum

- Propietario del producto: es la persona responsable de lograr el mayor valor de producto para los clientes, usuarios y resto de implicados.
- **Equipo de desarrollo:** grupo o grupos de trabajo que desarrollan el producto.

Una observación en este punto, sobre el rol de Scrum Master, por ser en ocasiones frecuente la duda de considerar si es un rol “comprometido” o “implicado”. Partiendo de que la división entre personas comprometidas y personas implicadas es más “conceptual” que “relevante”, pero cuando se trabaja con este rol presente, su responsabilidad es el funcionamiento de un scrum técnico en la organización.

Su responsabilidad directa, su misión, es por tanto la forma de trabajo, siendo por tanto el producto elaborado en los proyectos un objetivo de segundo nivel, o indirecto.

Por esta razón en el cuadro anterior no se considera el rol de Scrum Master, aunque que en cualquier caso no es una cuestión especialmente relevante. Si hubiera que forzar una respuesta, desde el criterio de que no está comprometido en el proyecto (sino en la mejora de la forma de trabajo) se debería considerar como un rol “implicado”

Propietario del producto

El propietario del producto (*product owner*) es quien toma las decisiones del cliente. Su responsabilidad es el valor del producto.

Para simplificar la comunicación y toma de decisiones es necesario que este rol recaiga en una única persona.

Si el cliente es una organización grande, o con varios departamentos, puede adoptar la forma de comunicación interna que consideren oportuna, pero en el equipo de desarrollo sólo se integra una persona en representación del cliente, y ésta debe tener el conocimiento suficiente del producto y las atribuciones necesarias para tomar las decisiones que le corresponden.

En resumen, el propietario de producto es quien:

- Decide en última instancia cómo será el resultado final, y el orden en el que se van construyendo los sucesivos incrementos: qué se pone y qué se quita de la pila del producto, y cuál es la prioridad de las funcionalidades.
- Conoce el plan del producto, sus posibilidades y plan de inversión, así como del retorno esperado a la inversión realizada, y se responsabiliza sobre fechas y funcionalidades de las diferentes versiones del mismo.

En los desarrollos internos para la propia empresa, suele asumir este rol el *product manager* o el responsable de marketing. En desarrollos para clientes externos, el responsable del proceso de adquisición del cliente.

Según las circunstancias del proyecto es posible incluso que delegue en el equipo de desarrollo, o en alguien de su confianza, pero la responsabilidad siempre es suya.

Para ejercer este rol es necesario:

- **Conocer perfectamente el entorno de negocio del cliente**, las necesidades y el objetivo que se persigue con el sistema que se está construyendo.
- Tener la **visión del producto**, así como las necesidades concretas del proyecto, para poder priorizar eficientemente el trabajo.

- Disponer de **atribuciones y conocimiento del plan del producto suficiente** para tomar las decisiones necesarias durante el proyecto, incluidas para cubrir las expectativas previstas de retorno de la Inversión del proyecto.
- Recibir y analizar de forma continua **retroinformación del entorno de negocio** (evolución del mercado, competencia, alternativas) y del proyecto (sugerencias del equipo, alternativas técnicas, pruebas y evaluación de cada incremento).

Es además recomendable que el propietario de producto:

- Conozca scrum para realizar con solvencia las tareas que le corresponden:
 - Desarrollo y administración de la pila del producto.
 - Exposición de la visión e historias de usuario, y participación en la reunión de planificación de cada sprint.
- Conozca y haya trabajado previamente con el mismo equipo.

La organización debe respetar sus decisiones y no modificar prioridades ni elementos de la pila del producto.

Equipo de desarrollo

Lo forman el grupo de profesionales que realizan el incremento de cada sprint.

Se recomienda que un equipo scrum tenga entre 3 y 8 personas. Más allá de 8 resulta más difícil mantener la comunicación directa, y se manifiestan con más intensidad los roces habituales de la dinámica de grupos (que comienzan a aparecer a partir de 6 personas). En el cómputo del número de miembros del equipo *de desarrollo* no se consideran ni el Scrum Master ni el propietario del producto.

No se trata de un grupo de trabajo formado por un arquitecto, diseñador o analista, programadores y testers. Es un equipo **multifuncional**, en el que todos los miembros trabajan de forma solidaria con responsabilidad compartida. Es posible que algunos miembros sean especialistas en áreas concretas, pero la responsabilidad es el incremento de cada sprint y recae sobre el equipo de desarrollo en conjunto.

Las principales responsabilidades, más allá de la autoorganización y uso de tecnologías ágiles, son las que se marcan la diferencia entre “grupo de trabajo” y “equipo”.

Un grupo de trabajo es un conjunto de personas que realizan un trabajo, con una asignación específica de tareas, responsabilidades y siguiendo un proceso o pautas de ejecución. Los operarios de una cadena, forman un grupo de trabajo: aunque tienen un jefe común, y trabajan en la misma organización, cada uno responde por su trabajo.

El equipo tiene espíritu de colaboración, y un propósito común: conseguir el mayor valor posible para la visión del cliente.

Un equipo scrum responde en su conjunto. Trabaja de forma cohesionada y autoorganizada. No hay un gestor para delimitar, asignar y coordinar las tareas. Son los propios miembros los que lo realizan.

En el equipo:

- Todos conocen y comprenden la visión del propietario del producto.
- Aportan y colaboran con el propietario del producto en el desarrollo de la pila del producto.
- Comparten de forma conjunta el objetivo de cada sprint y la responsabilidad del logro.
- Todos los miembros participan en las decisiones.
- Se respetan las opiniones y aportes de todos.
- Todos conocen el modelo de trabajo con scrum.

Scrum Master

Es el responsable del cumplimiento de las reglas de un marco de scrum técnico, asegurando que se entienden en la organización, y se trabaja conforme a ellas.

Proporciona la asesoría y formación necesaria al propietario del producto y al equipo.

Realiza su trabajo con un modelo de liderazgo servil: al servicio y en ayuda del equipo y del propietario del producto.

Proporciona:

- Asesoría y formación al equipo para trabajar de forma autoorganizada y con responsabilidad de equipo.
- Revisión y validación de la pila del producto.
- Moderación de las reuniones.
- Resolución de impedimentos que en el sprint pueden entorpecer la ejecución de las tareas.
- Gestión de las “dinámicas de grupo” en el equipo.
- Configuración, diseño y mejora continua de las prácticas de scrum en la organización. Respeto de la organización y los implicados, con las pautas de tiempos y formas de scrum.

Al crecer la fluidez de la organización y evolucionar hacia un marco de scrum más pragmático, puede eliminarse el rol de Scrum Master, cuando estas responsabilidades ya estén institucionalizadas en la organización.

Cultura y Valores

Scrum técnico define un marco que ayuda a organizar a las personas y el flujo de trabajo. Es la “carrocería” o el interfaz visible, pero el motor de la agilidad son los valores ágiles.

Las reglas de un equipo scrum pueden ser las de este marco técnico u otras. La agilidad no la proporciona el cumplimiento de prácticas, sino de valores.

- **Delegación de atribuciones** (empowerment) al equipo para que pueda autoorganizarse y tomar las decisiones sobre el desarrollo.
- **Respeto entre las personas**. Los miembros del equipo deben confiar entre ellos y respetar sus conocimientos y capacidades.
- **Responsabilidad y autodisciplina** (no disciplina impuesta).
- Trabajo centrado en el **valor para el cliente** y el desarrollo de lo comprometido.
- Información, **transparencia** y visibilidad del desarrollo del proyecto.

Medición y estimación ágil

¿Por qué medir?

La información es la materia prima para la toma de decisiones, y la que puede ser cuantificada proporciona criterios objetivos de gestión y seguimiento.

Desde el nivel concreto de la programación, hasta los más generales de la gestión global de la organización, tres son los fondos de escala o niveles de zoom con los que se puede medir el *trabajo*:

- Desarrollo y gestión de la solución técnica.
- Gestión de proyecto.
- Gestión de la organización.

En el primero se puede medir, por ejemplo, la proporción de polimorfismo del código de un programa, en el segundo, el porcentaje del plan del proyecto realizado, y en el tercero, también por ejemplo, el nivel de satisfacción laboral.

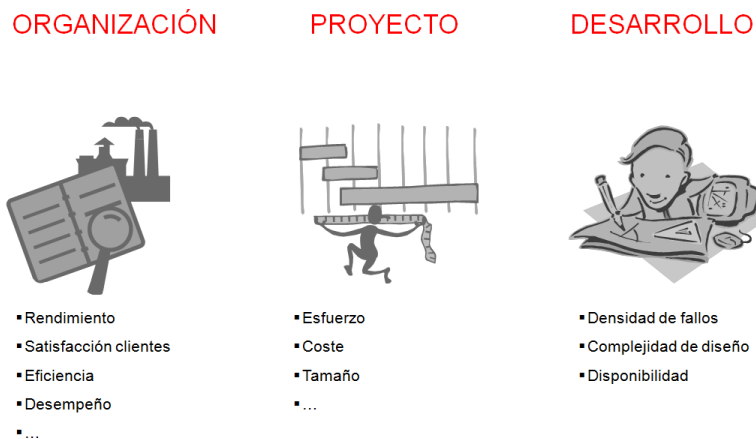


Ilustración 9: Ámbitos de medición

Este texto cubre la medición ágil en el ámbito proyecto, aunque las consideraciones generales de esta introducción son comunes a los tres.

Flexibilidad y sentido común

Medir es costoso

Antes de incorporar un procedimiento de medición, se debe cuestionar su conveniencia, y la forma en la que se aplicará.

Medir no es un fin en sí mismo

No se deben implantar procesos de medición simplemente por que sí.

Tomar una lista más o menos prestigiosa de métricas, e incorporarla a los procedimientos de la empresa puede seducir, al ofrecer un marco de trabajo monitorizado con indicadores y mediciones, pero no dice mucho a favor de las razones por las que se han adoptado.

Criterios para el diseño y aplicación de métricas

Cuantas menos, mejor

- Medir es costoso.
- Medir añade burocracia.
- El objetivo de un equipo scrum es ofrecer la mejor relación valor / simplicidad.

Aspectos que deben cuestionarse antes de monitorizar y medir un indicador:

- **¿Por qué?**
- **¿Qué valor proporciona esta medición?**
- **¿Qué valor se pierde si se omite?**

El objetivo de scrum es producir el mayor valor posible de forma continua, y la cuestión clave para la incorporación de indicadores en la gestión de proyectos es:

¿Cómo contribuye el uso del indicador en el valor que el proyecto proporciona al cliente?

¿El indicador es apropiado para el fin que se debe conseguir?

Medir no es, o no debería ser, un fin en sí mismo.

¿Cuál es el fin?

¿Cumplir agendas, mejorar la eficiencia del equipo, las previsiones...?

Sea crítico. Si después de analizarlo no le convence, si prefiere no incorporar un indicador, o cambiarlo: usted es el gestor.

Determinar qué medir es la parte más difícil. En el mejor de los casos, las decisiones erróneas sólo supondrán un coste de gestión evitable; pero muchas veces empeorarán lo que se intentaba mejorar.

Ejemplo: Medición de la eficiencia de los trabajos de programación

La organización XYZ ha adoptado métricas estándar de eficiencia de Ingeniería del Software:

LOC/Hour: Número total de líneas de código nuevas o modificadas en cada hora.

Además para aumentar la productividad, ha vinculado los resultados de esta métrica a la retribución por desempeño de los programadores, de forma que ha logrado producir más líneas de código sin incrementar la plantilla.

Para evitar que se trate de un incremento “hueco” de líneas de código, o aumente el número de errores por programar más deprisa, se ha dotado de mayor “rigor” al sistema de métrica, incorporando al poco tiempo otras métricas para complementar y mejorar el sistema de calidad:

Test Defects/KLOC, Compile Defects/KLOC y Total Defects/KLOC, para controlar que no crezca el número de errores deslizados en el código.

También se han incorporado indicadores “appraisal time” para medir tiempo y costes del diseño y la ejecución de las revisiones de código.

Y por temor a que el sistema de medición pueda resultar excesivamente costoso se incluyen indicadores de coste de calidad (COQ) que miden los tiempos de revisión y los contrastan con las mejoras en los tiempos eliminados por reducción de fallos.

¿Lo que vamos a medir es un indicador válido de lo que queremos conocer?

Hay tareas de programación relativamente mecánicas, orientadas más a la integración y configuración que en al desarrollo de nuevos sistemas.

Para aquellas puede resultar medianamente acertado considerar la eficiencia como volumen de trabajo realizado por unidad de tiempo.

Para las segundas sin embargo, es más apropiado pensar en la cantidad de valor integrado por unidad de desarrollo; expresadas éstas en horas, iteraciones o puntos de función.

¿Qué queremos conocer: la cantidad de líneas, o el valor entregado al cliente?

¿Está relacionado lo uno con lo otro?

¿Se puede medir objetivamente el valor entregado al cliente?

En nuestro trabajo son muchos los parámetros que se pueden medir con criterios objetivos y cuantificables: el tiempo de tarea, los tiempos delta, y los de las interrupciones, el nº de puntos de función, la inestabilidad de los requisitos, la proporción de acoplamiento, el nº de errores por línea de código...

¿No estaremos muchas veces midiendo esto, simplemente porque es cuantificable?

¿No estaremos midiendo el nº de líneas que desarrollan las personas cuando en realidad queremos saber el valor de su trabajo?

¿No nos estará pasando lo mismo cuando pretendemos medir: la facilidad de uso, la facilidad de mantenimiento, la flexibilidad, la transportabilidad, la complejidad, etc.?

Velocidad, trabajo y tiempo

Velocidad, trabajo y tiempo son las tres magnitudes que componen la fórmula de la velocidad, en gestión de proyectos ágil, definiéndola como la cantidad de trabajo realizada por unidad de tiempo.

$$\text{Velocidad} = \text{Trabajo} / \text{Tiempo}$$

Así por ejemplo, se puede decir que la velocidad de un equipo de 4 miembros es de 20 puntos por semana o de 80 puntos por sprint.

Tiempo

Para mantener un ritmo de avance continuo, el desarrollo ágil emplea dos tácticas posibles: incremento iterativo, o incremento continuo.



Ilustración 10: Agilidad con incremento iterativo o continuo

El avance a través de incrementos iterativos mantiene el ritmo apoyándose en pulsos de sprints. Por esta razón emplea normalmente el sprint como unidad de tiempo, y expresa la velocidad como trabajo o tareas realizadas en un sprint.

Nota: scrum técnico usa incremento iterativo, y por tanto define la velocidad como la cantidad de trabajo realizado en un sprint.

El avance a través de un incremento continuo mantiene un flujo de avance constante sin puntos muertos ni cuellos de botella. No hay sprints, y por tanto las unidades de tiempo son días, semanas o meses, de forma que la la velocidad se expresa en "puntos" (cantidad de trabajo) por semana, día, o mes

Tiempo real y tiempo ideal

¿Cuánto dura un partido de Baloncesto?



Tiempo ideal: 40 minutos

Tiempo real: > 2 horas

Fotografía co-by: SD Dirk

Una observación importante: la diferencia entre tiempo “real” y tiempo “ideal”.

Tiempo real, es el tiempo de trabajo. Equivale a la jornada laboral.

Para un equipo de cuatro personas con jornada laboral de ocho horas el tiempo real en una semana (cinco días laborables) es:

$$4 * 8 * 5 = 160 \text{ horas}$$

Tiempo ideal se refiere sin embargo al tiempo de trabajo en condiciones ideales, esto es, eliminando todo lo que no es estrictamente “trabajo”, suponiendo que no hay ninguna pausa por interrupción o atención de cuestiones ajenas a la tarea y que la persona se encuentra en buenas condiciones de concentración y disponibilidad.

El tiempo ideal se emplea normalmente en estimaciones, como unidad de trabajo o esfuerzo necesario. Ej: “Esa tarea tiene un tamaño de 3 horas ideales”.

Es un concepto similar al que PSP¹ denomina “Delta Time” como la parte del tiempo laboral que es realmente tiempo efectivo de trabajo.

Tiempo ideal no es una unidad de tiempo, sino de trabajo o esfuerzo necesario.

Trabajo

Medir el trabajo puede ser necesario por dos razones: para registrar el ya hecho, o para estimar anticipadamente, el que se debe realizar.

En ambos casos se necesita una unidad, y un criterio objetivo de cuantificación.

Trabajo ya realizado

Medir el trabajo ya realizado no entraña especial dificultad.

Se puede hacer con unidades relativas al producto (p. ej. líneas de código) o a los recursos empleados (coste, tiempo de trabajo...)

Para medirlo, basta contabilizar lo ya realizado con la unidad empleada: líneas de código, puntos de función, horas trabajadas, etc.

La gestión de proyectos ágil no mide el esfuerzo realizado para calcular el avance del trabajo.

La gestión ágil no determina el grado de avance del proyecto por el trabajo realizado, sino por el pendiente de realizar.

Es posible que otros procesos de la organización necesiten registrar el esfuerzo invertido, y por lo tanto sea necesario su registro, pero no debe emplearse para calcular el avance del proyecto.

Trabajo pendiente de realizar

Scrum mide el trabajo pendiente para:

¹ Personal Software Process

- Estimar esfuerzo y tiempo previsto para realizar un trabajo (tareas, historias de usuario o epics).
- Determinar el grado de avance del proyecto, y en especial en cada sprint.

Determinar con precisión, de forma cuantitativa y objetiva el trabajo que necesitará la construcción de un requisito, es un empeño cuestionable.

LA GESTIÓN ÁGIL MIDE EL TRABAJO PENDIENTE

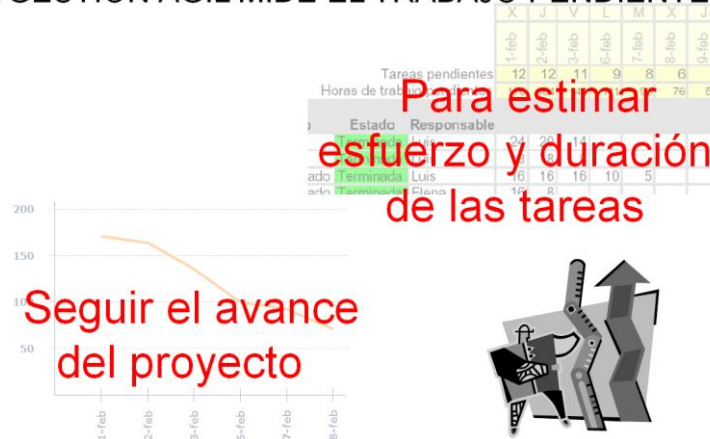


Ilustración 12: Medición del trabajo pendiente

El trabajo necesario para realizar un requisito o una historia de usuario no se puede prever de forma absoluta, porque las funcionalidades no son realidades de solución única, y en el caso de que se pudiera, la complejidad de la medición haría una métrica demasiado pesada para la gestión ágil.

Y si no resulta posible estimar con precisión la cantidad de trabajo que hay en un requisito, tampoco se puede saber cuánto tiempo necesitará, porque además de la incertidumbre del trabajo, se suman las inherentes al “tiempo”:

- No es realista hablar de la cantidad o de la calidad del trabajo que realiza una persona por unidad de tiempo, porque son muy grandes las diferencias de unas personas a otras.
- Una misma tarea, realizada por una misma persona requerirá diferentes tiempos en o situaciones distintas.

Sobre estas premisas:

- No es posible estimar con precisión, ni el trabajo de un requisito, ni el tiempo necesario para desarrollarlo.
- La complejidad de las técnicas de estimación crece exponencialmente en la medida que:
 - Intentan incrementar la fiabilidad y precisión de los resultados.
 - Aumenta el tamaño del trabajo estimado.

La estrategia empleada por la gestión ágil es:

- Trabajar con estimaciones aproximadas.
- Estimar con la técnica “juicio de expertos.”
- Descomponer las tareas en subtareas más pequeñas, si las estimaciones superan rangos de medio, o un día de tiempo real.

Unidades de trabajo

Un trabajo puede dimensionarse midiendo el producto que se construye, como los tradicionales puntos de función de COCOMO; o el tiempo que cuesta realizarlo.

$$\text{Velocidad} = \text{Trabajo} / \text{Tiempo}$$

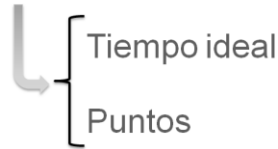


Ilustración 13: Velocidad

En gestión ágil se suelen emplear “puntos” como unidad de trabajo, empleando denominaciones como “puntos de historia” o simplemente “puntos” “puntos”.

La unidad “Story Point” de eXtreme Programming se define como la cantidad de trabajo que se realiza en un “día ideal”.

Cada organización, según sus circunstancias y su criterio institucionaliza su métrica de trabajo definiendo el nombre y las unidades.

Puede definir su “punto”

- Como tamaño relativo de tareas conocidas que normalmente emplea.
Ej: El equipo de un sistema de venta por internet, podría determinar que un “punto” representara el tamaño que tiene un “listado de las facturas de un usuario”.
- En base al tiempo ideal necesario para realizar el trabajo.
Ej: Un equipo puede determinar que un “punto” es el trabajo realizado en 4 horas ideales.

Es importante que la métrica empleada, su significado y la forma de aplicación sea consistente en todas las mediciones de la organización, y conocida por todas las personas:

Que se trate de un procedimiento de trabajo institucionalizado.

Velocidad

Velocidad es la magnitud determinada por la cantidad de trabajo realizada en un periodo de tiempo.

Velocidad en scrum técnico es la cantidad de trabajo realizada por el equipo en un sprint. Así por ejemplo, una velocidad de 150 puntos indica que el equipo realiza 150 puntos de trabajo en cada sprint.

Al trabajar en implantaciones de scrum pragmático, que pueden realizar sprints de diferentes duraciones, o no siempre con el mismo número de miembros en el equipo, la velocidad se expresa indicando la unidad de tiempo y en su caso también si se refiere a la total del equipo, o a la media por persona. Así por ejemplo: “La velocidad media del equipo es de 100 puntos por semana.” “La velocidad media de una persona del equipo es de 5 puntos por día.”

Medición: usos y herramientas

Gráfico de producto.

El gráfico de producto o gráfico “*burn up*” es una herramienta de planificación del propietario del producto, que muestra visualmente la evolución previsible del producto.

Proyecta en el tiempo su construcción, en base a la velocidad del equipo.

La proyección se realiza sobre un diagrama cartesiano que representa en el eje de ordenadas el esfuerzo estimado para construir las diferentes historias de la pila del producto, y en el de las abscisas el tiempo, medido en sprints o en tiempo real.

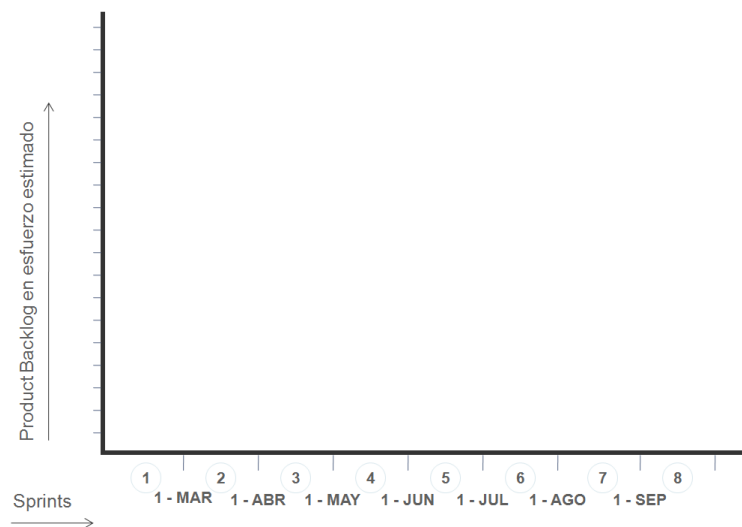


Ilustración 14: Gráfico de producto

Ejemplo

Convenciones empleadas por el equipo:

- Unidad para estimar el trabajo: puntos de scrum.
- Está previsto trabajar con sprints de duración fija: mensual (20 días laborables)
- El equipo está formado por 4 personas, y desarrolla una velocidad media de 400 puntos por sprint.

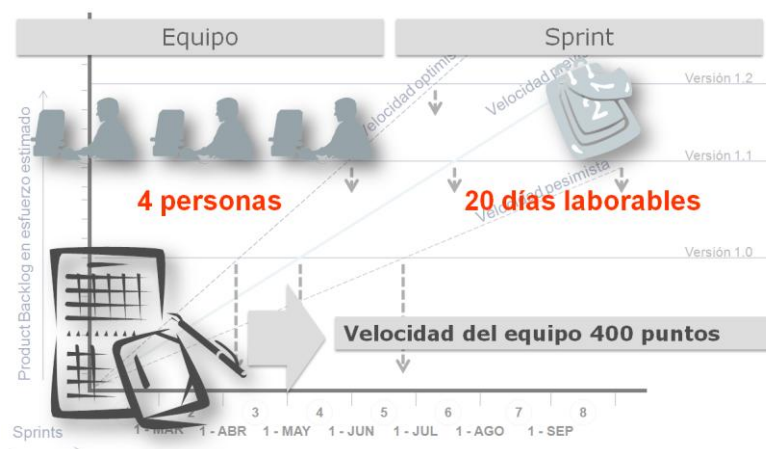


Ilustración 15: Gráfico de producto como plan de producto

Se traza en el gráfico la línea que representa el ritmo de avance previsto, según la velocidad media del equipo (en este ejemplo 400 puntos por sprint).

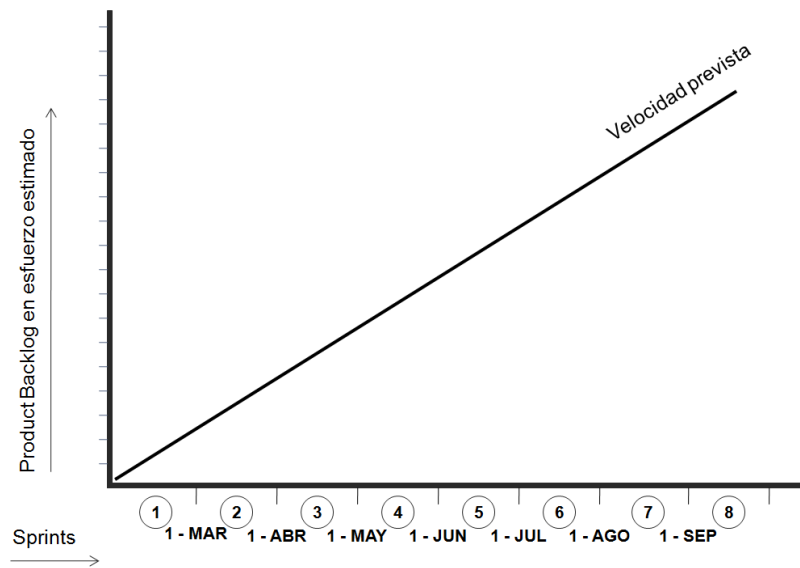


Ilustración 16: Gráfico de producto: velocidad prevista

Es recomendable trazar también los ritmos de avance con una previsión pesimista y otra optimista. Se dibujan basándose en la velocidad obtenida en los sprints anteriores que han ido peor y mejor de lo previsto, o en su defecto estableciendo un margen según el criterio del equipo (ej. $\pm 20\%$).

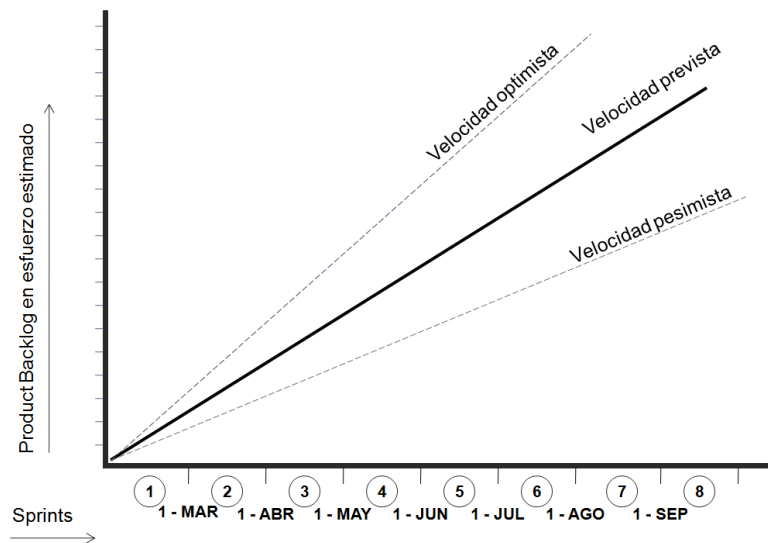


Ilustración 17: Gráfico de producto: velocidad optimista y pesimista

A continuación se toma la pila del producto. La figura siguiente representa la empleada en este ejemplo:

Id	Historias	Trabajo	Criterio de validación
1	Historia A 1.0	150	Lorem ipsum dolor sit amet
2	Historia B 1.0	250	consectetur adipiscing elit
3	Historia C 1.0	250	Aliquam vehicula accumsan tortor
4	Historia D 1.0	300	Pellentesque turpis
5	Historia A 1.1	250	Phasellus purus orci
6	Historia D 1.1	350	penatibus et magnis dis parturient
7	Historia E 1.0	150	Quisque volutpat ante sit amet velit
8	Historia B 1.1	500	Cras iaculis pede eu tellus
9	Historia C 1.1	150	Vestibulum vel diam sed pede blandit
10	Historia E 1.1	200	Suspendisse aliquam felis et turpis
11	Historia F 1.0	TBD	Nullam imperdiet lorem vitae justo
12	Historia A 1.2	TBD	Suspendisse potenti. In nec nunc
13	Historia B 1.2	TBD	Nam eros tellus, facilisis sed, pretium
14	Historia F 1.1	TBD	Morbi arcu tellus, condimentum

Ilustración 18: Ejemplo de pila del producto

En este caso, el propietario del producto tiene previsto lanzar la versión 1.0 cuando disponga de las cuatro primeras historias, que tienen un esfuerzo estimado en 950 puntos (150+250+250+300).

Además tiene también esbozadas las previsiones para versiones posteriores: 1.1 y 1.2 tal y como muestra la figura siguiente:

Id	Historias	Trabajo	Criterio de validación	
1	Historia A 1.0	150	Lorem ipsum dolor sit amet	Estimación: 950 PUNTOS
2	Historia B 1.0	250	consectetur adipiscing elit	
3	Historia C 1.0	250	Aliquam vehicula accumsan tortor	
4	Historia D 1.0	300	Pellentesque turpis	
5	Historia A 1.1	250	Phasellus purus orci	Versión 1.0 →
6	Historia D 1.1	350	penatibus et magnis dis parturient	1.700 PUNTOS
7	Historia E 1.0	150	Quisque volutpat ante sit amet velit	Versión 1.1 →
8	Historia B 1.1	500	Cras iaculis pede eu tellus	2.550 PUNTOS
9	Historia C 1.1	150	Vestibulum vel diam sed pede	
10	Historia E 1.1	200	Suspendisse aliquam felis et turpis	Versión 1.2 →
11	Historia F 1.0	TBD	Nullam imperdiet lorem vitae justo	
12	Historia A 1.2	TBD	Suspendisse potenti. In nec nunc	
13	Historia B 1.2	TBD	Nam eros tellus, facilisis sed, pretium	
14	Historia F 1.1	TBD	Morbi arcu tellus, condimentum	

Ilustración 19: Versiones del producto previstas

Para trazar la previsión, se sitúa cada versión en el eje vertical en la posición correspondiente al esfuerzo calculado para construir todas las historias que incluye.

Siguiendo con el ejemplo, la posición de la versión 1.0 se situaría sobre el valor 950 del eje de ordenadas:

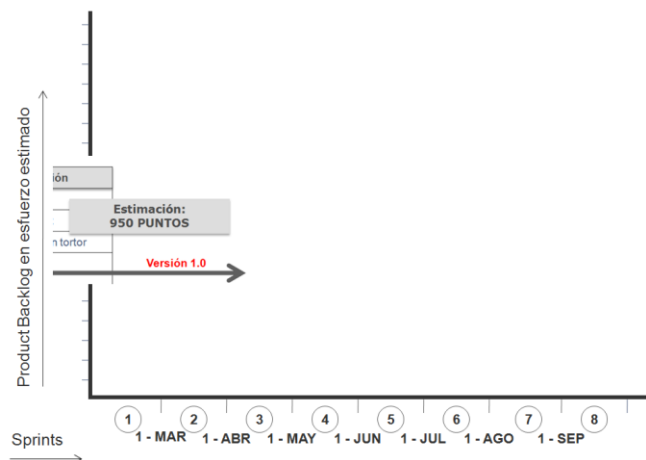


Ilustración 20: Representación de la versión 1 sobre el gráfico de producto

Los puntos de corte que marca esta posición con las líneas de velocidad del equipo (pesimista, realista y optimista) proyectan en el eje horizontal la fecha o sprint en el que se espera completar la versión.

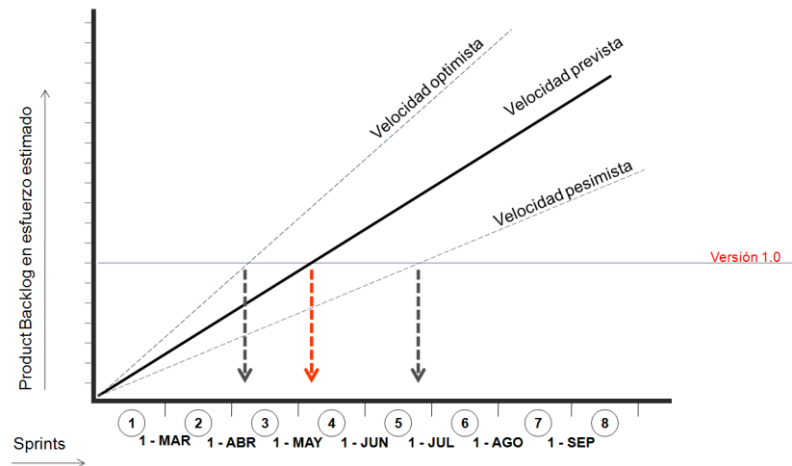


Ilustración 21: Previsión de fechas sobre el gráfico de producto

De igual forma se pueden proyectar las estimaciones tempranas de las futuras versiones previstas.

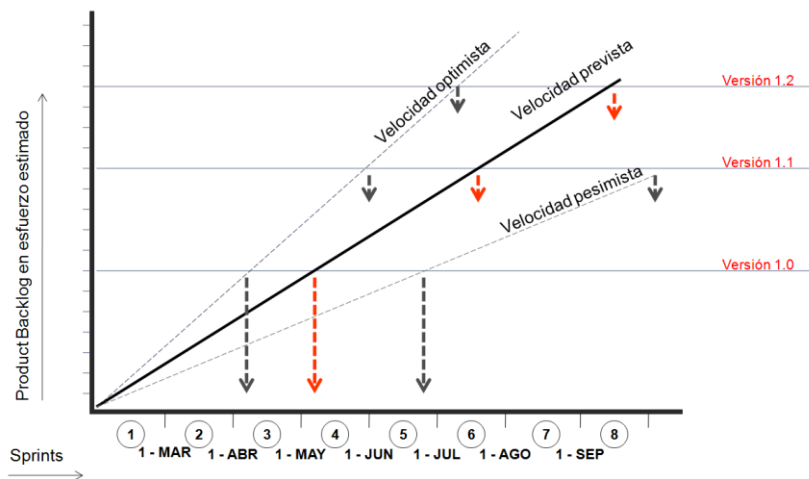


Ilustración 22: previsión de lanzamiento de versiones sobre gráfico de producto

Esta herramienta proyecta la previsión de la pila del producto, que es un documento vivo cuya evolución prevé la del producto.

Como herramienta ágil no debe considerarse como la representación de un plan estable, sino como la previsión de la pila del producto.

Gráfico de avance: monitorización del sprint

También se suele llamar a este gráfico con su nombre inglés: burn-down”.

Lo actualiza el equipo en el scrum diario, para comprobar el ritmo de avance, y detectar desde el primer momento si es el previsto, o por el contrario se puede ver comprometida o adelantada la entrega prevista al final de sprint.

La estrategia ágil para el seguimiento del proyecto se basa en:

- Medir el trabajo que falta, no el realizado.
- Seguimiento cercano del avance (diario de ser posible).

Y este gráfico trabaja con ambos principios:

- Registra en el eje Y el trabajo pendiente.
- Se actualiza a diario.

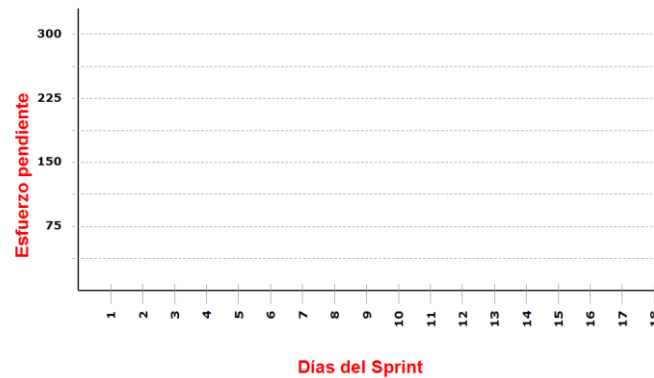


Ilustración 23: Gráfico de avance

El equipo dispone en la pila del sprint, de la lista de tareas que va a realizar, y en cada una figura el esfuerzo pendiente.

Esto es: el primer día, en la pila de tareas figura para cada tarea el esfuerzo que se ha estimado, puesto que aún no se ha trabajado en ninguna de ellas.

Día a día, cada miembro del equipo actualiza en la pila del sprint el tiempo que le queda a las tareas que va desarrollando, hasta que se terminan y van quedando 0 como tiempo pendiente.

La figura siguiente muestra un ejemplo de pila en el sexto día del sprint: las tareas terminadas ya no tienen esfuerzo pendiente, y del esfuerzo total previsto para el sprint: 276 puntos (A), en el momento actual quedan 110 (B).

SPRINT		INICIO	DURACIÓN														
1		1-mar-07	12	J	V	L	M	X	J	V	L	M	J	V	L		
				1-mar	2-mar	3-mar	4-mar	5-mar	6-mar	7-mar	8-mar	9-mar	10-mar	11-mar	12-mar	13-mar	14-mar
				23	23	19	16	16	13	9	9	9	9	9	9	9	9
				276	246	216	190	178	158	110	110	110	110	110	110	110	110
SPRINT BACKLOG				ESFUERZO													
Tarea	Estado	Responsable		J	V	L	M	X	J	V	L	M	J	V	L		
Descripción de la tarea 1	Terminada	Luis		16	16	16	16	16	16								
Descripción de la tarea 2	Terminada	Luis		12	8												
Descripción de la tarea 3	Terminada	Luis		4	4	4	4	4									
Descripción de la tarea 4	Terminada	Elena		8	4												
Descripción de la tarea 5	Terminada	Elena		16	16	4											
Descripción de la tarea 6	Terminada	Elena		6	6	2											
Descripción de la tarea 7	Terminada	Antonio		16	4												
Descripción de la tarea 8	Terminada	Antonio		16	16	20	12	4									
Descripción de la tarea 9	Terminada	Antonio		12	2												
Descripción de la tarea 10	En curso	Luis		12	12	12	12	12	12	12	12	12	12	12	12	12	12
Descripción de la tarea 11	Pendiente	Luis		8	8	8	8	8	8	8	8	8	8	8	8	8	8
Descripción de la tarea 12	En curso	Luis		14	14	14	14	14	14								
Descripción de la tarea 13	En curso	Antonio		8	8	8	8	8	6								
Descripción de la tarea 14	Pendiente	Antonio		16	16	16	16	16	16	16	16	16	16	16	16	16	16
Descripción de la tarea 15	Pendiente	Antonio		16	16	16	16	16	16	16	16	16	16	16	16	16	16
Descripción de la tarea 16	Terminada	Elena		8	8	8											
Descripción de la tarea 17	Terminada	Elena		12	12	12	8	4									
Descripción de la tarea 18	En curso	Elena		16	16	16	16	10	10	10	10	10	10	10	10	10	10
Descripción de la tarea 19	Terminada	Elena		12	12	12	12	12									
Descripción de la tarea 20	Pendiente	Elena		16	16	16	16	16	16	16	16	16	16	16	16	16	16
Descripción de la tarea 21	Pendiente	Elena		12	12	12	12	12	12	12	12	12	12	12	12	12	12
Descripción de la tarea 22	Pendiente	Antonio		8	8	8	8	8	8	8	8	8	8	8	8	8	8
Descripción de la tarea 23	Pendiente	Antonio		12	12	12	12	12	12	12	12	12	12	12	12	12	12

Ilustración 24: Pila del sprint

Con esta información de la pila del sprint se actualiza el gráfico poniendo cada día el esfuerzo pendiente total de todas las tareas que aún no se han terminado.

Se cumplimenta a diario

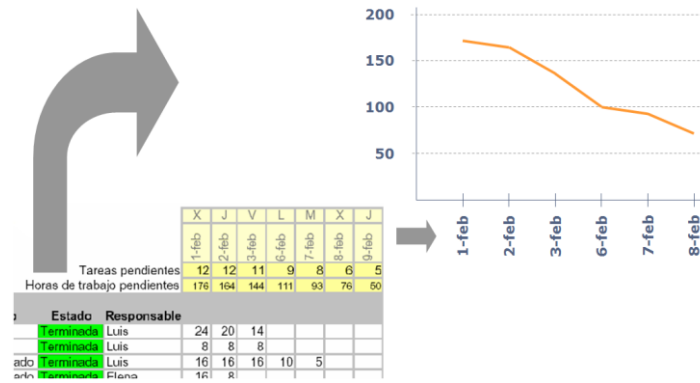


Ilustración 25: De la pila del sprint al gráfico de avance

El avance ideal de un sprint estaría representado por la diagonal que reduce el esfuerzo pendiente de forma continua y gradual hasta completarlo el día que termina el sprint.

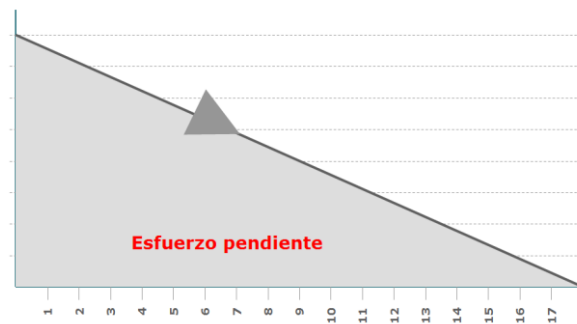


Ilustración 26: Gráfica de avance previsto

Las gráficas de diagonal perfecta no son lo habitual, y la siguiente imagen es un ejemplo de un patrón de avance más normal.

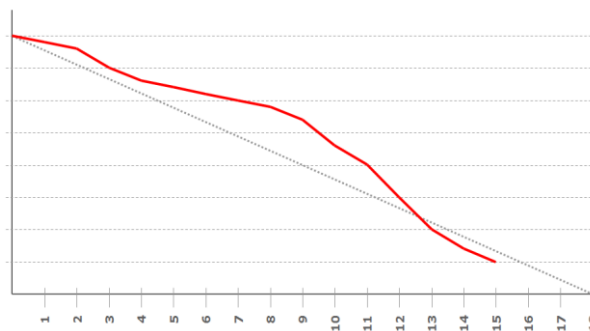


Ilustración 27: Gráfica de avance real

El siguiente sería el aspecto de la gráfica en un “sprint subestimado”

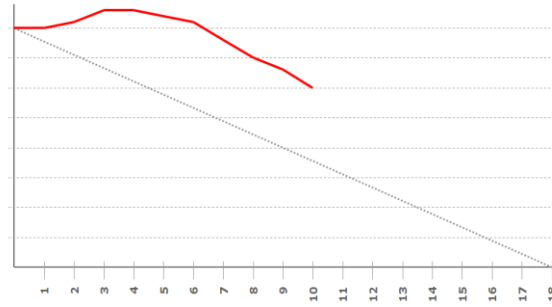


Ilustración 28: Gráfica de avance de un sprint subestimado

La estimación que realizó el equipo en la reunión de inicio del sprint es inferior al esfuerzo real que están requiriendo las tareas.

Y el siguiente sería el patrón de gráfica de un “sprint sobreestimado”.

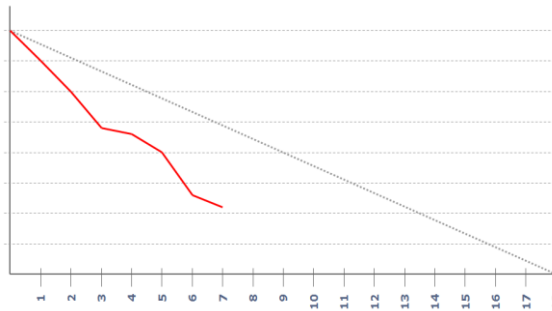


Ilustración 29: Gráfica de avance de un sprint sobreestimado

Estimación de póquer

Es una práctica ágil, para conducir las reuniones en las que se estima el esfuerzo y la duración de tareas.

James Grenning ideó este juego de planificación para evitar discusiones dilatadas que no terminan de dar conclusiones concretas.

El modelo inicial de Grenning consta de 8 cartas, con los números representados en siguiente figura,

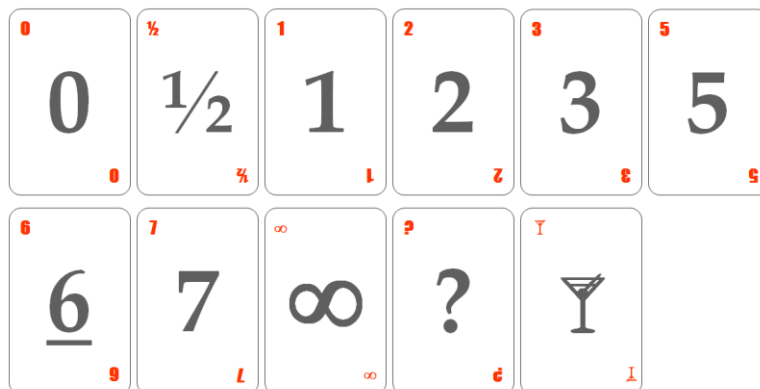


Ilustración 30: Cartas para planificación de póquer

El funcionamiento es muy simple: cada participante dispone de un juego de cartas, y en la estimación de cada tarea, todos vuelven boca arriba la combinación que suma el esfuerzo estimado.

Cuando se considera que éste es mayor de x horas ideales (el tamaño máximo considerado por el equipo para una historia), se levanta la carta “∞”.

Las tareas que exceden el tamaño máximo deben descomponerse en subtareas de menor tamaño.

Cada equipo u organización puede utilizar un juego de cartas con las numeraciones adecuadas a la unidad de esfuerzo con la que trabajan, y el tamaño máximo de tarea o historia que se va a estimar.

Variante: sucesión de Fibonacci

Basado en el hecho de que al aumentar el tamaño de las tareas, aumenta también la incertidumbre y el margen de error, se ha desarrollado esta variante que consiste en emplear sólo números de la sucesión de Fibonacci, de forma que:

- El juego de cartas está compuesto por números en sucesión de Fibonacci.
- La estimación no se realiza levantando varias cartas para componer la cifra exacta, sino poniendo boca arriba la carta con la cifra más aproximada a la estimación.

Así, si por ejemplo una persona cree que el tamaño adecuado de una tarea es 6, se ve obligado a reconsiderar y, o bien aceptar que el tamaño puede ser 5, o bien aceptar una estimación más conservadora y levantar el 8.

Para estimar tareas puede ser válido un juego de cartas como éste:



Ilustración 31: Cartas para estimación de póquer (Fibonacci)

Es frecuente emplear una carta con un símbolo de duda o interrogación para indicar que, por las razones que sean, no se puede precisar una estimación.

También es posible incluir otra carta con alguna imagen alusiva, para indicar que se necesita un descanso.

Operativa

- Cada participante de la reunión tiene un juego de cartas.
- Para cada tarea (historia de usuario o funcionalidad, según sea el nivel de requisitos que se va a estimar) el cliente, moderador o propietario del producto expone la descripción empleando un tiempo máximo.
- Hay establecido otro tiempo para que el cliente o propietario del producto atienda a las posibles preguntas del equipo.
- Cada participante selecciona la carta, o cartas que representan su estimación, y las separa del resto, boca abajo.
- Cuando todos han hecho su selección, se muestran boca arriba.
- Si la estimación resulta “infinito”, por sobrepasar el límite máximo establecido, la tarea debe dividirse en sub-tareas de menor tamaño.
- Si las estimaciones resultan muy dispares, quien asume la responsabilidad de gestionar la reunión, con su criterio de gestión, y basándose en las características del proyecto, equipo, reunión, nº de elementos pendientes de evaluar, puede optar por:
 - Preguntar a las personas de las estimaciones extremas: ¿Por qué crees que es necesario tanto tiempo?, y ¿por qué crees que es necesario tan poco tiempo? Tras escuchar las razones, repetir la estimación.
 - Dejar a un lado la estimación de esa tarea y retomar al final o en otro momento aquellas que hayan quedado pendientes.

- Pedir al cliente o propietario del producto que descomponga la funcionalidad y valorar cada una de las funcionalidades resultantes.
- Tomar la estimación menor, mayor, o la media.

Este protocolo de moderación, evita en la reunión los atascos de análisis circulares en ping-pong entre diversas opciones de implementación, hace participar a todos los asistentes, reduce el cuarto de hora o la media hora de tiempo de estimación de una funcionalidad, a escasos minutos, consigue alcanzar consensos sin discusiones, y además resulta divertido y dinamiza la reunión.

Bibliografía

- Bauer, F., Bolliet, L., & Helms, H. (1969). Software Engineering. Report on a conference sponsored by the NATO SCIENCE COMITEE. *Software Engineering* (pág. 136). Garmisch: Peter Naur & Brian Randell.
- Beck, K. (2000). *Extreme Programming Explained*. Addison-Wesley.
- Hino, S. (2006). *Inside the Mind of Toyota: Management Principles for Enduring Growth*. Productivity Press.
- Kniberg, H., & Skarin, M. (2009). *Kanban and Scrum, making the most of both*. crisp.
- Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-Creating Company*. University Press.
- Nonaka, I., & Takeuchi, H. (1986). The New New Product Development Game. *Harvard Business Review* .
- Nonaka, I., & Takeuchi, I. (2004). *Hitotsubashi on Knowledge Management*. Singapore: John Wiley & Sons.
- Ohno, T. (1988). *The Toyota Production System: Beyond Large-scale Production*. Productivity Press.
- Orr., K. (2003). CMM versus Agile Development: Religious wars and software development. *Cutter Consortium, Executive Reports* 3(7) .
- Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit for Software Development Managers*. Addison Wesley.
- Schwaber, K. (1995). *SCRUM Development Process*. Burlington: OOPSLA 95.
- Smith, A. (1776). *An Inquiry into the Nature and Causes of the Wealth of Nations*. Londres: W. Strahan & T. Cadell.
- Taylor, F. W. (1911). *The Principles of Scientific Management*. New York: Harper & Brothers.
- Turner, R., & Jain, A. (2002). Agile Meets CMMI: Culture Clash or Common Cause? *XP/Agile Universe 2002* , 153-165.
-

Tabla de ilustraciones

Ilustración 1: Marco scrum técnico	20
Ilustración 2: Incremento iterativo / continuo	22
Ilustración 3: Diagrama del ciclo iterativo scrum	23
Ilustración 4: Requisitos completos / evolutivos	23
Ilustración 5: Ejemplo de pila de producto.....	25
Ilustración 6: Ejemplo de pila de sprint con hoja de cálculo	26
Ilustración 7: Ejemplo de pizarra de trabajo	30
Ilustración 8: Roles estándar de scrum	33
Ilustración 9: Ámbitos de medición	38
Ilustración 10: Agilidad con incremento iterativo o continuo	41
Ilustración 11: Tiempo ideal y tiempo real	42
Ilustración 12: Medición del trabajo pendiente	43
Ilustración 13: Velocidad.....	44
Ilustración 14: Gráfico de producto.....	45
Ilustración 15: Gráfico de producto como plan de producto	45
Ilustración 16: Gráfico de producto: velocidad prevista.....	46
Ilustración 17: Gráfico de producto: velocidad optimista y pesimista.....	46
Ilustración 18: Ejemplo de pila del producto.....	47
Ilustración 19: Versiones del producto previstas	47
Ilustración 20: Representación de la versión 1 sobre el gráfico de producto.....	47
Ilustración 21: Previsión de fechas sobre el gráfico de producto	48
Ilustración 22: previsión de lanzamiento de versiones sobre gráfico de producto.....	48
Ilustración 23: Gráfico de avance	49
Ilustración 24: Pila del sprint.....	49
Ilustración 25: De la pila del sprint al gráfico de avance	50
Ilustración 26: Gráfica de avance previsto	50
Ilustración 27: Gráfica de avance real	50
Ilustración 28: Gráfica de avance de un sprint subestimado.....	51
Ilustración 29: Gráfica de avance de un sprint sobreestimado	51
Ilustración 30: Cartas para planificación de póquer	51
Ilustración 31: Cartas para estimación de póquer (Fibonacci).....	52

Índice

- Accionable, 25
 - Agilidad, 12
 - manifiesto, 12, 13
 - principios, 14
 - Autoorganización, 18
 - Burn-down, 27, 48
 - Cerdo, 32
 - Colaboración, 18
 - Desarrollo incremental, 17
 - Epic*, 43
 - Equipo, 34
 - Estimación de póquer, 51
 - Fibonacci, 52
 - Exploración, 24
 - Gallina, 32
 - Gráfico de avance, 27, 30, 48
 - Gráfico de producto, 45
 - Hecho, 27
 - Historia de usuario, 43
 - Incremento, 22, 27
 - Incremento continuo, 22, 41
 - Incremento iterativo, 22, 41
 - James Grenning, 51
 - Jeff Sutherland, 15
 - Métricas, 39
 - Estrategia de la gestión ágil, 43
 - Objetivo del sprint, 29
 - Pila del product
 - preparación, 25
 - Pila del producto, 22, 24
 - Pila del sprint, 26, 27
 - Plan de producto, 48
 - Planificación del sprint, 27, 28
 - Producto
 - Plan, 48
 - Propietario del producto, 33
 - Puntos de historia, 44
 - Requisitos del sistema, 23
 - Requisitos del software, 23
 - Requisitos del sprint, 24
 - Retrospectiva, 27, 32
 - Reunión de pie, 17
 - Revisión del sprint, 31, 32
 - Roles, 32
 - Scrum, 15
 - elementos, 22
 - eventos, 22
 - origen, 14
 - pragmático, 16, 35
 - roles, 22
 - técnico, 16, 22
 - Scrum diario, 17, 27, 31
 - Scrum Master, 34
 - Sprint, 17, 22
 - sobreestimado, 51
 - subestimado, 50
 - Stand-up meeting, 17
 - Story Point, 44
 - Tiempo, 41
 - Tiempo ideal, 42
 - Tiempo real, 42
 - Trabajo, 42
 - Valores, 35
 - Velocidad, 41, 44
-