

SCRUM development process

Ing. Natalia G. Caso

Universidad Tecnológica Nacional

Facultad Regional Buenos Aires

Septiembre 2004

natalia_caso@yahoo.com.ar

1. INTRODUCCIÓN

Las metodologías tradicionales para el desarrollo de software imponen un proceso definido. Sin embargo, un gran número de procesos de desarrollo no están completamente definidos, aunque se los trata como si lo estuvieran. SCRUM proporciona una **metodología** que trata a estos procesos como una **caja negra controlada** aplicando [mecanismos apropiados](#) para su control.

2. DEFINICIÓN

SCRUM es una metodología para gestión, mejora y mantenimiento de un sistema nuevo o existente.

SCRUM se concentra en como los miembros del equipo deberían funcionar a fin de producir un sistema flexible en un entorno que cambia constantemente.

Permite a los equipos de desarrollo **adaptarse** dentro de un ambiente complejo utilizando **procesos imprecisos**.

El origen de la palabra SCRUM procede del rugby: acto de preparar el avance del equipo en unidad pasando la pelota a uno y otro jugador. Igual que el juego, SCRUM es adaptativo, ágil, auto-organizante y con pocos tiempos muertos.

3. CARACTERÍSTICAS

- Empírico
- Iterativo
- Flexible
- Equipos auto-organizativos

Los **requerimientos** se capturan como ítems en un listado (product backlog)

SCRUM considera que el proceso del desarrollo de software involucra muchas variables de entorno y técnicas tales como: requerimientos, tiempo, recursos, costo, competencia, calidad y tecnología. Es probable que todas ellas cambien durante el proceso.

El **entregable** se determina durante el proyecto basándose en el **entorno**.

SCRUM utiliza técnicas de **orientación a objetos** para la construcción de los entregables.

Los proyectos SCRUM pueden **estimarse** empleando el método de **function points**. Sin embargo, la estimación sirve solo **para el comienzo**, debido a que el tiempo y el costo son determinados dinámicamente en respuesta a factores del entorno.

4. ORIGEN

SCRUM fue desarrollada por Ken Schwaber (Advanced Development Methods), Jeff Sutherland y Mike Beedle.

Como metodología ágil específicamente referida a ingeniería de software, SCRUM fue aplicado por Jeff Sutherland y elaborado más formalmente por Ken

Schwaber. Poco después Sutherland y Schwaber se unieron para refinar y extender SCRUM.

En SCRUM se aplicaron principios de procesos de control industrial, junto con experiencias metodológicas de Microsoft, Borland y Hewlett-Packard. Schwaber, en particular, había trabajado con científicos de Du Pont para comprender mejor los procesos definidos de antemano, y notaron que a pesar que **CMM** se concentraba en hacer que los **procesos** de desarrollo se tornaran **repetibles, definidos y predecibles**, muchos de ellos eran formalmente impredecibles e irrepitibles porque **cuando se está planificando** no hay primeros principios aplicables, **los procesos recién comienzan a ser comprendidos** y son complejos por naturaleza. Schwaber se dio cuenta entonces de que **un proceso necesita aceptar el cambio, en lugar de esperar predictibilidad**.

5. DIFERENCIA CON OTRAS METODOLOGÍAS

Comparación con el método iterativo:

SCRUM asume que todos los requerimientos no se conocen en profundidad y que el camino más rápido para explorarlos e implementarlos será descubierto empíricamente durante el proceso de desarrollo.

Comparación con metodologías “Waterfall” y espiral:

Las metodologías “**Waterfall**” y “**Espiral**” establecen el contexto y la **definición** de los **entregables** al **comienzo del proyecto**. En cambio, SCRUM y las metodologías iterativas inicialmente plantean el contexto y una

definición global de entregables y luego el **entregable irá evolucionando durante el proyecto en base al entorno**. SCRUM confirma que la definición de los procesos de desarrollo no están completamente definidos y utiliza mecanismos de control para mejorar la flexibilidad.

Comparación con CMM:

Los procesos de CMM tales como **testing e inspecciones** pueden ser utilizados en proyectos que utilizan SCRUM, pero la diferencia radica en que los mismos se utilizarán “**on demand**”; si el equipo siente que no se necesitan, no se utilizan.

6. PROCESO SCRUM

El proceso de SCRUM se divide en 3 grandes etapas:



Fig. 1 –Etapas de SCRUM

SCRUM trata la fase de desarrollo como una caja negra que requiere controles externos. Los controles se ponen en cada iteración de la fase de desarrollo para evitar el caos y a su vez maximizar la flexibilidad.

El proyecto está abierto al entorno hasta la fase de cierre. El entregable puede ser cambiado en cualquier momento durante las fases de inicio y desarrollo del proyecto.

Inicio (Pregame)

Esta etapa está conformada por dos subetapas:

- **Planeamiento:** Consiste en establecer la visión, el presupuesto, forma de financiamiento y el backlog del producto. En esta fase

se selecciona que funcionalidad es la más apropiada para desarrollo inmediato. También se establece el equipo de trabajo, se evalúan las herramientas de desarrollo y se define la fecha de entrega (es una fecha aproximada).

- **Arquitectura:**
Esta fase consiste en la conceptualización y análisis. Si el proyecto se trata de la mejora de un nuevo sistema, sólo se realiza un análisis limitado. Se realiza un diseño de alto nivel para actualizar los modelos del dominio y reflejar el contexto del nuevo sistema y los requerimientos y las modificaciones necesarias de la arquitectura del sistema. Los diseñadores y arquitectos dividen el proyecto en paquetes basándose en los ítems del backlog. En la jerga de SCRUM se llaman “paquetes” a los objetos o componentes que necesitan cambiarse en cada iteración.

Desarrollo (Game)

En esta etapa se realiza el desarrollo propiamente dicho. También se la conoce como “Ingeniería concurrente”. La misma se divide en iteraciones que proveen como resultado funcionalidades incrementales al fin de cada una de ellas. Dichas iteraciones se llaman **sprints**.

Un sprint es el proceso de adaptación a las variables que cambian con el entorno.

Un sprint dura aproximadamente entre una semana y 30 días. **Cada sprint incluye las fases tradicionales del desarrollo de software:**

requerimientos, análisis, diseño, desarrollo y entrega.

Los riesgos son evaluados continuamente y se establecen [mecanismos de control](#) adecuados.

Durante un sprint no se utilizan diagramas de gantt para seguimiento de tareas (estos parten del supuesto que las tareas de un proyecto se pueden identificar y ordenar), debido a que el desarrollo es semi-caótico y cambiante como para que se le aplique un proceso definido.

Durante un sprint no se pueden cambiar los miembros del equipo scrum. Tampoco pueden introducirse cambios durante un sprint (si surge algún cambio se incluirá en el backlog del producto, pero no en el del sprint).

El scrum master mantiene el sprint backlog. Actualiza las tareas finalizadas y para las que no lo están, el tiempo que el equipo piensa que tomará para terminarlas.

En cada sprint se realizan las siguientes actividades:

- **Planeamiento**
Consiste en una reunión de planeamiento. Ésta comprende dos fases. La **primera** consiste en decidir los **objetivos y la funcionalidad** a incluir en el sprint. La **segunda** consiste en establecer como ésta funcionalidad se **implementa** durante el sprint.

El product owner establece **prioridades** entre los ítems del product backlog y el equipo scrum determina cuales son las tareas que pueden completar durante el sprint partiendo de las que tienen más prioridad.

Se realizan estimaciones de los ítems del product backlog, pero

estas estimaciones solo sirven para asignar items al sprint.

Durante el sprint el equipo volverá a estimar los ítems del sprint backlog.

Para cada sprint se establecerán que funcionalidades del product backlog definido inicialmente se van a implementar. Ese conjunto de funcionalidades conformarán el “***sprint backlog***”.

- ***Desarrollo***
Consiste en definir los cambios para la implementación de los requerimientos del backlog en los paquetes, abrir dichos paquetes: realizar análisis, diseño, desarrollo, implementación, testeo y documentación de los cambios.
- ***Envoltura (Wrap)***
Consiste en cerrar los paquetes: creación de una versión ejecutable de los cambios y establecer como ésta implementa los requerimientos del backlog.
- ***Revisión***
Consiste en realizar una reunión de revisión para presentar el trabajo y resolución de problemas emergentes. Se revisan los riesgos y se definen las respuestas apropiadas.

- ***Ajuste***
Consiste en consolidar la información recolectada de la revisión en los paquetes afectados, esto incluye nuevas propiedades y nuevo look & feel.

Reuniones Scrum:

Durante un sprint, todos los días se realizan reuniones llamadas “SCRUM”. El objetivo de las mismas es quitar los impedimentos que le surgen a los miembros del equipo scrum.

Cada una de ellas dura aproximadamente 15 minutos. A cada miembro del equipo scrum se le pregunta:

¿Qué hizo durante las últimas 24 horas?
¿Qué planea hacer las próximas 24 horas?
¿Qué obstáculos se le han presentado en las últimas 24 horas ?

Estas reuniones deben realizarse obligatoriamente.

Cierre (Postgame)

Esta etapa comienza cuando el equipo de management decide que las variables de entorno, tales como los requerimientos se han completado. En esta etapa se genera la documentación final, se realiza el testing pre-lanzamiento y el lanzamiento propiamente dicho.

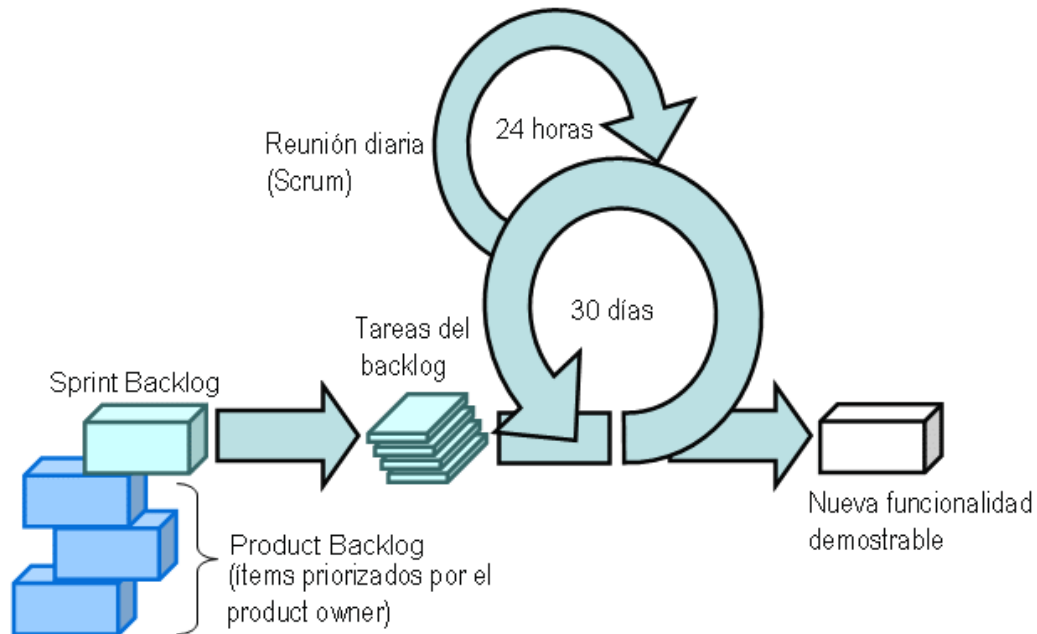


Fig. 2 – Proceso SCRUM

7. CONTROLES EMPLEADOS EN SCRUM:

Para controlar las variables de entorno y técnicas SCRUM emplea ciertos controles.

El control de riesgos es el principal, ya que el mismo conduce a cambios en otros controles.

Los controles empleados en SCRUM son los siguientes:

- **Backlog:** requerimientos sobre la funcionalidad del producto
- **Release:** ítems del backlog que en algún momento representan un release viable basado en las variables de requerimientos, tiempo, calidad y competencia.
- **Paquetes:** componentes del producto que deben ser cambiados para implementar ítems del backlog de un nuevo release.
- **Cambios:** Cambios que deben ocurrir en un paquete para implementar ítems del backlog
- **Problemas:** Problemas técnicos que ocurren y deben resolverse para implementar un cambio.
- **Riesgos:** que afectan el éxito del proyecto son continuamente evaluados y continuamente se planifican respuestas a los mismos. Otros controles se ven afectados como resultado de la evaluación de riesgos.

- **Soluciones:** a los problemas y riesgos
- **Temas:** Temas del proyecto que no son definidos en términos de paquetes, cambios y problemas

La gerencia emplea estos controles para gestionar el backlog del producto.

El equipo emplea estos controles para gestionar cambios y problemas. Ambos gestionan temas, riesgos y soluciones.

Estos controles son revisados y modificados en cada reunión de revisión de Sprint.

8. ROLES SCRUM

Existen seis roles en SCRUM que tienen diferentes tareas y responsabilidades durante el proceso.

- **Scrum master:** Es el responsable de llevar a cabo el proyecto de acuerdo a las reglas SCRUM, interactúa con el equipo y los clientes y también es **responsable de remover todo tipo de impedimento para el desarrollo.**
No asigna tareas a cada miembro del equipo ni toma decisiones para el equipo, los equipos son auto-organizativos. Hace más de “coach” que de líder.
- **Dueño del producto (Product owner):** Es oficialmente responsable de gestionar el proyecto y de hacer visibles los ítems del Product Backlog. Toma las decisiones finales sobre los ítems del product backlog.

Participa en la estimación de los ítems del backlog.

- **Equipo scrum:** Es el equipo del proyecto que tiene la autoridad para organizarse a sí mismo a fin de lograr los objetivos de cada sprint. Sus funciones son crear el sprint backlog, estimar los ítems del sprint backlog, revisar el product backlog y sugerir impedimentos para que estos se quiten del proyecto.
- **Cliente:** Participa en tareas relacionadas al product backlog.
Es importante que el cliente se encuentre involucrado en el proyecto debido a que continuamente provee feedback en el diseño del producto, específicamente al final del sprint.
- **Gerencia:** Toma las decisiones finales sobre los estándares y convenciones adoptadas en el proyecto. También participa en el establecimiento de objetivos y requerimientos.

9. VENTAJAS

- Al realizar reuniones periódicas y focalizarse en la remoción de los impedimentos recientes, evita los estancamientos en el proyecto.
- Seguimiento del proyecto: debido a las reuniones diarias efectuadas, ya que solo se pregunta lo que se hizo y lo que se hará durante el próximo día, lo cual no genera pérdida de tiempo. El sprint backlog se va actualizando continuamente,

lo cual permite conocer exactamente el estado de avance del ciclo.

- Seguimiento del equipo: también se debe a las reuniones diarias efectuadas, se puede ver que está haciendo cada uno de los miembros del equipo.
- Las pruebas y la documentación se realizan en paralelo con el desarrollo y se obtiene software “tangible” que puede incrementar su funcionalidad en cada sprint.
- Provee mecanismos de control para planear el lanzamiento de un producto y en consecuencia gestionar las variables a medida que avanza el proyecto. Esto posibilita a las organizaciones cambiar el proyecto y los entregables en cualquier punto del tiempo, entregando el lanzamiento más apropiado.
- Permite a los desarrolladores encontrar las soluciones más ingeniosas a lo largo del proyecto a medida que van aprendiendo y el entorno cambia.
- Mejora continua: SCRUM permite una reingeniería rápida, ágil y continua
- Se logra progresar en el producto aún cuando los requerimientos no son muy estables
- Aumenta la comunicación con el equipo

- El cliente obtiene feedback frecuente de cómo funciona realmente el producto

10. DESVENTAJAS

- SCRUM requiere que la gerencia delegue autoridad para la toma de decisiones al equipo SCRUM, incluso permitiéndoles fallar, si es necesario.
- SCRUM es una metodología diferente: las personas suelen ser resistentes al cambio y algunas no se sienten cómodas con la responsabilidad que genera formar parte del equipo SCRUM.

11. SCRUM Y XP

SCRUM es una metodología de desarrollo de productos. No posee ningún tipo de prácticas de ingeniería. A diferencia de ésta, XP es una metodología de ingeniería que consiste en prácticas que aseguran la calidad focalizada en el código. Sin embargo, XP no posee prácticas de management.

A pesar de sus diferencias ambas comparten los siguientes aspectos:

- Iteraciones
- Incrementos
- Solo se desarrolla la funcionalidad que el cliente pidió para la próxima iteración
- Equipos auto-organizativos

Se ha experimentado la aplicación de ambas metodologías en proyectos

realmente caóticos. Por un lado se aplicaron las siguientes prácticas SCRUM:

- Empleo de sprints y reuniones de planeamiento de sprints
- Reuniones diarias (SCRUM)
- Utilización del product backlog y priorizado por el product owner
- Sprint backlog
- Rol de scrum master

Por otro lado se aplicaron las siguientes prácticas xp:

- Diseño del sistema simple
- Testear antes de codificar
- Integración continua
- Refactoring
- Programación de a pares

Resultados:

El hecho de haber aplicado XP produjo un incremento en la calidad del código y en los test suites.

Por otro lado, el hecho de haber aplicado SCRUM produjo un incremento en la productividad del equipo de trabajo debido a que SCRUM se focaliza en entregar funcionalidad usable al cliente en vez de entregar artefactos de valor cuestionable. Por lo tanto, combinar SCRUM con XP incrementa la productividad y la calidad del código desarrollado.

12. GLOSARIO

Sprint: Ciclo que dura aproximadamente entre una semana y 30 días en el cual se realizará un trabajo para crear un entregable.

Backlog (registro): Los elementos del registro pueden ser requerimientos funcionales, no funcionales, correcciones de bugs,

restricciones, mejoras requeridas y actualizaciones de tecnología.

Product Backlog (registro del producto): Lista de requerimientos especificados por el cliente. Los mismos se encuentran priorizados. La lista se origina a partir de una variedad de fuentes. Pueden participar personal de marketing y ventas.

Comprende un listado de requerimientos técnicos y del negocio. El product backlog irá creciendo y cambiando a medida que se vaya aprendiendo más acerca del producto y sus clientes.

Product Backlog	Fecha:
	Estimado:
Tipo: Nuevo ___ Mejora ___ Arreglo: ___	Fuente:
Descripción	
Notas	

Sprint Backlog (registro del ciclo): Lista de requerimientos a implementarse en el sprint. Se descomponen en pequeñas tareas. **A diferencia del product backlog el sprint backlog es estable hasta que finalice el sprint.**

Sprint Backlog:	Fecha:
Propietario:	Trabajo Pendiente/Fecha
Status: Pendiente ___ Activo ___ Completo ___	
Descripción:	
Notas:	

SCRUM: Reunión diaria de 15 minutos de duración en la cual se

pretende remover impedimentos al equipo.

prioridad establecida por el product owner.

En proyectos caóticos tiende a estabilizar momentáneamente el conjunto de requerimientos (mediante el congelamiento del sprint backlog) para que los desarrolladores puedan focalizarse en la implementación de esos requerimientos y no en la gestión de nuevos cambios. Así, pueden avanzar y entregar una nueva funcionalidad al final de cada ciclo.

13 . CONCLUSIONES

SCRUM es una metodología de desarrollo de software en la que el entregable debe adaptarse al entorno. Ésto se debe a que el negocio se irá comprendiendo a lo largo del proceso de desarrollo.

Potencia la comunicación tanto interna (con los miembros del equipo) como externa (feedback del / al cliente) para agilizar el proceso de desarrollo.

Un aspecto positivo a destacar es que se entrega la funcionalidad más importante en el primer ciclo debido a que los ítems de product backlog se ordenan por la

14 . REFERENCIAS

<http://www.controlchaos.com>
<http://jeffsutherland.org/scrum/>
<http://www.mountangoatsoftware.com/>
<http://www.martinfowler.com>
<http://computerworld.com>
<http://books.slashdot.org/>