

# Seguimiento ágil de proyectos de desarrollo de software utilizando Gráficas Burn Down

María Isabel Marante, Maria Company y Patricio Letelier

Depto. de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia  
Camino de Vera s/n,  
46022 Valencia, España  
{mmarante, mcompany, letelier}@dsic.upv.es

**Abstract.** Las Gráficas Burn Down han sido impulsadas por la metodología ágil Scrum, en la cual son el artefacto protagonista para el seguimiento de cada iteración. Una Gráfica Burn Down muestra la evolución diaria del esfuerzo restante para finalizar el trabajo comprometido con el cliente en cada iteración. Aunque esta gráfica parece a priori muy sencilla, para que su uso sea efectivo requiere superar los siguientes desafíos: estimar el trabajo y diariamente recolectar y consolidar dichas estimaciones. Adicionalmente, para poder interpretar correctamente una Gráfica Burn Down, es necesario contar con información complementaria respecto de eventos diarios tales como trabajo añadido o quitado de la iteración, estimaciones faltantes, ajustes realizados a la estimación, etc. Las carencias en los aspectos mencionados están provocando que las Gráficas Burn Down no se estén aplicando o no estén resultando tan efectivas como se esperaba. En este trabajo se ilustra cómo hemos integrado las Gráficas Burn Down en la metodología y herramienta TUNE-UP, y cómo hemos superado los obstáculos antes indicados.

**Keywords:** Gráficas Burn Down, planificación y seguimiento de proyectos software, metodologías ágiles.

## 1 Introducción

Gran parte del éxito de un proyecto de ingeniería depende de su adecuada planificación y seguimiento. Por una parte se consigue prever la carga de trabajo y establecer la asignación de recursos, pudiendo así definir un presupuesto y establecer los plazos de entrega. Por otra parte, el seguimiento y control del proyecto permite la detección de desviaciones y la toma de decisiones oportunas. La planificación del proyecto es una práctica esencial y muy protagonista en guías y estándares tales como PMBOK [11], CMMI [3] y SPICE [6]. La Fig. 1 muestra un diagrama Gantt con el resumen de un plan de desarrollo de software, el cual delata la utilización de un modelo de proceso de desarrollo secuencial o en cascada.

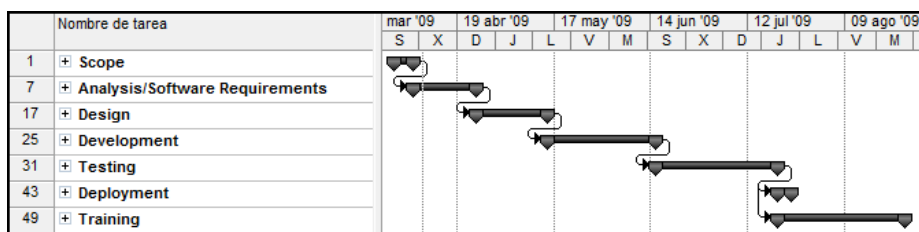


Fig. 1. Un plan asociado a un modelo de proceso en cascada

Todas las metodologías modernas para desarrollo de software proponen de forma exclusiva, o al menos preferente, un modelo de proceso iterativo e incremental. Este modelo es el más conveniente en la mayoría de las situaciones, y esencialmente cuando se prevén condiciones cambiantes ocasionadas por requisitos volátiles.

Por una parte, tanto en RUP [7] (la referencia más popular en el ámbito de metodologías de estilo tradicional) como en todas las metodologías ágiles, por ejemplo Extreme Programming (XP) [1] o Scrum [14], se utiliza un modelo de proceso iterativo e incremental. En particular, en las metodologías ágiles se enfatiza que las iteraciones sean muy cortas (no más de cuatro semanas). Sin embargo, los proyectos de desarrollo de software suelen presentar un plan que refleja un modelo de proceso secuencial o en cascada. ¿Cómo se explica esta aparente contradicción?, ¿realmente se aplica un modelo de proceso secuencial o en cascada, aún en contra de lo que sugieren las principales metodologías actuales? o ¿el plan es sólo para efectos contractuales y posteriormente no se utiliza como herramienta para seguimiento del proyecto?

Un plan de carácter secuencial (o en cascada) es sencillo de elaborar y fácil de entender para el cliente, lo cual es importante cuando se está negociando el proyecto. Un proceso iterativo e incremental exige definir, al menos de forma preliminar, cuáles serán las unidades de trabajo (Work Units o WUs) que formarán cada iteración y su priorización. Estas WUs podrían ser subsistemas o requisitos (*Features*, Casos de Uso, Historias de Usuario, o el artefacto que utilice la metodología)<sup>1</sup>. Así, el evitar entrar tempranamente en este nivel de detalle del sistema, podría ser un punto a favor de planes donde las WUs son actividades genéricas tales como: análisis, diseño, implementación, pruebas, etc. Estos serían argumentos comprensibles para que, en el contexto del inicio de un proyecto, se realicen este tipo de planes. Sin embargo, un proyecto no puede permitirse prescindir de un plan realista que permita realizar adecuadamente el seguimiento.

Las metodologías ágiles, y en particular Scrum, hacen especial énfasis en el seguimiento diario del proyecto. En XP se propone la *Stand Up Meeting* y su equivalente en Scrum es la *Daily Meeting*. Ambas son reuniones diarias, de corta duración, y en las que todo el equipo comenta respecto al estado de sus tareas y revisan el progreso sobre el plan de la iteración. En Scrum el seguimiento de la iteración (llamado *Sprint Backlog*) se suele apoyar en dos artefactos: un tablero Kanban (post-it en una pared o algún programa que los soporte) y una Gráfica Burn Down. El Kanban [5, 12] permite visualizar las tareas puestas en columnas que representan actividades, y estados de dichas actividades (*To Do*, *Doing* y *Done*). La Gráfica

<sup>1</sup> Obviamente la definición de dichas unidades de trabajo obliga a adquirir mayor conocimiento del sistema, con las consecuentes necesidades de tiempo/financiación. Esta inversión, que debería ser natural y razonable en todo proyecto, en muchos casos no se lleva a cabo. Paradójicamente, esto puede condenar el éxito del proyecto, ¡aún antes de empezarlo!

Burn Down permite visualizar la evolución y tendencia del esfuerzo restante para terminar el trabajo comprometido en una iteración, convirtiéndose en el artefacto protagonista en cuanto al seguimiento del proyecto. Sin embargo, existen varios desafíos que superar para poder conseguir una Gráfica Burn Down y para interpretarla correctamente. Primero, debe existir una infraestructura que facilite la recolección de información de esfuerzos invertidos y restantes en cada tarea, pues ir preguntando a cada uno de los miembros del equipo diariamente no resulta eficiente. Segundo, debe existir una disciplina continua de estimación y de ajustes de estimación por parte de los miembros del equipo. Tercero, esta información debe complementarse con datos respecto del trabajo añadido o quitado de la iteración, ajustes de estimación, estimaciones faltantes, etc., es decir, eventos que pueden incluso invalidar la interpretación del esfuerzo restante. Estos desafíos están causando que las Gráficas Burn Down no estén siendo utilizadas o que su aplicación no resulte tan efectiva como se esperaba para el seguimiento del proyecto.

TUNE-UP Software Process ([www.tuneupprocess.com](http://www.tuneupprocess.com)) [9] es una metodología y herramienta para la gestión ágil de proyectos de desarrollo y mantenimiento de software. TUNE-UP ha sido desarrollada en el grupo de Ingeniería de Software y Sistemas de Información (ISSI) de la Universidad Politécnica de Valencia (UPV). TUNE-UP ha surgido y se ha ido refinando en el contexto de diversos proyectos industriales y en el trabajo intensivo dentro de una PYME de desarrollo de software durante los últimos seis años. Además, TUNE-UP es parte central en dos asignaturas de último curso en la UPV, en las cuales también desde hace más de ocho años se ha ido experimentando y realizando mejoras. TUNE-UP se adhiere a los principios del Manifiesto para el Desarrollo Ágil de Software [8] y gracias a su herramienta permite, de manera muy sencilla, hacer una implantación metodológica de Scrum o XP. Sin embargo, también se ofrecen funcionalidades complementarias (que pueden habilitarse o no según se requiera) que permiten una escalabilidad de la metodología hacia enfoques más tradicionales, llegando a cubrir en profundidad muchas de las áreas de proceso de CMMI.

El objetivo de este trabajo es ilustrar cómo en TUNE-UP se han incorporado las Gráficas Burn Down para el seguimiento de proyectos superando los tres desafíos antes indicados.

El resto de este trabajo está organizado como se indica a continuación. En la sección 2 se explican las Gráficas Burn Down y su interpretación. En la sección 3 se presenta brevemente TUNE-UP. En la sección 4 se describe cómo se elaboran y explotan las Gráficas Burn Down en TUNE-UP. En la sección 5 se comentan los trabajos relacionados, y finalmente, en la sección 6 se presentan las conclusiones.

## **2 Gráficas Burn Down**

En la Fig. 2 se muestra una Gráfica Burn Down. La línea serpenteante muestra el esfuerzo restante día a día, el cual puede estar medido en Puntos o Días Ideales (u Horas Ideales). Scrum suele trabajar con Historias de Usuario (unidades de trabajo solicitadas por el cliente) y con Tareas (unidades de trabajo que descomponen las Historias de Usuario en términos de trabajo más detallado, definidas por el equipo de desarrollo). Un Punto es una medida relativa de esfuerzo entre Historias de Usuario (o entre Tareas). Cuando no se dispone de antecedentes o consenso respecto de esta medida, se suele sugerir el considerar que un Punto de una Historia de Usuario equivale a una Semana Ideal y un Punto de Tarea equivale a un Día Ideal. Ideal se refiere a un período de tiempo sin considerar interrupciones. La Gráfica Burn Down muestra la



estimado y del esfuerzo invertido. Finalmente, el comparar el esfuerzo invertido respecto del esfuerzo estimado supone un aprendizaje en cuanto a mejorar las estimaciones. Lo negativo de este enfoque, además de exigir mayor disciplina de registro de estimación y esfuerzo invertido, es que tiende a ser interpretado como un mecanismo de control respecto de los esfuerzos invertidos y de presión por cumplir las estimaciones. Sin embargo, con una buena contextualización de los mecanismos de estimación y registro de esfuerzos es posible superar estos inconvenientes y aprovechar el hecho de disponer de mayor información para el seguimiento de la iteración.

TUNE-UP permite utilizar uno de dichos enfoques en cada producto. Sin embargo, **en el resto del artículo sólo detallaremos el enfoque (b), por ser más completo y para no complicar la lectura yendo de un enfoque a otro.**

Además de utilizarse para el seguimiento de una iteración, las Gráficas Burn Down son también útiles para el seguimiento de una *release*, incluyendo todas las iteraciones planificadas. Una vez más y para simplificar nuestra explicación, **en el resto del artículo nos centraremos en el seguimiento de una iteración más que en el seguimiento de una *release*, aunque los conceptos y explicaciones son similares en ambos contextos.**

Respecto de la correcta interpretación de la Gráfica Burn Down, además de la diaria recolección y consolidación de datos para mantener actualizada dicha información, es necesario tener en cuenta los eventos que pueden ocurrir, y que podrían incluso invalidar cualquier interpretación de la gráfica. Por ejemplo, si existe trabajo sin estimar en un día o si la estimación se sabe que no es la acertada y no se ha actualizado, al observar un valor de, por ejemplo, 40 puntos restantes, podríamos realmente tener un valor muy por encima o por debajo del valor mostrado por la gráfica, sin darnos cuenta. Por otra parte, al querer interpretar los trazos de bajada del esfuerzo restante podríamos suponer que dichas bajadas se refieren a que se ha avanzado en el trabajo de la iteración, cuando realmente pueden deberse a que se ha quitado trabajo (eliminándolo o pasándolo a otra iteración) o que simplemente se han ajustado las estimaciones rebajándolas. Del mismo modo, un trazo de subida podría tener varias interpretaciones tales como: se ha añadido trabajo a la iteración o se ha ajustado la estimación incrementándola. “A menudo una gráfica de la iteración puede estar contando varias situaciones” [4]. El no disponer de esta información complementaria a la Gráfica Burn Down puede conducir a apreciaciones o decisiones incorrectas.

### 3 TUNE-UP

TUNE-UP es una metodología y su herramienta, ambas dirigidas a apoyar la gestión ágil de proyectos de desarrollo de software. Los conceptos básicos de TUNE-UP son:

- Iteración. Es el período de tiempo durante el cual el equipo trabaja en conseguir una nueva versión del producto<sup>2</sup>. Tal como en Scrum, en TUNE-UP se promueve que las iteraciones no sobrepasen las 4 semanas.
- Work Unit (WU). Las WUs son las unidades de trabajo que se incluyen en una iteración. Usamos este concepto para englobar cualquier tipo de cambio en el producto (nuevo

---

<sup>2</sup> Una versión del producto es el resultado del trabajo de una determinada iteración. En el resto del artículo usaremos indistintamente ambos términos.

requisito, mejora de requisito existente o corrección de un fallo), y también para referirnos a cualquier otra tarea que se enmarque en el trabajo de una iteración.

- Agente o miembro del equipo. Es un participante en el trabajo de la iteración (puede o no ser del equipo de desarrollo)

TUNE-UP se caracteriza fundamentalmente por combinar los siguientes elementos:

- **Modelo de proceso iterativo e incremental** para el desarrollo y mantenimiento del software.
- **Proceso de desarrollo dirigido por las pruebas (*Test-Driven*)**. La definición de una WU es básicamente la especificación de sus pruebas de aceptación acordadas con el cliente. A partir de allí, todo el proceso gira en torno a ellas. La idea que los requisitos sean especificados en detalle usando pruebas de aceptación es una innovación de TUNE-UP denominada *Test-Driven Requirement Engineering* (TDRE) [10].
- **Workflows flexibles** para la orquestación del trabajo asociado a cada unidad de trabajo. Los productos, según sus características, tienen asociados un conjunto de workflows que son utilizados para realizar cada WU. Los workflows flexibles permiten: saltar hacia adelante o hacia atrás en el workflow, realizar cambios de agentes asignados e incluso cambios de workflow, trabajar en paralelo, añadir actividades no contempladas en la definición del workflow, etc. Esta flexibilidad a su vez evita que la especificación del workflow se complique cuando se intenta añadir explícitamente todas las posibles transiciones.
- **Planificación y seguimiento continuo basado en la gestión de esfuerzos**. TUNE-UP promueve y apoya con funcionalidad específica, que en todo momento esté actualizado el estado de una iteración, de las WUs incluidas en ella, y el trabajo asignado a los miembros del equipo. El equipo puede actuar oportunamente con dicha información, tomando decisiones tales como: negociar el alcance de la iteración con el cliente, conseguir más recursos, redistribuir carga de trabajo entre los miembros del equipo, etc.

### 3.1 Seguimiento de una iteración en TUNE-UP

En TUNE-UP el seguimiento de una iteración incluye los siguientes mecanismos de apoyo:

- Panel Kanban en el cual se sintetizan todas las actividades de los workflows en las que cada miembro del equipo tiene trabajo asignado. En el Kanban se puede visualizar en qué actividades se encuentran las WUs de una iteración.
- Módulo *Version Contents and Tracking* (VCT). En este módulo se ofrecen varias vistas del contenido y estado de las WUs en una iteración. En la Fig. 3 se observa la lista de WUs en una determinada iteración. Gracias al enfoque TDRE, también se tiene un seguimiento detallado respecto del estado de aplicación de las pruebas de aceptación asociadas a la WU. En la Fig. 3 las columnas Programación y Testeo ilustran los OKs y KOs asociados al éxito o no de las pruebas de aceptación de cada WU en dichos niveles de testeo. La columna “*Current Activity*” indica las actividades (y miembros del equipo asignados) en las cuales se encuentra actualmente una WU.
- Alertas y notificaciones a los miembros del equipo. Automáticamente se generan alertas y notificaciones ante ciertos eventos, por ejemplo, cuando el esfuerzo invertido sobrepasa el esfuerzo estimado en una actividad de una WU, cuando se sobrepasa el tiempo de postergación definido para una actividad, cuando se cambia de versión una WU, etc.

- Dashboard de una iteración (ver Fig. 4). En el panel superior izquierdo está la Gráfica Burn Down (en la cual nos centraremos en el resto del artículo). En el panel inferior izquierdo se muestra día a día la cantidad de esfuerzo asociado a WUs terminadas y a WUs no terminadas. En el panel superior derecho se muestra el esfuerzo invertido en cada WU, el cual se clasifica en tiempo ordinario, tiempo dedicado a responder mensajes y tiempo asociado a la participación en reuniones. Además, en la pestaña del panel superior derecho puede consultarse el esfuerzo asociado al retrabajo en cada WU de la iteración. Finalmente en el panel inferior derecho puede consultarse la cantidad de Pruebas de Aceptación y el estado de aplicación de éstas en cada WU de la iteración.

Programación	Testeo	WU	Order	Version	WU Name	Current Activity
		424		0.0.3	Alta apuesta Campeón	Aplicar Pruebas de Aceptación / Sergio
		429		0.0.3	Modificación de apuesta Especial	Aplicar Pruebas de Aceptación / Maria
		430		0.0.3	Realizar apuesta Campeón	Aplicar Pruebas de Aceptación / Pablo
		435		0.0.3	Simulación de apuesta Campeón	Diseñar e Implementar / Miguel
		608		0.0.3	Mis eventos	Introducir / Jorge
		619		0.0.3	Perfil de usuario - Saldo en cuenta	Aplicar Pruebas de Aceptación / Pablo
		629		0.0.3	Modificación y baja de noticia	Aplicar Pruebas de Aceptación / Sergio

Fig. 3. Seguimiento respecto de actividad actual y de niveles de testeo



Fig. 4. Dashboard de una iteración

#### 4 Gráfica Burn Down de TUNE-UP

TUNE-UP ofrece una Gráfica Burn Down para cada iteración de un producto, junto con una tabla llamada Daily Events con información complementaria para la correcta interpretación de la gráfica (esta tabla se explica en detalle más adelante).

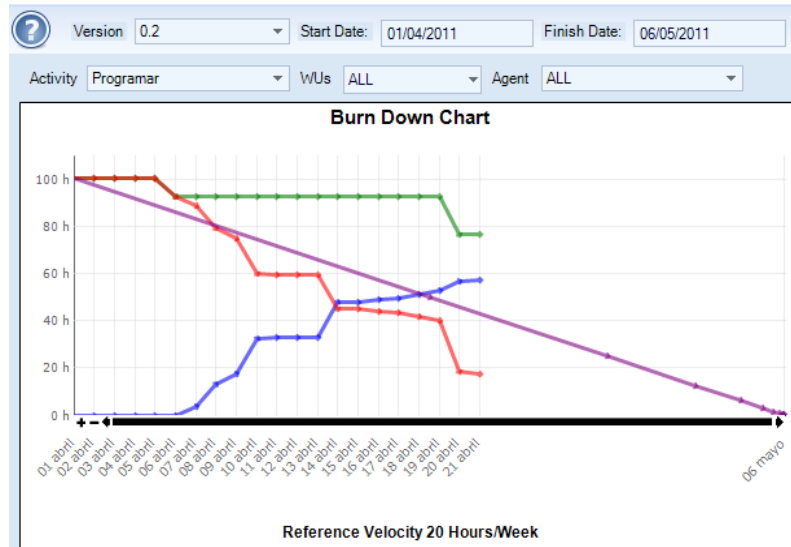


Fig. 5. Gráfica Burn Down de una iteración

La Gráfica Burn Down de TUNE-UP (ver Fig. 5) incluye dos gráficas en una; una gráfica básica Burn Down (asociada a la línea serpenteante descendente) y una gráfica Burn Up (asociada a la línea ascendente representando el esfuerzo invertido). Además, se incluye una línea que representa el esfuerzo estimado y una línea diagonal que representa el esfuerzo restante de referencia (considerando una velocidad de proyecto constante). Al disponer en una misma gráfica los esfuerzos estimados, invertidos y restantes, se facilita la interpretación del estado de la iteración y cómo se ha ido desarrollando. Situaciones tales como una bajada o subida pronunciada del esfuerzo restante podrían visualmente explicarse por una correspondiente bajada o subida en la línea de esfuerzo estimado, o bien en una subida o bajada en la línea de esfuerzo invertido. Sin embargo, la confirmación de estas interpretaciones, como veremos a continuación, exige contar con la información detallada de los eventos que pueden haber ocurrido entre dos puntos consecutivos de la gráfica. El esfuerzo restante de referencia corresponde a la línea que se traza desde el punto de mayor esfuerzo restante hacia el punto de esfuerzo restante 0 en el día de fin de la iteración. Además, asociada a esta línea se muestra en la parte inferior de la gráfica la velocidad requerida para conseguir la tendencia ilustrada por ese esfuerzo restante de referencia. Toda la información de la Gráfica Burn Down puede ser filtrada por Actividad, WU y/o miembro del equipo.



Date : 20 abril (2 items)		
Event Type : Ajuste Estimación - Decremento (3 items)		
Activity	WU	Name
Programar	31	Unirse a peña
Programar	41	Creacion galeria fotografica ruta
Programar	80	Ofrecer el máximo disponible día ocupado.
Event Type : Introducción de Estimación Faltante (2 items)		
Activity	WU	Name
Programar	85	Mostrar usuario que inserta en la foto en la galeria
Programar	86	Mostrar el numero de comentario totales en cada foto de la galeria
Date : 19 abril (1 item)		
Event Type : Actividad Sin Estimación (2 items)		
Activity	WU	Name
Programar	85	Mostrar usuario que inserta en la foto en la galeria
Programar	86	Mostrar el numero de comentario totales en cada foto de la galeria
Date : 18 abril (1 item)		
Date : 17 abril (1 item)		

**Fig. 6.** Tabla Daily Events

La Tabla Daily Events que muestra la Fig. 6, contiene los eventos diarios que permiten interpretar con precisión la Gráfica Burn Down. Haciendo clic en un punto de la gráfica de la Fig. 5, se despliega en la tabla de la Fig. 6 la lista de eventos ocurridos entre el día previo y el día seleccionado. Para cada evento se indica el miembro del equipo, la actividad e información de la WU donde se produce. En TUNE-UP se supervisan todos los eventos que pueden influir en la correcta interpretación del esfuerzo restante, dichos eventos se describen a continuación:

#### **Eventos que invalidan la lectura del esfuerzo restante**

- Actividad con estimación sobrepasada. El esfuerzo invertido por el miembro del equipo en la actividad sobrepasa el estimado (lo cual llevaría a un esfuerzo restante negativo). El miembro del equipo asignado debería re-estimar
- Actividad sin estimación. La actividad no está estimada o su valor es 0. El miembro del equipo asignado debería estimar

#### **Eventos que provocan una variación en el esfuerzo restante observado**

- Cambios del esfuerzo invertido. El esfuerzo invertido se ha modificado. Por ejemplo, se había registrado 10 horas de trabajo cuando realmente debían de ser 5 horas.
- Ajuste en Estimación. Incremento o Decremento de la estimación.
- Introducción de estimación faltante. Indica que se ha estimado una actividad que el día anterior no tenía estimación.
- Actividad asignada/desasignada a/de un miembro del equipo. Esto es sólo relevante cuando se trata de la Gráfica Burn Down filtrada con un miembro del equipo.
- WU nueva. WU creada y añadida a la iteración
- WU eliminada.
- WU desestimada. Su esfuerzo restante se considera igual a 0
- WU añadida. WU que se ha añadido a la iteración (ya existía sin iteración asignada o en otra iteración).
- WU quitada. WU que estaba el día anterior en la iteración pero se ha cambiado a otra.
- WU terminada. Su esfuerzo restante se considera igual a 0

Para ilustrar el uso de estos eventos en la interpretación de la gráfica, a continuación comentamos un ejemplo. La Fig. 5 muestra la Gráfica Burn Down correspondiente a la actividad Programación de la versión 0.2 de un determinado producto. En la gráfica, el día 20 de Abril se observa un descenso pronunciado del esfuerzo restante. Este descenso, a priori lo podemos asociar al descenso del esfuerzo estimado (observable en la línea superior de la gráfica). Pero cuestiones tales como: ¿por qué ha descendido el esfuerzo estimado?, ¿se ha movido, eliminado o desestimado trabajo?, ¿algún miembro del equipo ha ajustado alguna estimación?, no se pueden responder a simple vista. Para responder tales cuestiones es esencial la información de la Tabla Daily Events. En la Fig. 6 vemos los eventos asociados a la actividad Programación que ocurrieron el día 20 de Abril; hubo un decremento en la estimación de la actividad Programación en 3 WUs y se han introducido 2 nuevas estimaciones que faltaban (en la Fig. 6 vemos que el día 19 de Abril faltaba por estimar la actividad Programación en dichas WUs). De esta forma sabemos que el descenso del esfuerzo restante se debe a un decremento en la estimación de 3 WUs, aunque además se hayan incluido 2 nuevas estimaciones antes no consideradas.

Finalmente, el disponer de información detallada de lo realizado durante cada iteración, aporta información útil para reuniones de revisión de la iteración o reuniones de retrospectiva, pudiendo llegar a evaluar acciones de mejora en el proceso mediante la comparación de datos de diferentes iteraciones y su tendencia.

#### 4.1 Infraestructura para generar la Gráfica Burn Down de TUNE-UP

Para aprovechar la información de una Gráfica Burn Down, es esencial que el equipo adquiera una disciplina diaria de registro de esfuerzo invertido y de actualización (cuando corresponda) de las estimaciones de las actividades en las cuales participan. Para facilitar esto, TUNE-UP integra el registro de esfuerzos invertidos en el trabajo diario del equipo proporcionándole varias facilidades para que dichos registros se realicen de forma asistida en el mismo instante en el cual se está trabajando (aunque también se permite posteriormente registrar o corregir el esfuerzo invertido, es decir, como se suele hacer en otras herramientas). Por ejemplo, cuando un miembro del equipo accede a una WU estando ésta preparada para su trabajo, TUNE-UP comienza automáticamente a computar tiempo en dicha WU/Actividad. De forma similar, al dejar de trabajar en una WU activa ésta se pausa automáticamente y cuando en una WU se están computando tiempos, si se abre otra WU preparada para trabajar, se pregunta si se quiere pausar la que ya estaba activa. Este cambio en los hábitos de trabajo, en nuestra experiencia lo hemos apoyado con ciertas acciones, tales como: cada miembro del equipo dispone de dos monitores para poder tener siempre abierta la herramienta en uno de ellos, y por otra parte, el sistema ofrece un informe diario de la dedicación de cada miembro del equipo que facilita detectar registros incorrectos de esfuerzo invertido (lo cual resulta útil mientras se adquiere esta nueva disciplina de trabajo).

En TUNE-UP, los datos que se muestran en la Gráfica Burn Down se recopilan automáticamente mediante un servicio instalado en el servidor de base de datos. Diariamente genera una *snapshot* del estado de los esfuerzos invertidos, estimados y restantes de cada WU/Actividad/Miembro del equipo de las iteraciones no finalizadas. Además, opcionalmente el equipo puede lanzar manualmente la generación de la *snapshot* del día actual para disponer de la información diaria hasta ese momento. De esta forma, asociado a cada día existe una *snapshot* de la cual se obtienen los puntos diarios incluidos en la Gráfica Burn Down.

Comparando la *snapshot* del día anterior con la del día actual se determinan los eventos que se muestran en la Tabla Daily Events (aquellos eventos indicados en el apartado anterior).

## 5 Trabajos Relacionados

Hemos agrupado los trabajos relacionados en dos categorías: bibliografía asociada a metodologías ágiles, y herramientas de apoyo a la gestión ágil de proyectos software.

Las Gráficas Burn Down están presentes en mucha bibliografía sobre agilidad pero de forma muy poco detallada en cuanto a cómo elaborarlas e interpretarlas. El tema referente al seguimiento de una iteración y las estimaciones y/o registros de esfuerzo está muy presente en todos los foros de debate de metodologías ágiles, dejando patente el hecho que en la práctica cada equipo lo aplica a su manera. Además, debido a las dificultades para recolectar la información oportunamente y con una aceptable precisión, se llega incluso a prescindir de las Gráficas Burn Down.

Más allá de los conceptos básicos de una Gráfica Burn Down (indicados por ejemplo en textos de referencia para Scrum [13, 15]), respecto de los problemas en cuanto a su interpretación, pocos autores proponen mejoras. M. Cohn en [2], sugiere utilizar la Gráfica Burn Down Bar como alternativa más expresiva respecto de la Gráfica Burn Down, para ilustrar situaciones en las cuales se añade o se quita trabajo en una iteración, o se re-estima el trabajo existente en la iteración. En esta gráfica, cuando se añade trabajo durante una iteración la barra correspondiente se extiende en el cuadrante negativo del eje Y. Análogamente, cuando se quita trabajo durante la iteración, la barra correspondiente se recorta por debajo en el cuadrante positivo del eje Y. Estas dos situaciones se ilustran en la Fig. 7 (extraída de [2]). En la gráfica de la izquierda de la Fig. 7 se observa que en las iteraciones 2 y 3 se ha añadido trabajo en la iteración. En la gráfica de la derecha puede observarse que al quitar trabajo se llega incluso al caso en el cual el trabajo restante es menor que el trabajo al comienzo de la *release*. Esta representación también es sugerida por G. Dinwiddie en [4] denominándola “Gráfica Burn Down con piso variable”, aunque reconoce que es complicado de representar, especialmente cuando el alcance cambia frecuentemente.

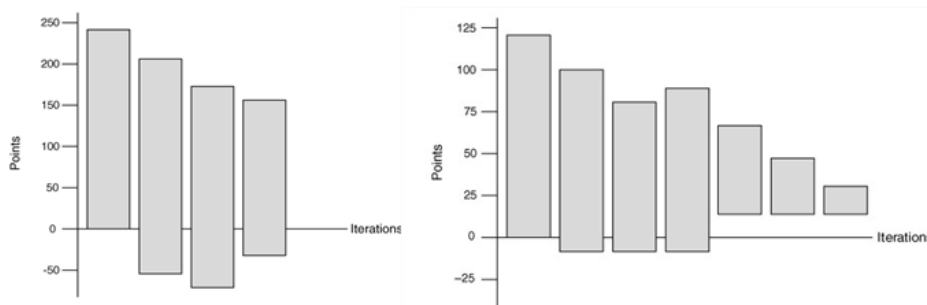
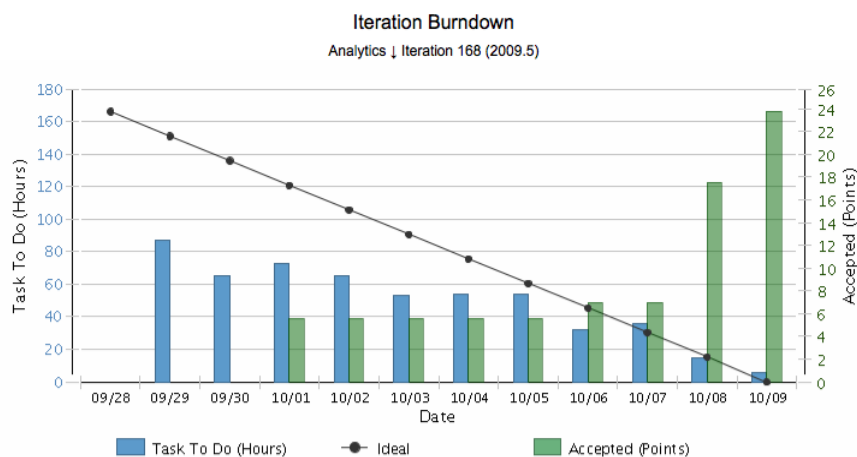


Fig. 7. Gráfica Burn Down Bar

Desde el punto de vista de las herramientas para la gestión ágil de proyectos software, hemos estudiado las Gráficas Burn Down en VersionOne ([www.versionone.com](http://www.versionone.com)), Rally

(www.rallydev.com) y JIRA (www.atlassian.com), tres de las herramientas más populares en su gama. En cuanto a las unidades de trabajo, las dos primeras trabajan con Historias de Usuario que se dividen en Tareas. En JIRA el concepto genérico utilizado es Issue, el cual también puede dividirse en Tareas. En cuanto a la unidad para medir el esfuerzo, en VersionOne para las Historias de Usuario se usan Puntos y para las Tareas se usan Horas Ideales, en Rally las unidades de esfuerzo son configurables (Puntos, Días u Horas Ideales, etc.) y en JIRA se utilizan Días u Horas Ideales. Respecto de la recolección de datos de esfuerzo, en VersionOne y en JIRA se registra diariamente el esfuerzo invertido, mientras que en Rally, existe una extensión llamada *Rally Time Tracker* que permite registrar esfuerzos invertidos en una hoja de horas trabajadas (*TimeSheet*), pero éstos no se consideran en su Gráfica Burn Down. Con respecto a la forma de representación de los esfuerzos, en VersionOne se muestran dos gráficas separadas; una Gráfica Burn Down y una Gráfica Burn Up de las tareas (en Horas). En Rally se sintetiza en una sola gráfica normalizada el Burn Down de las tareas (usando Horas) y los Puntos de Historias de Usuario completadas (ver Fig. 8). Por último, JIRA ofrece en una sola gráfica tanto el Burn Down como el Burn Up de los Issues (en Días u Horas Ideales).



**Fig. 8.** Gráfica Burn Down de una iteración en Rally

Las metodologías ágiles no promueven el registro de esfuerzos invertidos, sin embargo, como hemos visto, las herramientas VersionOne y JIRA sí que lo permiten y con ello elaboran las Gráficas Burn Up. M. Cohn en [2] sugiere no registrar el esfuerzo invertido por considerar que los riesgos y el trabajo de hacerlo, por lo general, superan los beneficios. Sin embargo, cuando el equipo no registra el esfuerzo invertido, no se cuestiona si las estimaciones fueron acertadas, y se puede invertir mucho más (o menos) esfuerzo del que estimó inicialmente, sin que esta situación sea evidente. Esto no favorece el aprendizaje y mejora de las estimaciones.

Los enfoques estudiados asumen que los miembros del equipo actualizan diariamente el esfuerzo restante para completar su trabajo, sin proveer ningún mecanismo de apoyo para esto. La actualización de estos valores se hace de forma manual lo cual conlleva el riesgo de que no estén actualizados oportunamente o los datos no sean correctos, además del trabajo que esto supone.

Por otra parte, en ninguno de los trabajos mencionados (bibliografía o herramientas) puede conocerse si lo añadido o quitado se debe a nuevo trabajo o a un trabajo ya definido y posiblemente proveniente de iteraciones previas. De forma análoga, tampoco puede distinguirse entre trabajo quitado (que se ha movido a otras iteraciones posteriores), eliminado, terminado o desestimado. En resumen, en ninguno de los trabajos relacionados que hemos estudiado existe una identificación integrada de todos los eventos que pueden influir en la interpretación de una Gráfica Burn Down.

## 6 Conclusiones

Las técnicas para realizar el seguimiento y control de un proyecto de desarrollo de software deben estar alineadas con la metodología de trabajo utilizada. Los Diagramas Gantt resultan claramente inapropiados cuando se utiliza una metodología que usa un modelo de proceso iterativo e incremental. Scrum establece como artefacto esencial la Gráfica Burn Down para realizar el seguimiento de una iteración [15], sin embargo, en la literatura de Scrum no se profundiza en los inconvenientes que se presentan al intentar aplicarlos en el marco de un proyecto.

Uno de los principios básicos del Manifiesto para el Desarrollo Ágil de Software es “valorar a los individuos y su interacción sobre el proceso y las herramientas”. Este principio muchas veces se tiende a interpretar como no usar proceso ni herramientas. Sin embargo, esto resulta contradictorio en el caso de las Gráficas Burn Down, las cuales requieren de una infraestructura de recolección, disciplina de registro e información complementaria, todo ello indispensable para sacarles partido. A primera vista se trata de una técnica muy sencilla, pero al tratar de aplicarla surgen muchos obstáculos que deben ser resueltos con apoyo de disciplina (proceso) y de herramientas. TUNE-UP provee dos niveles de detalle para la recolección y explotación de la información de la Gráfica Burn Down lo cual permite al equipo comenzar de una forma sencilla (sólo registrando esfuerzo restante) y más adelante posiblemente cambiar a otro nivel más exigente (registrando esfuerzo invertido y restante). Además, en TUNE-UP existen varias ayudas para computar de forma casi automática el esfuerzo invertido en el momento en el cual el miembro del equipo está trabajando en una actividad. También, se puede, si se desea, registrar el esfuerzo después de haber realizado una actividad (tal como lo hacen la mayoría de las herramientas para gestión ágil de proyectos) o hacer correcciones de los esfuerzos registrados.

Respecto de los planteamientos teóricos o implementaciones de Gráficas Burn Down en herramientas, nuestra propuesta se diferencia por:

- Incluir el concepto de Evento Diario (*Daily Event*). Los Eventos Diarios permiten confirmar cualquier interpretación de la gráfica, ofreciendo información detallada respecto de lo ocurrido cada día y que pueda haber influido en el esfuerzo restante para acabar la iteración. Además, utilizando filtros se puede acotar la información a nivel de cada miembro del equipo, actividad y WU.
- Contar con una infraestructura para el cálculo automático, tanto de los esfuerzos como de los Eventos Diarios, basado en la comparación de *snapshots* diarias de la base de datos.
- Ofrecer diversas ayudas para promover el registro actualizado de los esfuerzos, creando una disciplina de gestión de esfuerzos.

La información detallada de TUNE-UP y en particular de su Gráfica Burn Down se encuentra accesible en [www.tuneupprocess.com](http://www.tuneupprocess.com). Allí también es posible acceder a un *SandBox* para “jugar” con unos proyectos de ejemplo y poder consultar sus Gráficas Burn Down.

Estamos trabajando en el desarrollo de un cuadro de mandos asociado a cada producto, el cual incluya información resumida de las iteraciones y permita observar tendencias respecto de datos tales como: velocidad del equipo y WUs implementadas por tipo. También estamos evaluando alternativas para mejorar la visualización de los Eventos Diarios en el contexto de la Gráfica Burn Down.

## References

1. Beck K. Extreme Programming Explained: Embrace Change. Addison-Wesley (2000).
2. Cohn, M. Agile Estimating and Planning. Prentice Hall PTR (2005).
3. CMMI v1.2. Capability Maturity Model Integrated. SEI-Software Engineering Institute, Carnegie Mellon. 2009. [www.sei.cmu.edu/cmmi/models/](http://www.sei.cmu.edu/cmmi/models/) (Visitado Junio de 2009).
4. Dinwiddie, G. Feel The Burn, Getting the Most out of Burn Charts from Better. Software Magazine, Volume 11, Issue 5, pp. 26-31 (Julio/Agosto 2009).
5. Hiranabe K. Kanban Applied to Software Development: from Agile to Lean. <http://www.infoq.com/articles/hiranabe-lean-agile-kanban>, (2008) (Visitado Octubre 2010).
6. ISO-International Organization for Standardization. ISO/IEC 15504, SPICE: Software Process Improvement and Capability dEtermination (1998).
7. Kroll P., Kruchten P. Booch G. The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP. Addison-Wesley Professional (2003).
8. Manifesto for Agile Software Development. <http://agilemanifesto.org/> (2001) (Visitado Abril 2010).
9. Marante, M., Letelier, P., Suarez, F. TUNE-UP: Seguimiento de proyectos software dirigido por la gestión de tiempos. XIV Jornadas de Ingeniería de Software y Bases de Datos (JISBD 2009). pp. 57-68, San Sebastián (2009).
10. Marante, M., Letelier, P., Suarez, F. Gestión de requisitos basada en pruebas de aceptación: Test-Driven en su máxima expresión. XV Jornadas de Ingeniería de Software y Bases de Datos (JISBD 2010). pp. 61-72, Valencia (2010).
11. PMI-Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK Guide) - Fourth Edition (2008)
12. Quesada X. Visual Management for Agile teams. <http://vimeo.com/16918299> (2010) (Visitado Abril 2010).
13. Schwaber, K. Agile Project Management with Scrum. Microsoft Press, Washington (2004).
14. Schwaber K., Beedle M. Agile Software Development with Scrum. Prentice Hall, Series in Agile Software Development (2001).
15. Schwaber K., Sutherland J. The Scrum Guide, the official Scrum Body Of Knowledge. <http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf> (Febrero 2010) (Visitado Abril 2011).