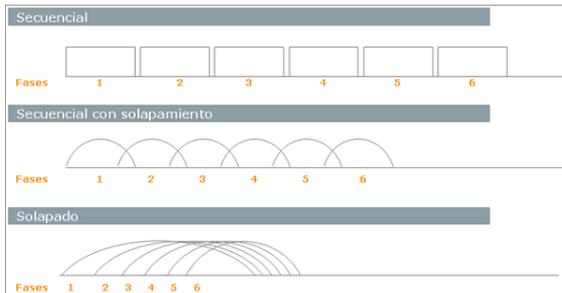
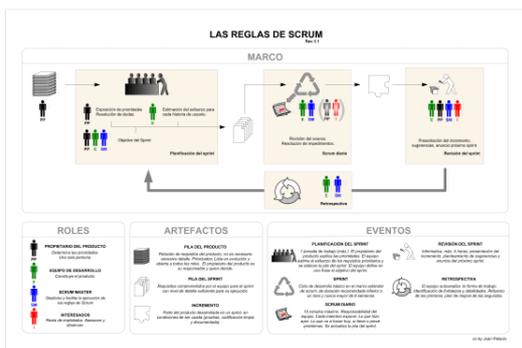


Scrum

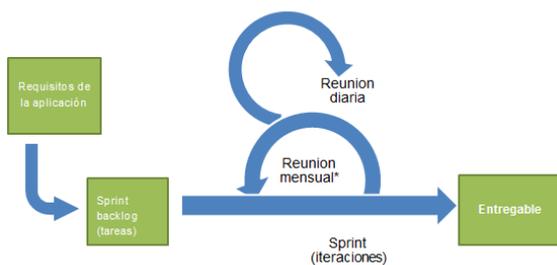
umh2818-TADS



Ciclos de desarrollo.



Ficha sinóptica



Marco de Trabajo SCRUM

Scrum es el nombre con el que se denomina a los marcos de desarrollo ágiles caracterizados 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 auto organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o en cascada.

1 Historia

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 melé (scrum en inglés) 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 cualquier tipo de proyecto 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).

2 Características de Scrum

SCRUM es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales

en Scrum son el *Scrum Master*, que procura facilitar la aplicación de scrum y gestionar cambios, el *Product Owner*, que representa a los *stakeholders* (interesados externos o internos), y el *Team* (equipo) que ejecuta el desarrollo y demás elementos relacionados con el. Durante cada *sprint*, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo y debe ser lo más corta posible), el equipo crea un incremento de software *potencialmente entregable* (utilizable). El conjunto de características que forma parte de cada sprint viene del *Product Backlog*, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar (PBI, Product Backlog Item). Los elementos del *Product Backlog* que forman parte del sprint se determinan durante la reunión de *Sprint Planning*. Durante esta reunión, el *Product Owner* identifica los elementos del *Product Backlog* que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo conversa con el *Product Owner* buscando la claridad y magnitud adecuadas (Cumpliendo el INVEST) para luego determinar la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint.^[1] Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Scrum permite la creación de equipos auto organizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado *requirements churn*), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Las características más marcadas que se logran notar en Scrum serían: gestión regular de las expectativas del cliente, resultados anticipados, flexibilidad y adaptación, retorno de inversión, mitigación de riesgos, productividad y calidad, alineamiento entre cliente y equipo, por último equipo motivado. Cada uno de estos puntos mencionados hacen que el Scrum sea utilizado de manera regular en un conjunto de buenas prácticas para el trabajo en equipo y de esa manera obtener resultados posibles.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas “post-it” y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar. Así, si se utiliza una pizarra con notas autoadhesivas cualquier miembro del equipo podrá ver tres columnas: trabajo pendiente (“backlog”), tareas en proceso (“in progress”) y hecho (“done”). De un solo vistazo, una

persona puede ver en qué están trabajando los demás en un momento determinado.^[2]

3 Roles en Scrum

3.1 Roles Principales

Product Owner El *Product Owner* representa la voz del cliente. Se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El *Product Owner* escribe *historias de usuario*, las prioriza, y las coloca en el *Product Backlog*.

ScrumMaster (o Facilitador) El *Scrum* es facilitado por un *ScrumMaster*, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El *ScrumMaster* no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El *ScrumMaster* se asegura de que el proceso Scrum se utilice como es debido. El *ScrumMaster* es el que hace que las reglas se cumplan.

Equipo de desarrollo El equipo tiene la responsabilidad de entregar el producto. Es recomendable un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

3.2 Roles Auxiliares

Los roles auxiliares en los “equipos Scrums” son aquellos que no tienen un rol formal y no se involucran frecuentemente en el “proceso Scrum”, sin embargo deben ser tomados en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados (“stakeholders”). Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

Stakeholders (Clientes, Proveedores, Vendedores, etc)

Son las personas que hacen posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su desarrollo. Sólo participan directamente durante las revisiones del “sprint”.

Administradores (Managers) Son los responsables de establecer el entorno para el desarrollo del proyecto.

4 Reuniones en Scrum

4.1 Daily Scrum o Stand-up meeting

Cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama *daily standup* o *Stand-up meeting*. El scrum tiene unas guías específicas:

- La reunión comienza puntualmente a su hora.
- Todos son bienvenidos, pero sólo los involucrados en el proyecto pueden hablar.
- La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.
- La reunión debe celebrarse en la misma ubicación y a la misma hora todos los días.

Durante la reunión, cada miembro del equipo contesta a tres preguntas:^[3]

- ¿Qué has hecho desde ayer?
- ¿Qué es lo que harás para mañana?
- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? (Es el papel del ScrumMaster recordar estos impedimentos).

El objetivo último de las reuniones diarias es que cada miembro del equipo sepa si se están cumpliendo los plazos marcados para el “sprint”.

4.2 Scrum de Scrum

Estas reuniones, por lo general, se realizan cuando en la organización existan varios equipos Scrum, y les permiten discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración. Se hace normalmente cada día después del “Daily Scrum” o, como máximo, cada dos días. Asiste una persona asignada por cada equipo Scrum.

La agenda será la misma que la del Daily Scrum, añadiendo, además, las siguientes cuatro preguntas:

- ¿Qué ha hecho tu equipo desde nuestra última reunión?
- ¿Qué hará tu equipo antes que nos volvamos a reunir?
- ¿Hay algo que demora o estorba a tu equipo?
- ¿Estás a punto de poner algo en el camino del otro equipo?

4.3 Reunión de Planificación del Sprint (Sprint Planning Meeting)

Al inicio de cada ciclo de Sprint (cada 15 o 30 días), se lleva a cabo una *reunión de planificación del Sprint*. Se pretende:

- Seleccionar qué trabajo se hará.
- Preparar, con el equipo completo, el Sprint Backlog que detalla el tiempo que llevará hacer el trabajo.
- Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint.
- Realizarse esta planificación en ocho horas como tiempo límite.

Al final del ciclo Sprint se celebran dos reuniones más: la *reunión de revisión del Sprint* y la *retrospectiva del Sprint*.

4.4 Reunión de Revisión del Sprint (Sprint Review Meeting)

- Revisar el trabajo que fue completado y no completado
- Presentar el trabajo completado a los interesados (alias “demo”)
- El trabajo incompleto no puede ser demostrado
- Cuatro horas como límite

4.5 Retrospectiva del Sprint (Sprint Retrospective)

Después de cada sprint, se lleva a cabo una retrospectiva del propio sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

5 Sprint

El Sprint es el período en el cual se lleva a cabo el trabajo en sí. Es recomendado que la duración de los sprints sea constante y definida por el equipo con base en su propia experiencia. Se puede comenzar con una duración de sprint en particular (2 o 3 semanas) e ir ajustándolo con base en el ritmo del equipo, aunque sin relajarlo demasiado. Al final de cada sprint, el equipo deberá presentar los avances logrados, y el resultado obtenido es un producto que, potencialmente, se puede entregar al cliente.

Así mismo, se recomienda no agregar objetivos al sprint o *sprint backlog* a menos que su falta amenace al éxito

del proyecto. La constancia permite la concentración y mejora la productividad del equipo de trabajo.

6 Beneficios de Scrum

- **Flexibilidad a cambios.** Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado. El marco de trabajo está diseñado para adecuarse a las nuevas exigencias que implican proyectos complejos.
- **Reducción del Time to Market.** El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado.
- **Mayor calidad del software.** El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad.
- **Mayor productividad.** Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- **Maximiza el retorno de la inversión (ROI).** Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.
- **Predicciones de tiempos.** A través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil cuando se podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.
- **Reducción de riesgos** El hecho de desarrollar, en primer lugar, las funcionalidades de mayor valor y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada.^[4]

7 Documentos

7.1 Product backlog

El *product backlog* se trata como un documento de alto nivel para todo el proyecto. Es el conjunto de todos los requisitos de proyecto, el cual contiene descripciones genéricas de funcionalidades deseables, priorizadas según

su retorno sobre la inversión (ROI) . Representa el *qué* va a ser construido en su totalidad. Es abierto y solo puede ser modificado por el *product owner*. Contiene estimaciones realizadas a grandes rasgos, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido. Esta estimación ayuda al *product owner* a ajustar la línea temporal (KEV) y, de manera limitada, la prioridad de las diferentes tareas. Por ejemplo, si dos características tienen el mismo valor de negocio la que requiera menor tiempo de desarrollo tendrá probablemente más prioridad, debido a que su ROI será más alto.

7.2 Sprint backlog

El *sprint backlog* es el subconjunto de requisitos que serán desarrollados durante el siguiente sprint. Al definir el sprint backlog, se describe el *cómo* el equipo va a implementar los requisitos durante el sprint. Por lo general los requisitos se subdividen en tareas, a las cuales se asignan ciertas *horas de trabajo* pero ninguna tarea con una duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser dividida en otras menores. Las tareas en el *sprint backlog* nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca adecuado.

7.3 Burn down chart

La *burn down chart* es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, podremos ver el progreso del proyecto. Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el Backlog). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.

8 Notas

[1] *Agile Project Management with Scrum*, Ken Schwaber, Microsoft Press, January 2004, 163pp, ISBN 0-7356-1993-X

[2] Leader Summaries (ed.). «Resumen del libro Scrum, de Jeff Sutherland». Consultado el 25 de enero de 2016.

[3] page 135

[4] <http://www.proyectosagiles.org/beneficios-de-scrum>

9 Referencias

- (PDF) Rising, L., Janoff, N.S. (2000). *The Scrum Software Development Process for Small Teams* Retrieved March 15, 2007
- (PDF) Schwaber, K. Advanced Development Methods. *SCRUM Development Process* Retrieved July 01, 2010
- (video) Jeff Sutherland in *Scrum Tuning: Lessons learned from Scrum implementation at Google* Retrieved 2007-12-15
- (video) Ken Schwaber in *Scrum et al.* Retrieved 2008-01-19

10 Véase también

- Agilística
- Arquitectura orientada a servicios
- Back office
- Ciclo de vida del producto
- Desarrollo ágil de software
- Desarrollo de software
- Kanban
- Lean software development

11 Enlaces externos

- Libro gratuito sobre Scrum
- “Scrum y XP desde las trincheras”, traducción de “Scrum and XP from the trenches” por Henrik Kniberg
- Artículo de introducción a Scrum
- PPT reutilizable de introducción a Scrum, original de Mike Cohn, traducción de Ernesto Grafeuille

12 Origen del texto y las imágenes, colaboradores y licencias

12.1 Texto

- **Scrum** *Fuente:* <https://es.wikipedia.org/wiki/Scrum?oldid=89082779> *Colaboradores:* Julie, Rosarino, Ecelan, Dodo, Ascánder, Fritz11, Jgalgarra, Jdemarcos, Mescalier, RobotQuistnix, Alhen, Yrbot, Oscar ., FlaBot, BOTijo, Martingala, Maldoror, Jago84, Qwertyytrewqq-werty, CEM-bot, Alexav8, Mansoft, Osepu, Juan.palacio, Ray iceman, Thijs!bot, Mpeinadopa, TXiKiBoT, Kijote, LKF, Jmbeas, VolkovBot, Technopat, Penelopina, Galandil, Muro Bot, J.M.Domingo, Mushii, Loveless, Sindestino, Marcelo, Saulnoelm, Fanattiq, Poco a poco, Hernaldo, Susibel, Santiagobas, LucienBOT, MastiBot, Adelpine, Ginosbot, Diegusjaimes, Caskete, Saloca, Kamusclown, Mac1800, Ynnek, Billinghamurst, Jmdmmx, Xavier Albaladejo, C lamarque, Parlamento, Xqbot, Jkbw, Aquileaaquilea, Alex of Bcn, Dossier2, Ragg-ha-eswiki, Fpablos, Jacorream, BokimBot, MondalorBot, Hprmedina, Moises003, RedBot, Solde, Leugim1972, PatruBOT, Tarawa1943, RNL89, GrouchoBot, EmausBot, Savh, ZéroBot, Hhiroshi, Fran.naranjo, EGA Futura, WikitanvirBot, Leoparra86, Diamondland, Hcquin-teroz, Dieguen, Tokvo, Deibyod, Edc.Edc, KLBot2, Obed.mx, Arturo.Van, N.garbezza, Allan Aguilar, Mega-buses, Takumi 1, YFdyh-bot, Rotlink, Maxie Ayala, Addbot, Henaldog, Balles2601, Mhidalgolache, Treiscort, Miguel Visuete, Vegafernando, Jarould, Chavesrd, Iafar86 y Anónimos: 188

12.2 Imágenes

- **Archivo:Ciclos_de_desarrollo.png** *Fuente:* https://upload.wikimedia.org/wikipedia/commons/8/82/Ciclos_de_desarrollo.png *Licencia:* CC BY 2.5 *Colaboradores:* No machine-readable source provided. Own work assumed (based on copyright claims). *Artista original:* No machine-readable author provided. Juan.palacio assumed (based on copyright claims).
- **Archivo:Ficha_scrum.png** *Fuente:* https://upload.wikimedia.org/wikipedia/commons/a/a5/Ficha_scrum.png *Licencia:* CC BY-SA 2.5 *Colaboradores:* No machine-readable source provided. Own work assumed (based on copyright claims). *Artista original:* No machine-readable author provided. Juan.palacio assumed (based on copyright claims).
- **Archivo:Scrumm.PNG** *Fuente:* <https://upload.wikimedia.org/wikipedia/commons/e/e5/Scrumm.PNG> *Licencia:* CC BY-SA 3.0 *Colaboradores:* Trabajo propio *Artista original:* Maxie Ayala

12.3 Licencia del contenido

- Creative Commons Attribution-Share Alike 3.0