

INSTITUTO TECNOLÓGICO SUPERIOR ZACATECAS OCCIDENTE

MATERIA: FUNDAMENTOS DE INGENIERÍA DE SOFTWARE

# TEMA: SCRUM

# DOCENTE:

I.S.C. ERIKA JAZMÍN ROBLES GÓMEZ

ELABORADO POR:

OSIEL CHÁVEZ FLORES

RUBÉN GÓMEZ BARRIENTOS

OSIEL BARRIENTOS RAMÍREZ

FABIÁN ARMANDO HERRERA AVALOS

JESÚS AGUSTÍN JUÁREZ GUERRERO

**FECHA:** 5 DE DICIEMBRE DEL 2014

**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 En su estudio, Nonaka y Takeuchi compararon la nueva forma de trabajo en equipo, con el avance en formación de melé 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, 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.

**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 ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders, y el Team que incluye a los desarrolladores. Durante cada sprint, un periodo entre una y cuatro semanas, el equipo crea un incremento de software potencialmente entregable. 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. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owneridentifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. 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 autoorganizados 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, 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.

**Roles en Scrum**

**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 escribehistorias 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 utiliza 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. Un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

**Roles Auxiliares**

Los roles auxiliares en los "equipos Scrum" 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. 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 o Clientes, Proveedores, Vendedores, etc**

Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirán el beneficio acordado que justifica su producción. Sólo participan directamente durante las revisiones del sprint.

**Administradores (Managers)**

Es la gente que establece el ambiente para el desarrollo del producto.

**Reuniones en Scrum**

**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 ocurrir en la misma ubicación y a la misma hora todos los días.

Durante la reunión, cada miembro del equipo contesta a tres preguntas:

¿Qué has hecho desde ayer?

¿Qué es lo que harás hasta la reunión de mañana?

¿Has tenido algún problema que te haya impedido alcanzar tu objetivo?

**Scrum de Scrum**

Cada día normalmente después del “Daily Scrum”:

Estas reuniones permiten a los grupos de equipos discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración.

Asiste una persona asignada por cada equipo.

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?

**Reunión de Planificación del Sprint**

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 hacen dos reuniones más: la reunión de revisión del Sprint y la retrospectiva del Sprint.

**Reunión de Revisión del Sprint**

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

**Retrospectiva del Sprint**

Después de cada sprint, se lleva a cabo una retrospectiva del 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.

**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 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 potencialmente entregable al cliente. Asimismo, se recomienda no agregar objetivos al sprint o sprint backlog a menos que la falta de estos objetivos amenace al éxito del proyecto. La constancia permite la concentración y mejora la productividad del equipo de trabajo.

**Documentos**

**Product backlog**

El product backlog es un documento de alto nivel para todo el proyecto. Contiene descripciones genéricas de todos los requisitos, funcionalidades deseables, etc. priorizadas según su retorno sobre la inversión. Es el qué va a ser construido. 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 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.

**Sprint backlog**

El sprint backlog es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horaspero 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 oportuno.

**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. 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.

**Scrum aplicado al desarrollo de software**

Scrum es el término dado por Nonaka y Takeuchi al método de desarrollo de nuevos productos realizado con equipos reducidos, multidisciplinares, que trabajan con comunicación directa y empleando ingeniería concurrente, en lugar de ciclos o fases secuenciales. Esta forma de trabajo logra niveles de eficiencia y valor en el producto superiores a los obtenidos con ingeniería secuencial y producción basada en procesos. En los 80, Nonaka y Takeuchi analizaron esta forma de producción, observando cómo trabajaban los equipos de las empresas tecnológicas que lograban mayores niveles de eficiencia y valor en sus productos Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox y Hewlett-Packard. Diez años más tarde Ken Schwaber presentó en OOPSLA la descripción de la implementación de Scrum para software que él empleaba en el desarrollo de Delphi. Las implementaciones de Scrum para desarrollo de software se vienen enriquecendo desde entonces, y poco tienen que ver las implementaciones actuales con la original de Ken. Ahora es muy raro que alguien configure un campo de Scrum con los controles originales el Backlog único ha evolucionado a Backlog de producto y Backlog de Sprint. También es habitual usar un backlog estratégico o "Epics" de producto. La evolución añadió a la reunión de revisión de sprint, otra de inicio y más tarde otra de retrospectiva. Tampoco se suele usar la fase de cierre, etc.

**BIBLIOGRAFÍA**

* [(PDF) Rising, L., Janoff, N.S. (2000). *The Scrum Software Development Process for Small Teams*](http://web.archive.org/web/20080516073429/http:/members.cox.net/risingl1/Articles/IEEEScrum.pdf) Retrieved [March 15](http://es.wikipedia.org/w/index.php?title=March_15&action=edit&redlink=1), [2007](http://es.wikipedia.org/wiki/2007)
* [(PDF) Schwaber, K. Advanced Development Methods. *SCRUM Development Process*](http://jeffsutherland.com/scrumpapers.pdf) Retrieved [July 01](http://es.wikipedia.org/w/index.php?title=July_01&action=edit&redlink=1" \o "July 01 (aún no redactado)), [2010](http://es.wikipedia.org/wiki/2010)
* [(video) Jeff Sutherland in *Scrum Tuning: Lessons learned from Scrum implementation at Google*](http://www.youtube.com/watch?v=9y10Jvruc_Q) Retrieved [2007-12-15](http://es.wikipedia.org/w/index.php?title=2007-12-15&action=edit&redlink=1)
* [(video) Ken Schwaber in *Scrum et al.*](http://www.youtube.com/watch?v=IyNPeTn8fpo) Retrieved [2008-01-19](http://es.wikipedia.org/w/index.php?title=2008-01-19&action=edit&redlink=1)

**Véase también**

* [Agilmática](http://es.wikipedia.org/w/index.php?title=Agilm%C3%A1tica&action=edit&redlink=1)
* [Arquitectura orientada a servicios](http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios)
* [Back office](http://es.wikipedia.org/wiki/Back_office)
* [Ciclo de vida del producto](http://es.wikipedia.org/wiki/Ciclo_de_vida_del_producto)
* [Desarrollo ágil de software](http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software)
* [Desarrollo de software](http://es.wikipedia.org/wiki/Desarrollo_de_software)
* [Kanban](http://es.wikipedia.org/wiki/Kanban)
* [Lean software development](http://es.wikipedia.org/wiki/Lean_software_development)

**Enlaces externos**

### Comunidades o grupos de usuarios

* [Asociación Agile-Spain](http://www.agile-spain.org/) : la comunidad agilista en España
* [Ágiles](http://agiles.org/) : la comunidad agilista latinoamericana
* [ScrumManager](http://www.scrummanager.net/ok) : Plataforma profesional de conocimiento libre
* [Comunidad ágil Colombia](http://www.agilespin.com/) : Comunidad agilista Colombia
* [Comunidad agilmática](http://comunidad.agilmatica.com/)
* [Grupo de usuarios de Scrum en Cataluña](http://www.scrum.cat/)

### Libros originales o traducciones al español

* [Libro gratuito sobre Scrum](http://www.navegapolis.net/content/view/694/61/)
* [Traducción de "The Scrum Primer"](http://scrumtraininginstitute.com/library)
* ["Scrum y XP desde las trincheras", traducción de "Scrum and XP from the trenches" por Henrik Kniberg](http://www.proyectalis.com/2008/02/26/scrum-y-xp-desde-las-trincheras/)

### Portales temáticos y blogs

* [Scrum Manager: Comunidad ágil de habla hispana. Cursos online gratuitos](http://www.scrummanager.net/)
* [proyectosAgiles.**org**: base de conocimiento gratuita de Scrum en español](http://www.proyectosagiles.org/)
* [Metodología de gestión de proyectos basada en Scrum](http://navegapolis.metocube.com/)
* [Artículos sobre Scrum de Rodrigo Corral](http://geeks.ms/blogs/rcorral/archive/tags/Scrum/default.aspx)
* [Thinking-in-process](http://b-efficient.blogspot.com.es/) Blog sobre procesos, gestión de proyectos y Scrum

### Artículos y otros recursos

### [Hoja Excel para gráficos de progreso scrum](http://www.navegapolis.net/index.php?option=com_content&task=view&id=268&Itemid=84)

* [Artículo de introducción a Scrum](http://www.chuidiang.com/ood/metodologia/scrum.php)
* [Explicando Scrum a mi abuela por Jorge Serrano](http://geeks.ms/blogs/jorge/archive/2007/05/09/explicando-scrum-a-mi-abuela.aspx)
* [Artículo sobre Scrum en Español](http://swsaber.com/scrum/)
* [PPT reutilizable de introducción a Scrum, original de Mike Cohn, traducción de Ernesto Grafeuille](http://www.mountaingoatsoftware.com/system/asset/file/66/SpanishRedistributableIntroToScrum.ppt)
* [Sencillo generador online de gráficas burndown](http://www.burndowngenerator.com/)
* [Herramientas y Software para Scrum](http://www.unbugalavez.net/2008/03/herramientas-para-scrum.html)
* [Notas de Scrum](http://sedici.unlp.edu.ar/handle/10915/5519)
* [Una comparación entre Scrum y DevOps](http://b-efficient.blogspot.com.es/2012/09/introduccion-devops-el-siguiente-paso.html)
* [Scrum en 1 página](http://www.scribd.com/doc/134880180/SCRUM-La-guia-de-1-pagina)
* Scrum *Te*oría revolucionara, *Amo*rtizando los errores para controlarlos *Ana*lizarlos y solucionarlo al momento