

SCRUM: Experiencia de Aplicación en una Empresa de Desarrollo de Software del NEA.

Walter G. Barrios¹, María V. Godoy¹,
Mirta G. Fernández¹ y Sonia I. Mariño^{1,2}
Fernando Martin Ferreira³ y César Tomás Zarrabeitia³

¹Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura. 9 de Julio 1449. 3400. Corrientes– Argentina

²Facultad de Humanidades. Av. Las Heras 727. 3500 Resistencia. Chaco.

³Facultad de Ciencias Económicas. Av. Las Heras 727. 3500 Resistencia. Chaco. Universidad Nacional del Nordeste.

mvgodoy@exa.unne.edu.ar, simarinio@yahoo.com, waltergbarrios@yahoo.com.ar,
mirtagf@hotmail.com

Resumen.

Con objeto de gestionar proyectos de una manera eficiente han surgido las metodologías ágiles para el desarrollo de software como herramientas que permiten mejorar los procesos productivos. En este trabajo se analiza la adaptación e implementación de la metodología SCRUM en una empresa de desarrollo de software del NEA (Nordeste Argentino) utilizada bajo un enfoque de administración estratégica y rediseñada para su aplicación en una microempresa. El desafío consistió en lograr una efectiva vinculación tecnológica (entre management y sistemas) para la innovación en la simplificación de los roles y agilización en la ejecución de la metodología. Se realiza una introducción teórica al tema para luego exponer los aspectos prácticos del caso y analizar los resultados.

Palabras Claves: SCRUM, Caso de aplicación, vinculación universidad-empresa, trabajo interdisciplinario.

1. Introducción

1.1 Scrum

La Ingeniería del Software [1] comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después que se utiliza. Dentro de esta área una metodología [14] permite determinar las tareas a llevar a cabo con miras a la mejora del esfuerzo realizado por el equipo de recursos humanos involucrados.

El término SCRUM es una estrategia que originalmente proviene del deporte rugby, y se entiende como volver a poner en juego un balón perdido, y en la que todo el equipo coopera y decide rápidamente la siguiente acción.

Como metodología ágil específicamente referida a la IS, en 1993, Jeff Sutherland aplicó el modelo SCRUM al desarrollo de software en Easel Corporation (Empresa que en los macro juegos de compras y fusiones se integraría en VMARK, luego en

Informix y finalmente en Ascential Software Corporation). En 1996 Sutherland presentó, junto con Ken Schwaber, las prácticas que empleaba como proceso formal, para gestión del desarrollo de software en OOPSLA 96 [6]. Las prácticas diseñadas por Schwaber y Sutherland para gestionar el desarrollo de software están incluidas en la lista de modelos ágiles de Agile Alliance [2].

En [3] se mencionan algunas implementaciones de SCRUM. Takeuchi y Nonaka [4] observaron por primera vez diversas variantes del enfoque de dicha metodología para el desarrollo de nuevos productos con equipos pequeños de alto rendimiento en empresas como Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox y Hewlett-Packard. Un enfoque similar aplicado al desarrollo de software en Borland lo indicó Coplien [5]. Además Sutherland [4] utilizó un enfoque refinado de este proceso al desarrollo en Smalltalk y Schwaber [6] a la producción en Delphi.

SCRUM es hoy, utilizado por empresas grandes y pequeñas, incluyendo Yahoo, Microsoft, Google, Lockheed Martin, Motorola, SAP, Cisco, GE, CapitalOne y la Reserva Federal de los EE.UU [7].

Su fundamento es la motivación del equipo del trabajo y una ágil reacción al cambio orientada al resultado final. Aquellos proyectos que mudan rápidamente de requerimientos se presentan como los más apropiados para su aplicación. Sus principales características se pueden resumir en [8]:

- El desarrollo de software se realiza mediante iteraciones, denominadas Sprints, con una duración de 30 días. El resultado de cada uno de ellos es un incremento ejecutable que se muestra al cliente.
- Reuniones a lo largo del proyecto, es destacable la breve reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Las prácticas empleadas por SCRUM para mantener un control ágil en el proyecto son: i) revisión de las iteraciones, ii) desarrollo incremental, iii) desarrollo evolutivo, iv) auto-organización del equipo y v) colaboración.

Los roles, artefactos y eventos principales se resumen en la Figura 1.

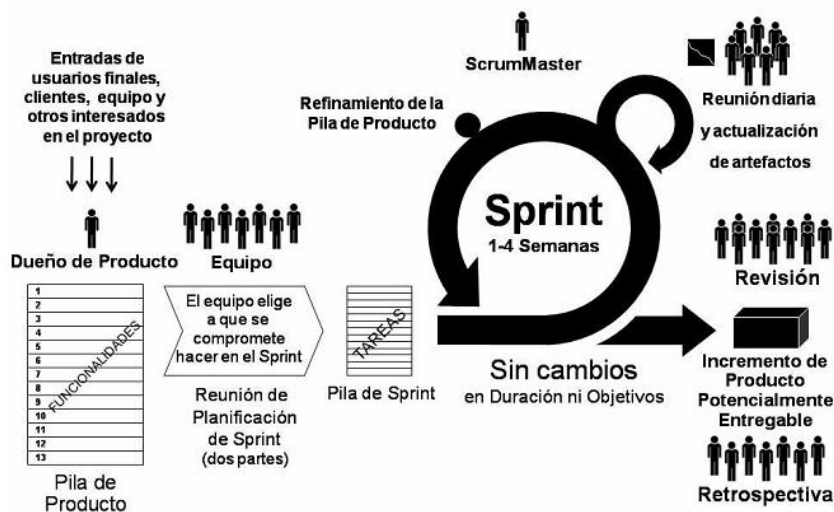


Figura 1. Roles, artefactos y eventos principales de SCRUM [2].

1.2 Sprint

La fase de Sprint es aquella en la cual el desarrollo de software se lleva a cabo. Es una lista de requerimientos seleccionados para ser implementados en la próxima iteración y seleccionados por el equipo de trabajo, en conjunto con el SCRUM Master y el propietario del producto en la reunión de planificación del Sprint.

1.3 Roles

Existen tres roles principales: Propietario del producto (Product Owner), el equipo (developers y testers) y el SCRUM Master.

Propietario del producto (Product Owner): Responsable de identificar las funcionalidades, definiendo una lista priorizada de las mismas, decidiendo cuales deberían ir al principio para el siguiente Sprint, repriorizando y refinándolas continuamente. En algunas ocasiones el dueño del producto y el cliente suelen ser la misma persona.

El equipo: Se encarga de la construcción del producto que usará el cliente, es “auto-organizado” presenta un alto grado de autonomía y responsabilidad, no hay un jefe de proyecto. El equipo decide sus compromisos y como hacer lo mejor para cumplir con lo pautado. Generalmente está compuesto por siete personas.

El SCRUM Master: Encargado del funcionamiento de SCRUM en el proyecto. Atiende los aspectos que la organización necesite según el conocimiento, experiencia con el modelo o aquellos que no son cubiertos por otras personas con la formación e idoneidad apropiada.

1.4 Las reuniones en SCRUM

SCRUM realiza el seguimiento y la gestión del proyecto a través de tres reuniones que forman parte del modelo:

- **Planificación del Sprint :** Toma como base las prioridades y necesidades de negocio del cliente, y se determina cuáles y cómo serán las funcionalidades que incorporará el producto tras el siguiente Sprint.
- **Seguimiento del Sprint :** Diaria breve, de no más de 15 minutos, en la que cada miembro del equipo explicita las tareas en las que está trabajando, si se ha encontrado o prevé encontrarse con algún impedimento, y actualiza sobre la pila del Sprint las terminadas, o los tiempos de trabajo restantes.
- **Revisión del Sprint:** Realizada al final del Sprint, con una duración máxima de 4 horas, y donde el equipo presenta al propietario del producto (Product Owner), clientes, usuarios, gestores, el entregable construido en el Sprint.

1.5 Documentos en SCRUM

Entre los documentos generados se mencionan [9]:

- **Product Backlog (pila del producto).** Es un documento de alto nivel para el proyecto, en constante evolución. Contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables, etc. priorizadas según su valor

para el negocio (Business Value). Contiene estimaciones, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido. Un término muy utilizado para designar a los requerimientos o funcionalidades del Product Backlog es “User Story”.

- **Sprint Backlog (pila del Sprint).** Describe en forma detallada cómo el equipo implementará los requerimientos durante el siguiente Sprint. Las tareas se dividen en horas, con un límite de 16 horas. Si esto sucede, se deben particionar con mayor detalle. Las tareas en el Sprint Backlog no son asignadas, son tomadas por los miembros del equipo del modo como consideren oportuno.
- **Burndown.** Es una gráfica pública que mide la cantidad de requerimientos en el Backlog del proyecto, pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, se puede 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 requerimientos 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 requerimientos pendientes en el Backlog). Si durante el proceso se añaden nuevos requerimientos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos la pendiente variará o incluso valdrá cero en algunos tramos.

1.6 Fases de SCRUM

La metodología propone las siguientes tres fases [10]:

1. **Fase de Planeamiento.**
 - **Planeación:** Se define el equipo, herramientas, el sistema de desarrollo y se crea el Product Backlog con la lista de requerimientos conocidos junto con sus prioridades y se estima el esfuerzo necesario para llevarlo a cabo (Sprint Backlog).
 - **Diseño Arquitectónico:** Se define la arquitectura del producto que permita implementar los requerimientos.
2. **Fase de Desarrollo:** Es la parte ágil, donde el sistema se *desarrolla en Sprints*. Cada Sprint incluye las fases tradicionales del desarrollo de software: relevamiento de requerimientos, análisis, diseño, implementación y entrega.
3. **Fase de Finalización:** Incluye integración, testing y documentación. Indica la implementación de todos los requerimientos, quedando el Product Backlog vacío.

2. Experiencia

2.1 Origen de la empresa

La empresa NoNameSoft se encuentra radicada físicamente en la capital de la provincia de Corrientes, brindando servicios de desarrollo de software a la región NEA del país especialmente. Surge, mediante la integración de profesionales y/o

estudiantes avanzados de la Licenciaturas en Sistemas (UNNE¹) e Ingeniería en Sistemas de Información (UTN²- FRR), para afrontar en conjunto una primera y gran oportunidad de trabajo, desarrollar un sistema integral de software, que incluye administración de venta, gestión de clientes, inventario, contabilidad, logística entre otras, para una cadena de muebles de la provincia de Formosa. La pequeña empresa se constituyo originalmente por 3 (tres) socios fundadores, quienes estimaron que el proyecto solicitado por el contratista, insumiría una primera etapa de desarrollo oscilante entre 12 y 15 meses, seguida de un período de prueba de 6 meses y del mantenimiento del sistema correspondiente.

2.2 Situación Estructural de la empresa

Actualmente, la empresa se encuentra en la etapa del emprendedor creativo en sus inicios, donde la primera crisis se presenta por la aparición de un gran volumen de operaciones que escapan al control sin una herramienta formal.

La composición actual de la Pymes³ es interdisciplinaria, conformada por 2 (dos) Programadores Universitarios de Aplicación, 2 (dos) Licenciados en Administración y 3 (tres) Ingenieros en Sistemas de Información, quienes a través de un procedimiento rudimentario pretendían controlar sus operaciones; sin embargo al surgir oportunidades y negocios, y para ser estos aprovechados eficazmente se hace necesario contar con el apoyo de una estructura organizacional acorde a las características requeridas.

2.3 Origen del Problema

Luego de un diagnóstico organizacional, llevado a cabo por los profesionales de Administración, se pudo identificar los siguientes problemas y riesgos asociados a estos [11]:

- **Estructura organizacional deficiente:** El riesgo surge de la falta de experiencia y/o conocimiento de administración y/o management de los fundadores de empresas de software, provocando que estas tiendan a quedar estancadas en un cierto nivel de desarrollo y crecimiento donde la estructura organizacional no puede soportar creciente volúmenes de operación.
- **Gestión de proyectos ineficiente:** Está vinculado con el área de producción de software específicamente, donde el desarrollo es a medida. En este proceso se incurre en costos de producción que tienen una alta incidencia en los costos totales; principalmente horas de desarrollo (horas-hombre), que si no son gestionadas correctamente pueden generar una importante pérdida para la empresa. Estas representan en la mayoría de los proyectos el 90% de los costos totales. Ligado muy estrechamente a este problema se encuentra el proceso de presupuestación, donde es vital contar con una estimación que se acerque de manera lo mas ajustada posible a la realidad. Existe además, un

¹ UNNE: Universidad Nacional del Nordeste – Facultad de Ciencias Exactas y Naturales y Agrimensura

² UTN: Universidad Tecnológica Nacional – Facultad Regional Resistencia

³ PyMES: Pequeñas y Medianas Empresas

factor esencial en la gestión de proyectos, el trabajo en equipo; encontrar la manera óptima de tramitarlo en forma flexible y de acuerdo a la capacidad instalada que será clave para el éxito.

2.4 Motivos de elección de SCRUM

Por lo expuesto, se definió como Objetivo: Fortalecer la gestión operativa de la empresa, utilizando una herramienta formal, que se adecue al perfil empresarial y que además incorpore una visión más ágil y flexible con lineamientos y principios del management para la innovación en su aplicación.

A continuación se mencionan las principales ventajas que fundamentan la elección de esta metodología respecto a otras existentes:

- **Es simple:** Es fácil trasladar el conocimiento a otros, que lo comprendan y por tanto lo puedan poner en práctica. Se necesita un SCRUM Master que entienda su papel y esté dispuesto a realizarlo con dedicación y un equipo que se autogestione.
- **Hace hincapié en la visibilidad:** Es un factor vital para SCRUM que los progresos realizados sean visibles, es decir que los entregables sean funcionales y demostrables. Existe un momento claro en que esa característica se lleva a su máxima expresión, durante el Sprint Review, donde el equipo presenta los avances realizados durante el último Sprint. Además el cliente puede decidir poner en producción lo ya desarrollado, y así determinar el inicio de la recuperación de su inversión.
- **Usa roles y normas claras:** Todas las actividades y flujos de trabajo están definidos explícitamente usando normas claras. El rol SCRUM Master, hace posible y controla que se cumplan las mismas.
- **Basado en el compromiso personal:** El equipo se compromete a implementar una determinada funcionalidad en un cierto tiempo (generalmente los 15 días de duración de un Sprint). No tiene potestad sobre el 'Qué', solo sobre el 'Cuánto' (cuanto podemos hacer en un Sprint) y el 'Cómo' (responsable de la solución técnica).
- **Soluciona los problemas día a día:** El Daily SCRUM Meeting (las reuniones diarias de SCRUM) sirven para, de una manera efectiva (no puede durar más de 15 minutos), detectar los problemas, realizar un seguimiento diario y planificar el corto plazo. Permitiendo concretar y optimizar el uso del tiempo.
- **Permite realizar mediciones ágiles:** Estas facilitan la retroalimentación. Por ejemplo puede medirse la velocidad de desarrollo del equipo; es posible comparar el tiempo de avance entre diferentes Sprints y proyectos que luego facilitan y agregan precisión al proceso de estimación de tiempos y presupuestación.

2.5 Implementación de SCRUM

Luego del trabajo de diagnóstico organizacional llevado a cabo por los especialistas en Administración, se inició el proceso de implementación de SCRUM que consistió en tres etapas:

1. Acercamiento a los conceptos y principios de las metodologías ágiles y SCRUM: En primera instancia el equipo de desarrolladores propuso esta metodología como la más adecuada para fortalecer la gestión de proyectos de la empresa. La propuesta fue verificada con el informe de diagnóstico de los especialistas en administración. Se preparó una reunión de capacitación en la metodología SCRUM.
Se expusieron los conceptos y principios más importantes del Movimiento Ágil en general y la base teórica de SCRUM, para luego presentar y discutir los conceptos de aplicación. Concluida esta etapa todos los miembros de la empresa tuvieron un claro conocimiento de la metodología a adoptar.
2. Aplicación de SCRUM a un proyecto vigente: En esta etapa el equipo de administradores debió diseñar y asumir el rol de facilitadores de SCRUM. Este llevó adelante el objetivo de guiar a los desarrolladores en la aplicación, es decir, en volcar el contenido conceptual aprendido en un proyecto vigente de desarrollo en la empresa, logrando la relación teoría-práctica real.
3. La innovación en esta etapa estuvo asociada al hecho de que el rol de SCRUM Master no fue formalmente asumido por ningún desarrollador, la idea fue la de fomentar la autogestión del equipo y que dicho rol sea rotativo entre los miembros de un Sprint a otro.
4. Seguimiento del proceso de aplicación de SCRUM en la empresa: Fue soportado por el equipo de administradores en su rol de facilitadores. Se implementó un servicio web de almacenamiento común para realizar el seguimiento de la documentación del proyecto (planillas de seguimiento de horas de trabajo, Burndown, etc.).

2.6 El proyecto

Para la realización del sistema integral de software de gestión para una cadena de mueblerías ubicada en la provincia de Formosa, se aplicaron los conceptos arriba expuestos:

Los roles asignados a los distintos participantes en el proyecto, son:

- El Dueño del Producto (Product Owner): Una de las particularidades del caso es que el Product Owner se encuentra fuera de la provincia de Corrientes. Esto determinó que la comunicación entre el equipo y él se realizase a través de correo electrónico, mensajería instantánea, skype, etc., apoyado con documentación post-planning y los reviews online.
- El SCRUM Master (posición rotativa): la aplicación estuvo más enfocada hacia la autogestión del equipo que en la preponderancia del SCRUM master.
- Equipo facilitador (administradores): se encargaron no solo de fomentar la aplicación de SCRUM sino que además incorporaron una visión holista de la gestión del desarrollo del software.
- Los miembros del Equipo de desarrollo, conformado por los 4 (cuatro) analista-programadores: asumieron un rol activo en todo el proceso de aplicación de SCRUM.

2.6.1 Backlog inicial del producto

Se desarrolló el Product Backlog a partir de entrevistas con el Product Owner e interesados claves en el proyecto, el objetivo principal fue contar con toda la información necesaria acerca de la idea del negocio (Business Case). Luego se confeccionó la lista de requerimientos o funcionalidades priorizadas, estas se obtuvieron del análisis de las necesidades inmediatas del negocio.

Esta etapa estuvo caracterizada por la dificultad de conciliar objetivos, al manejar lenguajes distintos, desde los técnicos de la empresa y el Product Owner. Previamente a la reunión de planificación del primer sprint del proyecto, el equipo de trabajo confeccionó una lista de requerimientos en base a las necesidades planteadas por el Product Owner.

2.6.2 Velocidad inicial del equipo

Para la estimación de velocidad de trabajo del equipo, primero se determinó cuántas horas disponibles poseía cada uno de los miembros durante los 15 días de duración del Sprint [12]. Se obtuvo un total de 101,5 hs semanales, considerando que el 100% de las horas disponibles no son trabajadas en su totalidad. Se decidió aplicar un factor de dedicación (porcentaje de trabajo efectivo a realizar). Su valor se determinó en un 40%, en base a experiencias de trabajo recopiladas en publicaciones actuales [12], es decir se estableció un tiempo de trabajo efectivo de 81,2 hs para los 15 días de duración del Sprint. Así el equipo se comprometió a desarrollar sus tareas de una manera más acorde a su capacidad real.

2.6.3 Planificación del Sprint

Es la reunión más importante en la aplicación de SCRUM; es vital dedicarle el tiempo necesario. A partir de lo aquí convenido se definieron plazos, estimaciones y entregables. Tal importancia radica en el hecho de que es el punto de partida del proceso de desarrollo del software. A continuación se exponen los resultados:

2.6.3.1 Duración del Sprint

Se acordó utilizar Sprints de 15 días hábiles durante todo el proyecto, se definió como oportuno dicho lapso debido que se propuso generar un contacto continuo con el cliente y así responder oportunamente a los cambios solicitados y a los posibles problemas que surjan.

2.6.3.2 Desglose de las Funcionalidades del Product Backlog

El Product Backlog cuyas funcionalidades fueron previamente priorizadas en acuerdo con el Product Owner, se desglosó, en base a la experiencia técnica del equipo, en **Tareas concretas y manejables**. El objetivo fue determinar tareas simples que permitieran una asignación práctica más conveniente y que al final del Sprint, se cuente con un entregable funcional y visible para el cliente.

2.6.3.3 Estimación de las Tareas: Utilización del Poker Planning

Luego de desglosar las Funcionalidades en Tareas, se contó con un panorama general del volumen de las mismas. Se especificó:

- Nombre de la Tarea.
- Orden de trabajo.
- Estimación en tiempo.
- Comentarios técnicos para su ejecución.

Para llevar a cabo la estimación del tiempo de las tareas se utilizó la técnica Poker Planning: cada miembro del equipo contó con una baraja de 13 cartas (Figura 2). Cada vez que debió estimarse una Tarea, se seleccionó una carta que representase su estimación de tiempo (en horas de desarrollo) y se colocó con la numeración para abajo, en la mesa. Una vez que todos los miembros del equipo seleccionaron sus cartas, estas se dieron vuelta al unísono. Así se obligó a cada integrante a pensar por sí mismo en lugar de seguir la estimación de otro. Luego de un debate de opiniones técnicas se acordaron los tiempos específicos para cada tarea. Las estimaciones fueron registradas en una planilla de cálculo, del tipo desplegado en la Figura 3. El objetivo de esta planificación fue:

- Involucrar a cada uno de los miembros con cada tarea.
- Obtener una visión propia de cada integrante sin que esta se vea afectada por la del experto.
- Resolver conflictos previamente al desarrollo. Explorando posibles inconvenientes y anticipando soluciones.
- Discutir aspectos técnicos de cada tarea.
- Compartir experiencias.

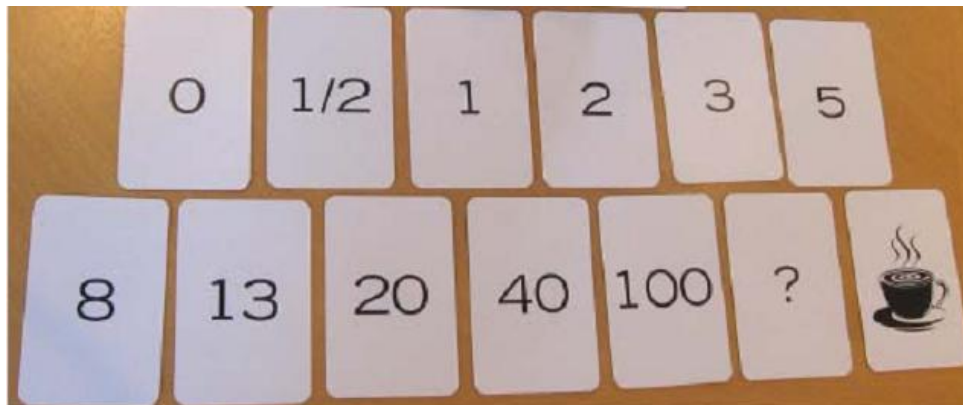


Figura 2. Tarjetas del Poker Planning

	tiempo previsto
Carga de stock de productos	26
Cargar BD	4
Diseño de la pantalla	5
Consulta Traer pedido correspondiente	1
Imprimir Etiquetas	4
Programación	12
Etiquetas	30
Investigación de la Teconología	15
Diseño	4
Revisar BD	1
Programación	7
Generar Código de Barra	4
Imprimir Etiquetas	4
Carga de factura compras	20
Obtener pedidos cargados	2
Revisar BD	1
Diseño	5
Programación	12
No Planificadas	
Pantalla Alta Pedidos	8
Unificar Cooperator	5
Modificar Consultas	12

Figura 3. Planilla de tareas planificadas

2.6.3.4 Construcción de los Sprints

Se asignaron las funcionalidades priorizadas y sus correspondientes tareas (estimadas en horas de desarrollo) al primer Sprint. Para esto, se consideró, la velocidad de desarrollo del equipo y la duración determinada del Sprint.

2.6.3.5 Creación del tablero del Sprint

Se contó, entonces, con la información necesaria para armar el tablero para el Sprint, compuesto por 4 secciones. La primera consiste en las tareas pendientes, la segunda contiene las asignadas, la tercera muestra las terminadas y la última representa las no planificadas necesarias para el cumplimiento del Sprint. En el mismo se vuelcan todas las Funcionalidades, sus respectivas Tareas y el gráfico de Burndown.

Para los listados y comentarios se utilizaron pequeñas hojas de papel autoadhesivo de varias dimensiones, formas y colores (Post It), que por sus formatos permiten una utilización dinámica y ágil.

2.6.3.6 Trabajo diario

Al comenzar el día, los desarrolladores seleccionaron las tareas a realizar, tomándolas libremente, por orden de importancia, del grupo de tareas Pendientes. Cada uno eligió aquella con la que se sintió más cómodo, escribiendo su nombre y pasándola a la columna de tareas completadas a medida que se terminaban. En cada reunión diaria se actualizaba la información del tablero.

2.6.3.7 Finalización del Sprint

Al momento de finalizar cada Sprint se contó con un modulo del producto final entregable. Además se desarrolló reunión de retrospectiva. Allí se discutieron los resultados Sprint, analizando el grafico Burndown (Figura 4), determinando las modificaciones a realizar en el próximo, de modo de incrementar la productividad, corrigiendo inexactitudes y proponiendo mejoras.

El mayor problema detectado estuvo vinculado a la subestimación en horas de trabajo de las tareas desglosadas en los primeros sprint, testeo, modificaciones en base de datos, corrección y demostración de cada funcionalidad. Esto derivó de la falta de experiencia en estimación del equipo y se ocasionaron retrasos en la finalización de las tareas.

Además se detectaron inconvenientes a la hora de solucionar problemas derivados de:

- Falta de información de la lógica del negocio a la hora de encarar un problema.
- Falta de comunicación intra-equipo.

Este problema se controló con la implementación de la unidad virtual de almacenamiento de información, que se configuró como un espacio común de intercambio y actualización, mas precisamente la herramienta que se utilizó fue Dropbox, en su versión gratuita, que es un servicio de alojamiento de archivos multiplataforma en la nube [13]. El servicio permite a los usuarios almacenar y sincronizar archivos en línea y entre computadoras y compartir archivos y carpetas con otros. Una vez instalado el programa cliente de Dropbox, fue posible dejar cualquier archivo en una carpeta designada. Ese archivo es sincronizado en la nube y en todas las demás computadoras del cliente de Dropbox.

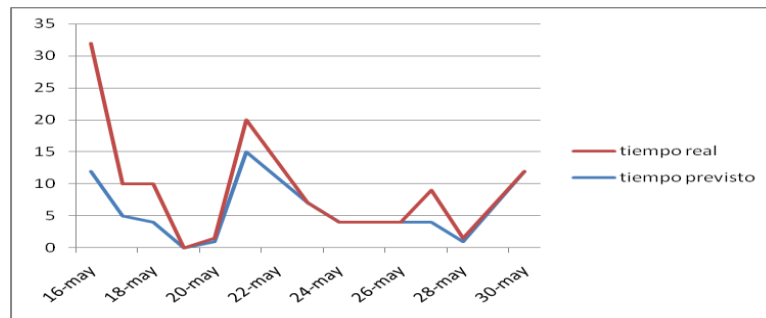


Figura 4. Gráfico Burndown

3 Conclusiones

La vinculación inter-disciplinaria de Licenciados e Ingenieros en Sistemas y Licenciados en Administración permitió encarar la aplicación de SCRUM no sólo como una práctica aislada dentro de la Pymes y para el desarrollo de software, sino como una acción dentro del plan estratégico de la misma. El objetivo principal fue iniciar un proceso de formalización de la gestión de proyectos y crear una estructura organizativa dinámica que predisponga a la empresa para futuras etapas de crecimiento.

Se considera que la experiencia de utilizar SCRUM ha sido muy positiva. Se realizó una aplicación completa y adaptada de todas las prácticas, involucrando de manera integral a los actores de la organización que aportaron al proyecto. Esto generó un crecimiento en la experiencia técnica del equipo.

Los errores de planificación del primer Sprint constituyeron un feedback importante para volver a planificar los siguientes. Las primeras etapas han significado un incremento en la eficiencia productiva, demostrando que es posible aplicar una metodología que agilice la gestión y que optimice los costos, además de generar software con requerimientos dinámicos.

Una de las derivaciones más evidentes de la aplicación de esta práctica fue el notable involucramiento de las personas con el proyecto y con los objetivos de la empresa; exteriorizado a través del alto grado de compromiso y participación de cada miembro en las etapas de planificación, diseño y ejecución.

El proceso de adaptar esta práctica a la estructura y cultura de la empresa fue además una medida acertada; que permitirá en el mediano plazo adoptar totalmente SCRUM, incorporándolo a las prácticas de la empresa de forma natural.

Referencias

- [1] Sommerville, I.: Ingeniería del Software, Séptima Edición, (2006).
- [2] Palacio, J. y Ruata, C.: “*Scrum Manager Proyectos*”. En http://www.scrummanager.net/files/sm_proyecto.pdf. (2010).
- [3] Sutherland J.: ScrumWeb Home Page: A Guide to the SCRUM Development Process Jeff Sutherland's Object Technology Web Page, (1996)
- [4] Takeuchi H. y Nonaka I.: “The New Product Development Game.” Harvard Business Review, (1986).
- [5] Coplien, J.: “*Borland Software Craftsmanship: A New Look at Process, Quality and Productivity*”. Proceedings of the 5th Annual Borland International Conference, Orlando, Florida (1994).
- [6] Schwaber K.: “Controlled Chaos: Living on the Edge.” American Programmer, (1996).
- [7] Deemer P., Benefield G., Larman G. y Vodde B.: The Scrum Primer. Scrum Training Institute. <http://assets.scrumtraininginstitute.com/downloads/1/scrumprimer121.pdf>. (2010).
- [8] Canós J., Letelier P. y Penadés M.: “*Metodologías Ágiles en el Desarrollo de Software*”. Universidad Politécnica de Valencia. En: <http://www.willydev.net/descargas/prev/TodoAgil.pdf>. (2010).
- [9] Palacio J.: Flexibilidad con Scrum, (2008).
- [10] Mendes Calo, K., Estevez, E. y Fillottrani, P.: “*Un Framework para Evaluación de Metodologías*”. <http://www.egov.iist.unu.edu/cegov/content/download/1878/47500/version/8/file/Un+Framework+para+Evaluacion+de+Metodologias+Agiles.pdf>. (2011).
- [11] Chiavenato H.: Introducción a la Teoría General de la Administración, (2006).
- [12] Henrik K., Traducción Proyectapolis: Scrum y XP desde las Trincheras, Cómo hacemos Scrum, (2007)
- [13] <http://es.wikipedia.org/wiki/Dropbox>, consulta 16/07/2011.
- [14] Oliveros, A.: Curso Administración de Proyectos de Software. Maestría en Ingeniería del Software. Universidad de La Plata (2007).