

Notas de Scrum.

Licenciado Villarreal, Gonzalo Luján.

Sólo en uno de cada tres proyectos de software se cumple el plan inicial: el sistema realiza las funcionalidades inicialmente previstas, y se desarrolla sin desbordar ni agenda ni presupuesto.

Introducción.

La ingeniería de software es una ciencia relativamente nueva, tan nueva como el concepto de software en sí. Como toda ciencia, evoluciona constantemente, crece, incorpora nuevas herramientas, metodologías y conceptos. Esta evolución natural ha llevado al desarrollo de métodos de desarrollo de proyectos denominados “agiles”, en lo cuales se deja de lado la excesiva documentación y reglamentación, típico de métodos orientados a procesos o predictivos (donde de antemano se pretende conocer todo: objetivos, requisitos, tiempos, personas, costes) y se priorizan otros factores fuertemente ligados con el dinamismo que caracteriza al sector (personas que desarrollan, el cambio y constante adaptación, innovación).

Con el nacimiento del concepto de Agile Programming surgieron algunas técnicas, como ser Extreme Programming (XP), DSDM, Scrum, Adaptive Software Development ASD, Pragmatic Programming (PP), Agile Modeling (AM) y Feature Driven Development (FDD). En este artículo nos centraremos en el método Scrum y veremos que nuevos conceptos introduce, sus características y como se lleva adelante un proyecto con este método.

Cambios conceptuales.

La Ingeniería de Software tradicional considera como método de desarrollo una serie de etapas o fases que se realizan en cascada, donde cada fase comienza una vez que ha finalizado la anterior. Eventualmente, se considera cierto solapamiento entre una fase y su sucesora. En scrum, todas las fases se solapan constantemente, intentando maximizar la concurrencia.

No se consideran equipos con personas especializadas en cada una de las áreas, sino que se piensa en un equipo multidisciplinario donde todos puedan hacer, al menos básicamente, cualquiera de las partes del desarrollo. Si bien es cierto que en cualquier grupo de desarrollo existen personas mejor capacitadas para interactuar con la base de datos, o para programar en objetos, desarrollar interfases de comunicación, etcétera, la idea es que todos puedan eventualmente realizar cualquier tarea, solucionar cualquier problema en cualquier área y principalmente, dejar en claro que nadie es 100% imprescindible.

Al ser un método ágil, no se tienen a priori todos los requisitos del sistema detallados, sino se van descubriendo a medida que el producto se va desarrollando. Lo que si se tiene es una visión del producto, un concepto de "Que se quiere, a donde se quiere llegar". Se parte de una visión muy general, y paulatinamente se va descubriendo durante el desarrollo, generando el detalle buscado paralelo al desarrollo. Aquí se observa claramente el solapamiento entre fases, ya que el análisis de requisitos se

realiza junto con el desarrollo del proyecto.

Debido a lo mencionado previamente, no existe un plan detallado y riguroso preestablecido. El plan tiene plazos cortos, y esta constantemente adaptándose a los cambios. Estos plazos se llaman iteraciones, y son ciclos que se repiten constantemente desde el inicio del proyecto hasta que el mismo finaliza. Luego de cada iteración, se obtiene un incremento, que es una nueva versión del software **funcional**, con un conjunto de mejoras introducidas en el último ciclo. Las iteraciones son de entre 2 y 6 semanas. A diferencia de Extreme Programming, Scrum NO admite cambios a mitad de un incremento (sprint, iteración)

Algunas características.

Hay incertidumbre. Esto es porque se parte de algo muy general que se va especificando paulatinamente. El equipo tiene un margen de libertad bien amplio, con gran autonomía y responsabilidad sobre las tareas (aquí se piensa en tareas en vez de procesos).

Los equipos son auto-organizados. No existe una persona que marca pautas o asigna tareas. Existe una persona que marca la dirección, pero no interviene en la organización de los equipos (muy similar a una empresa Start-up). Esto requiere de los equipos:

- autonomía (elección de estrategias de solución)
- auto-superación (evalúan sus soluciones y las mejoran)
- auto-enriquecimiento (aprenden unos de otros)

Las fases de desarrollo son ACTIVIDADES solapadas, o sea que se realizan en forma simultánea o a demandan (según cada iteración)

Para evitar el descontrol (dado que el equipo trabaja en un entorno ambiguo, inestable y autónomo), se establecen "puntos de control". No es un control rígido, sino que se debe generar un ecosistema adecuado para un "control entre iguales".

Se comparten las experiencias, se promueve la difusión de conocimiento entre las personas. Esto debe aplicarse tanto a nivel de proyecto como de organización.

El cliente es un miembro más del equipo, que se integra y colabora con el grupo de trabajo. Se evita la negociación contractual, rígida y predictiva.

Etapas.

Cada proyecto sigue una serie de etapas básicas y simples.

1. Concepto.

- Se crea la visión del producto o servicio que se quiere obtener.
- Se selecciona el o los equipos de trabajo.
- Se define el alcance del proyecto.
- Se comparte esta información con TODO el equipo.

2. Especulacion.

- El equipo construye hipótesis sobre la visión definida en 1, que SE SABE es muy general e insuficiente como para determinar implicaciones de un desarrollo (requisitos, diseños, costes)
- Se determina la primera aproximación a lo que se quiere construir, y se identifican las limitaciones del entorno.
- La gestión ágil investiga y desarrolla tomando como partida la visión del producto. Durante el desarrollo confronta la realidad de lo que va obteniendo. Su valor, posibilidades y la situación de negocio del entorno en cada momento.
- Esta etapa de repite en cada iteración! Consiste básicamente en:
 - revisión de requisitos generales
 - desarrollo de una lista de funcionalidades esperadas
 - construcción de un plan de entrega: fechas para las versiones, hitos e iteraciones (se refleja el esfuerzo que requerirá el proyecto)
 - estrategias/planes para la gestión de riesgos

3. Exploracion.

Desarrollo de las funcionalidades que el equipo determinó necesarias (en la etapa anterior) para llegar al siguiente incremento del producto

4. Revision.

La realizan el equipo y los usuarios, sobre el producto real. Se analiza si aún se está alineado con el objetivo.

5. Cierre.

El producto se termina de acuerdo a lo fijado en 1) y a lo revisado (varias veces) en 2).

El proyecto no termina necesariamente aquí, pero ahora seguirá una etapa de GESTION, donde se agregan sucesivas mejoras y nuevas versiones para no quedar obsoleto y permanecer en el mercado. Sería la etapa de mantenimiento, pero en realidad es la continuidad del proyecto (en ciclos incrementales) para ir acercándose a la visión del producto, que también es posible que vaya evolucionando con el tiempo, al ritmo de su entorno tecnológico y de negocio.

Prioridades.

En Scrum se le da prioridad a la innovación antes que a la planificación (gestión predictiva, respetando fechas y planes). Por esto, los requisitos se diagraman lo menos posible. Si se diagrama demasiado, se limitan las posibilidades de limitación.

Nos apoyamos mucho en el valor de las personas. No importan aquí los procesos (ya que no existen) sino las interacciones de las personas entre sí y lo que cada persona le aporta a cada tarea del proyecto. Por ello, el ambiente laboral no está basado en el control, existe muy poca normalización y mínima jerarquía. (primer manifiesto Ágil: el protagonismo deben tenerlo las personas, y no los procesos).

El Scrum Management es un modelo de gestión de conocimiento "tácito". El gestor no aporta "ejecución" para implantar un modelo, sino "talento" para desarrollar su propio modelo. De aquí, no consiste en aplicar un modelo, a modo de proceso, sino en construir uno con su conocimiento sobre

esas bases.

El modelo.

El desarrollo se inicia desde la visión general de producto, dando detalle solo a las funcionalidades que, por ser las de mayor prioridad para el negocio, se van a desarrollar en primer lugar, y pueden llevarse a cabo en un periodo de tiempo breve (entre 15 y 60 días).

Cada uno de los ciclos de desarrollo es una iteración (sprint) que produce un incremento terminado y operativo del producto.

Scrum gestiona su evolución a través de **reuniones breves** de seguimiento en las que todo el equipo revisa el trabajo realizado desde la reunión anterior y el previsto hasta la reunión siguiente. Las reuniones son DIARIAS. La evolución de los proyectos se controla de forma empírica y adaptable.

Como funciona el proceso completo?

Cuando comienza cada sprint (15-60 días), se determinan que partes se van a construir (en base a la prioridad y a la cantidad de trabajo "manejable").

Reuniones.

Planificación del sprint.

Previa al inicio de cada sprint. Se determina cual es el trabajo y cuales son los objetivos a cumplir. Se genera el sprint backlog (lista de tareas). Aquí se determina también el "Objetivo del Sprint": lema que define la finalidad de negocio a lograr.

Seguimiento del sprint.

Breve reunión diaria. Se repasa el avance de cada tarea y el trabajo previsto para la jornada. Cada participante de la reunión responde 3 preguntas simples:

- a. Que hice desde la reunión anterior?
- b. Que voy a hacer de aquí a la próxima reunión?
- c. Que impedimentos debo solventar para poder realizar el trabajo.

Revisión del sprint.

Análisis y revisión del incremento generado. Es simplemente una presentación normal de resultados, los miembros del equipo no deben verlo como un acontecimiento especial o destacado, deben tomarlo como algo común, como una tarea más.

Elementos.

Con las sucesivas reuniones se generan algunos documentos. Si bien Scrum prioriza siempre el diálogo directo entre las personas antes que la excesiva documentación, es claramente necesario dejar asentado todo lo conversado, por una cuestión organizativa y porque el concepto "a las palabras se las lleva el viento" se aplica en todos los entornos, incluso aquí.

Product backlog.

Seria el documento de requisitos del sistema. Se parte de la visión del resultado que se desea obtener; y evoluciona durante el desarrollo . Es un listado de características a obtener, en orden de prioridad. El documento esta en constante evolucion, y aportan todas las personas que intervienen en el desarrollo.

El responsable del backlog es una UNICA persona y se denomina: propietario del producto.

Sprint backlog.

Lista de los trabajos que realizará el equipo durante el sprint para generar el incremento previsto. El equipo asume el compromiso de la ejecución. Las tareas están asignadas a personas, y tienen estimados el tiempo y los recursos necesarios.

Incremento.

Resultado de cada sprint. Se trata de un resultado completamente terminado y en condiciones de ser usado.

Roles.

Propietario del producto.

Es el responsable del producto. Es una UNICA persona, y conoce el entorno del negocio del cliente y de la vision del producto. Es el responsable del Product Backlog.

Es responsable de la financiación necesaria para el proyecto, de decidir cómo debe ser el resultado final, del lanzamiento y del retorno de la inversión

- En desarrollos internos puede ser el product manager, o responsable de marketing... quien asume este rol.
- En desarrollos para clientes externos lo más aconsejable es que sea el responsable del proceso de adquisición del cliente.

El equipo.

Todo el equipo de desarrollo conoce la metodología Scrum. El equipo cubre todas las habilidades necesarias para generar el resultado (multidisciplinario).

Se auto-gestiona y auto-organiza, y dispone de atribuciones suficientes en la organización para tomar decisiones sobre cómo realizar su trabajo

Scrum Manager o Scrum Master.

No es tanto un rol, sino que designa la responsabilidad de funcionamiento del modelo. En algunos casos puede ser necesario un rol exclusivo (una persona que se dedica a eso) y en otros puede ser mejor que las responsabilidades de funcionamiento las asuman los responsables del departamento de calidad o procesos, o del área de gestión de proyectos

Scrum considera un conjunto de herramientas para realizar el seguimiento (graficos burn-up, burn-down, juegos y protocolos de decision,...), de conceptos y metricas (tiempo real o tiempo de trabajo, tiempo teorico o tiempo de tarea, puntos de funcion, estimaciones, velocidad absoluta, velocidad relativa) y Valores:

- delegación de atribuciones -empowerment- al equipo que le permita auto-organizarse 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.
- reponsabilidad y auto-disciplina (no disciplina impuesta).
- trabajo centrado en el desarrollo de lo comprometido
- información, transparencia y visibilidad del desarrollo del proyecto

Herramientas.

Estas herramientas ayudaran a los equipos de trabajo a organizarse, trabajar juntos y aumentar la eficiencia.

Grafico Burn-up

Es una herramienta de planificación y seguimiento del propietario del producto, que muestra en un grafico muy simple el plan general de desarrollo del producto, y la traza de su evolución.

Grafico Burn-Down

Herramienta de seguimiento para el equipo, que muestra el avance del sprint día a día y revela de forma temprana posibles desviaciones. Es un grafico cartesiano que representa en el eje x los días laborables del sprint, y en el y la cantidad de esfuerzo estimada.

Juegos y protocolos de decisión

Estimación de póquer

Es un protocolo de trabajo agil, que resulta util en las reuniones en las que el equipo debe estimar el esfuerzo de las tareas con criterio de “juicio de expertos”.

Este protocolo de reunión evita los atascos de analisis 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 ademas... resulta divertido y dinamiza la reunión.

Conclusion.

Antes que nada, lo primero que aclaran los creadores de Scrum es que no se debe seguir el método al pie de la letra, sino que se debe adaptar a las necesidades de cada empresa. No es lo mismo una empresa de 1000 empleados distribuidos en varios continentes, que una empresa de 5 desarrolladores que estan intentando despegar en el negocio de sistemas. Pero tambien aclaran que, para que Scrum funcione, deben realizarse todos los pasos de manera apropiada. Son especialmente importantes las reuniones y las interacciones entre las personas, asi como tambien la responsabilidad de cada integrante en el proyecto.

Sobre el autor.

Gonzalo Lujan Villarreal es Analista de Sistemas y Licenciado en Sistemas, recibido en la Universidad Nacional de La Plata, Argentina. Actualmente trabaja en los proyectos PrEBi (<http://www.prebi.unlp.edu.ar>) y SeDiCi (<http://sedici.unlp.edu.ar>) de la UNLP en tareas de investigación y desarrollo de sistemas, fuertemetne relacionado a las areas de bibliotecas digitales y de procesamiento de imagenes digitales. Asimismo, realiza tareas de coordinación del grupo informatico en ambos proyectos junto con asistencia y soporte a usuarios del exterior.

gonetil@sedici.unlp.edu.ar