

## Estimación y control de costos en métodos ágiles para desarrollo de *software*: un caso de estudio

### *Estimation and Control in Agile Methods for Software Development: a Case Study*

Mitre-Hernández Hugo A.

Centro de Investigación en Matemáticas (CIMAT)  
Ciencias de la Computación, Zacatecas, México  
Correo: hmitre@cimat.mx

Ortega-Martínez Edgar

Centro de Investigación en Matemáticas (CIMAT)  
Ciencias de la Computación, Zacatecas, México  
Correo: eortegam@cimat.mx

Lemus-Olalde Cuauhtémoc

Centro de Investigación en Matemáticas (CIMAT)  
Ciencias de la Computación, Zacatecas, México  
Correo: clemola@cimat.mx

Información del artículo: recibido: diciembre de 2012, reevaluado: abril de 2013, aceptado: julio 2013

#### Resumen

El desarrollo de *software* utilizando métodos ágiles está en crecimiento debido a la productividad asociada a estas metodologías, además de la flexibilidad demostrada para equipos pequeños. Sin embargo, estas metodologías cuentan con debilidades claras de estimación y gestión de costos de desarrollo, al igual que los administradores de proyectos no cuentan con las suficientes evidencias para la comprobación del gasto del presupuesto en un proyecto debido a la poca documentación generada y por la falta de seguimiento en el gasto de los recursos. Se presenta una propuesta de estimación y control de costos en métodos ágiles para resolver estas carencias. En este sentido, se realizó un caso de estudio en una empresa de desarrollo de *software* ágil utilizando la propuesta en proyectos de *software* de servicio (SaaS) y aplicaciones Web. En los resultados se encontró que la propuesta genera un alto grado de evidencias para los administradores de proyectos, pero presenta carencias en la administración de las evidencias para el control y toma de decisiones, lo cual dio origen a una definición de un proceso de toma de decisiones para ser aunado a la propuesta de medición.

#### Descriptores:

- gestión del *software*
- gestión de costos
- métodos ágiles
- desarrollo de *software* ágil
- SaaS

## Abstract

The development of software (SW) using agile methods is growing due to the productivity associated with these methodologies, in addition to the flexibility shown in small teams. However, these methods have clear weaknesses of software development in cost estimation and management, as well as the fact that project managers do not have enough evidence to verify the budget spending on a project due to the poor documentation generated and the lack of monitoring of resource spending. A proposal estimation and cost control in agile methods to solve these shortcomings. To this end, a case study was conducted in an agile software development company using the proposal for Software as a Service (SaaS) and Web application projects. The results found were that the proposal generates a high degree of evidence for project managers, but it has shortcomings in the administration of the evidence for the control and decision making, which led to a definition of a decision making process to be coupled with the measurement proposal.

### Keywords:

- software management
- cost management
- agile methods
- agile software development
- SaaS

## Introducción

Las metodologías ágiles son bien adoptadas por las pequeñas y medianas empresas (PYME) debido a que les permiten tener procesos organizados, repetibles y mejorables sin una alta inversión de presupuesto y de tiempo en su implementación. Las metodologías más utilizadas en la industria actualmente son: Extreme Programming (Jeffries *et al.*, 2000), SCRUM (Deemer *et al.*, 2010) y Feature Drive Development (Anderson, 2003).

Las metodologías ágiles toman su nombre después de que en 2001 un grupo de 17 expertos en desarrollo de *software* reunieron sus ideas y crearon el manifiesto ágil, estableciendo doce principios y en donde cada metodología debía cumplir con los siguientes preceptos: individuos e interacciones sobre procesos y herramientas, *software* funcionando sobre documentación extensiva, colaboración con el cliente sobre negociación contractual, respuesta ante el cambio sobre seguir un plan.

La administración de proyectos de *software* es una de las partes fundamentales para obtener resultados exitosos según Jones (2004), considerando un proyecto exitoso como el que termina dentro del tiempo, costo y calidad esperada (Jones, 2003; Agarwal y Rathod, 2006; Chow y Cao, 2008), principalmente preservando los doce principios del manifiesto ágil de Chow y Cao (2008). En la industria de manufactura ágil se han presentado buenos resultados de adaptabilidad a los cambios externos con menos pérdidas de costos (Laanti y Kettunen, 2006), en cuanto al desarrollo ágil de *software* esta adaptabilidad es necesaria en los cambios de los productos de *software* (SW).

En la administración de métodos ágiles la medida utilizada para evaluar el rendimiento de un equipo es conocer la velocidad con que se desarrollan los elemen-

tos del registro del producto (PBI, *Product Backlog Items*) (Yap, 2006), además el tamaño del proyecto se mide en el esfuerzo total que se necesita para su desarrollo. Conforme a lo anterior, las medidas más comunes en métodos ágiles son el esfuerzo y el tamaño, sin embargo, esto da pie a diversas pérdidas económicas por falta de medidas de costos.

El problema con los costos en los métodos ágiles es la falta de una administración y monitorización efectivas (Yap, 2006; Keaveney y Conboy, 2006; Sulaiman y Barton, 2006; Rusk, 2009), además de carecer de la generación de evidencias (Jones, 2004; Sulaiman y Barton, 2006; Rusk, 2009) y estimaciones de costos guiada por procesos mejorables (Keaveney y Conboy, 2006; Pham A. y Pham P., 2011), afectando con esto al dueño de la empresa de SW y ocasionando un descontrol con los administradores de proyectos y expertos en métodos ágiles.

Si se logra crear un método capaz de estimar costos basados en evidencias históricas, controlar y monitorizar los costos periódicamente, se logrará minimizar la pérdida de costos y ajustar el programa a los tiempos y compromisos con el cliente.

El objetivo de esta investigación es proveer un método para la administración y monitoreo del presupuesto además de la estimación de costos en los métodos ágiles, siguiendo los principios del manifiesto ágil y abordando los problemas de administración, monitorización, generación de evidencias y estimación de costos.

El presente artículo está organizado de la siguiente manera: la sección de trabajos relacionados presenta algunos trabajos que proponen métodos para el control del presupuesto y la estimación de costos; en la sección de propuesta de medición de costos se desarrolla la propuesta del control de presupuesto y del proceso de estimación de costos; en la cuarta sección se desarrolla

el caso de estudio, desde su diseño hasta la interpretación de los resultados; por último se dan las conclusiones y el trabajo futuro; en la parte final del documento se agregó un anexo con la terminología más utilizada en el ambiente de los métodos ágiles.

## Trabajos relacionados

Se realizó una revisión de la literatura con temas que abordan los problemas en la administración de costos en métodos ágiles, se encontraron dos temas principales: el primero tiene que ver con el monitoreo y control del presupuesto de un proyecto y el segundo corresponde a la estimación del costo de desarrollo de un proyecto. En la sección de monitoreo y control del costo se expone el monitoreo del presupuesto y la sección siguiente trata acerca de la estimación de costos. El método ágil para administración de proyectos más común que se encontró en la literatura fue SCRUM (Deemer *et al.*, 2010), por lo que será este en el que se base este artículo.

### Monitoreo y control del presupuesto

En el artículo de Sulaiman y Barton (2006) se define una metodología llamada AgileEVM, que tiene el propósito de monitorear el avance de un proyecto en los atributos de tiempo y costo. AgileEVM se basa en el método de administración del valor ganado (EVM, *Earned Value Management*). EVM se define en Project Management Institute (2003) como: “EVM es un método para la integración del alcance, cronograma y recursos, y para medir el desempeño del proyecto”.

En AgileEVM las métricas de SCRUM se relacionan directamente con las métricas de EVM tradicional buscando no agregar burocracia innecesaria al proceso de desarrollo.

El monitoreo del desempeño y el presupuesto del proyecto en AgileEVM se realiza al final de cada iteración, en estas revisiones se recolectan cuatro parámetros: el número de iteración actual (n), los puntos de historia terminados durante la iteración (PC), los puntos de historia agregados o eliminados de la lista de trabajo que se tiene que desarrollar en la siguiente iteración (*lista backlog*) durante la iteración actual (PA) y el costo de la iteración (SC).

Las métricas de monitoreo que AgileEVM muestra son el índice del desempeño del costo (CPI), el índice del desempeño en el tiempo (SPI) y los puntos de historia SP (*Story Points*) faltantes durante cada iteración (figura 1). Entonces, conociendo el CPI y el SPI es posible conocer si el proyecto marcha de acuerdo a lo planeado,

ya que si es así, el valor de CPI y SPI debe ser igual a 1. Si el CPI es mayor que 1 entonces se está gastando menos de lo planeado, un CPI menor que 1 significa que se está gastando más del presupuesto. Si el SPI es mayor que 1 entonces se terminará el proyecto antes de lo planeado y si el SPI es menor que 1 el proyecto terminará tarde.

En el artículo de Dan Rawsthorne (2008) se toma como base el método AgileEVM (Sulaiman y Barton, 2006), pero agregando a los índices CPI y SPI el valor del negocio ganado durante cada iteración (EBV), ya que Rawsthorne lo considera una métrica importante para conocer el valor de negocio que el equipo de desarrollo agrega al producto y que beneficia al cliente durante cada iteración. En el método propuesto se asigna un costo de desarrollo a cada SP y la estimación del costo del proyecto está basada en días por persona.

Al término de cada iteración, además de obtener los parámetros necesarios para conocer el CPI y el SPI, es indispensable conocer el valor de negocio (BV, *Business Value*) que aporta cada HU (*historia de usuario*), según Rawsthorne la línea derivada del EBV de un proyecto debe ser muy parecida a una línea S como la mostrada en la figura 2.

La investigación de Rusk en 2009 también está relacionada con el control y monitoreo de costos, su propuesta está fundada en el EVM descrito en el estándar ANSI/EIA-748. Rusk, basado en su experiencia menciona que el enfoque tradicional de EVM encaja bastante bien en los métodos ágiles y que es muy fácil entender su aplicación. Según Rusk, conocer el contexto financiero de los proyectos desarrollados en métodos ágiles no es posible utilizando únicamente las gráficas de *burn down* proporcionadas por la administración ágil.

Para monitorear el desempeño del proyecto y el contexto financiero, se utilizó una gráfica en donde la métrica de presupuesto y SP desarrolladas se representan con valores porcentuales durante cada iteración en relación con el total de SP del producto. Rusk

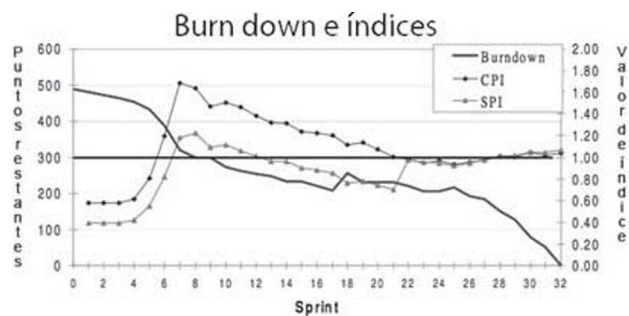


Figura 1. Gráfica para mostrar el valor ganado, CPI y SPI en AgileEVM (Sulaiman y Barton, 2006)

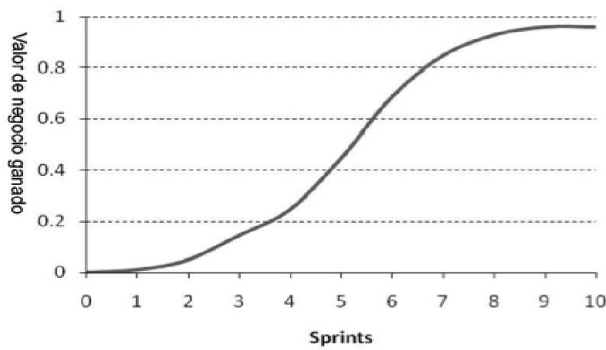


Figura 2. Línea S ideal para el EBV en un proyecto según Rawsthorne (2008)

utilizó una gráfica, como la mostrada en la figura 3, en donde dibujó una línea punteada correspondiente a una función lineal, basada en la velocidad del equipo de desarrollo, en la que se supone una velocidad constante para mostrar el avance esperado en cada iteración; utilizó una línea negra para mostrar el avance de SP desarrollado al finalizar cada iteración y una línea gris para ilustrar el gasto del presupuesto en cada iteración, a esta gráfica le da el nombre de gráfica de tiempo y presupuesto. En ella es posible identificar situaciones de riesgo cuando el presupuesto se consume más rápido de lo planeado o el costo del esfuerzo es mayor al esperado.

En el artículo de Miranda y Bourque (2010), se propone una técnica de monitorización de proyectos basándose en la técnica de línea de balance (LOB, *line of balance*) y utilizando puntos de control en las actividades del proceso de desarrollo, además se propone que la recolección de información se realice en los sistemas de control de versiones utilizados para la integración del producto. En esta técnica el monitoreo se realiza durante cada iteración aunque la información se recoge durante cada compromiso (*commit*) realizado al servidor del control de versiones por el equipo. Para obtener información desde los repositorios de código fue necesario estandarizar la manera en que se envían los commits, escribiendo la HU a la que pertenece y el punto de control en el que está ubicada esa HU. En la figura 4 se observa la forma en que este modelo muestra la información durante el monitoreo del desempeño en cada punto de control, la información mostrada en la gráfica tiene como métrica básica una HU. En este modelo solo es posible conocer la cantidad de requerimientos planeados y reales en cada punto de control en forma de HU durante un tiempo determinado.

Kang y Choi (2010) proponen un modelo de monitorización dinámico, la métrica básica de monitoreo son

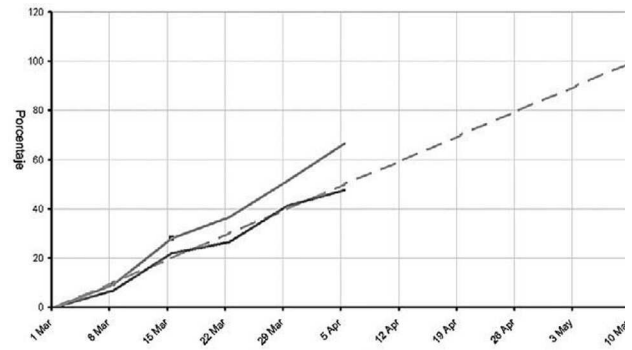


Figura 3. Gráfica utilizada por Rusk para monitorear el contexto financiero y el desempeño en un proyecto (2009)

los puntos de función FP (*Function Points*), los cuales describen la funcionalidad de cada HU. En este modelo el monitoreo es diario, por lo que se tiene una base histórica de información; este modelo utiliza el filtro de Kalman y el modelado de espacios de estado con la finalidad de hacer proyecciones muy precisas del avance en puntos de función que se espera tenga el producto según la velocidad histórica del equipo. El modelo espacio de estado se alimenta con la información de los FP faltantes y las variaciones del alcance en el proyecto. En la figura 5 se muestra la gráfica utilizada para el monitoreo (Kang y Choi, 2010), se puede ver una gráfica del tipo *burn down* en donde los FP son los valores del eje Y. En el eje X se muestran los FP faltantes por día.

En las metodologías mostradas en esta sección se encontraron puntos de mejora. Los métodos basados en EVM (Sulaiman y Barton, 2006; Rusk, 2009; Rawsthorne, 2008) no toman en cuenta el cambio en el alcance de un proyecto, es decir, la replaneación no se muestra en las gráficas de apoyo para el monitoreo, siempre se considera únicamente el valor obtenido de la primera planeación como un todo, por lo que la información que se

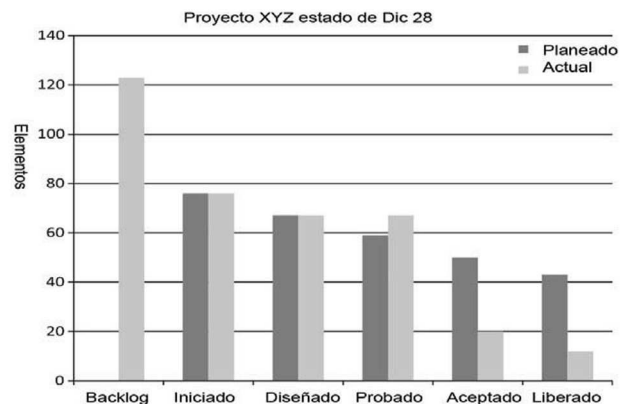


Figura 4. Gráfica para el monitoreo de HU por Punto de Control utilizado en LOB (Miranda y Bourque, 2010)



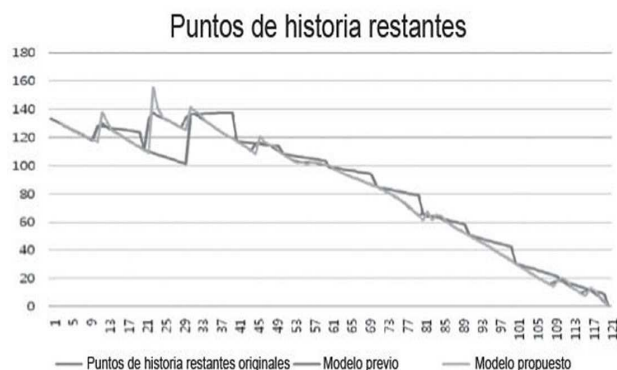


Figura 5. Gráfica con la información resultante del filtro Kalman (Kang y Choi, 2010)

muestra no es confiable al realizar cambios en el alcance del proyecto. En el método descrito en Kang y Choi (2010) se utiliza la aserción de que los FP son valores absolutos, por lo que son más precisos para planear el costo y tiempo de un producto, la inversión de mayor esfuerzo para planear un producto en FP no está justificada para obtener una planeación más precisa. En el método de Miranda y Bourque (2010) la precisión de cada punto de control se muestra con una unidad mínima de HU, por lo que el esfuerzo invertido a cada HU no se refleja en su metodología.

Los puntos de mejora identificados en las propuestas de Kang y Choi (2010) y Miranda y Bourque (2010) fueron contemplar los cambios de alcance del producto en la replaneación durante cada iteración, además de utilizar los puntos de historia como métrica para el esfuerzo.

### Estimación de costos

En los métodos ágiles se utilizan prácticas empíricas basadas en el juicio de expertos para estimar el costo de desarrollo para un nuevo producto, en la literatura se identificaron dos prácticas comunes para esta finalidad, la más común involucra como métrica base a los SP (Cohn, 2005; Sulaiman y Barton, 2006; Rusk, 2009; Rawsthorne, 2008; Miranda y Bourque, 2010) y la segunda se basa en FP (Kang y Choi, 2010).

La primera manera identificada, que es la más común en métodos ágiles, tiene como métrica principal a los SP (Cohn, 2005). En esta práctica la primera tarea a realizar es la estimación del esfuerzo total necesario para desarrollar el proyecto, esta estimación la realiza el equipo encargado de su desarrollo. En seguida, se define una velocidad base adquirida de datos históricos no muy confiables, o del juicio de expertos, para conocer el tiempo aproximado en el que el proyecto termi-

nará. Conociendo el tiempo necesario para desarrollar el proyecto, el líder del proyecto calcula el costo de las personas que forman parte del equipo para obtener el presupuesto necesario para el recurso humano, a tal estimación además se agregan otros tipos de gastos que el proyecto necesite y se agrega una suma acumulada de gastos al total del presupuesto estimado (Cohn, 2005).

Otra manera de obtener la estimación de costos se propuso en Kang y Choi (2010), donde la métrica base para realizar estas estimaciones es utilizar FP, lo que involucra más esfuerzo en la planeación del producto, pero los autores aseguran que su modelo es más preciso, ya que los FP son valores absolutos y no relativos como los SP. La forma en que realizan la estimación del costo del proyecto es similar a la que se utiliza comúnmente en métodos ágiles, con la diferencia que a cada HU le corresponde un conjunto de FP, los cuales tienen un costo basado en líneas de código o en personas por mes.

Las prácticas de estimación de costos basadas en juicios de expertos y que utilizan SP carecen de un modelo formal para realizar estimaciones, los datos históricos de los proyectos de un equipo en una empresa no siempre se consideran para realizar futuras proyecciones en la estimación de costos, posiblemente debido a la burocracia que acarrearía generar esta información. En la metodología propuesta por Kang y Choi (2010) se toman en cuenta los datos históricos, pero únicamente los del proyecto en que se trabaja, además de que el costo está en función de FP, al ser una métrica de esfuerzo que implica un mayor detalle en la planeación de las HU.

Las mejoras potenciales que se encuentran en la estimación de costos son realizar estimaciones de costos basados en datos históricos de los proyectos, con atributos similares en una empresa, utilizando SP y ajustar el costo por SP durante cada iteración basado en el desarrollo del proyecto actual.

### Propuesta de medición de costos

En la literatura se encontraron algunas deficiencias en los métodos de monitoreo y control del presupuesto además de la estimación de costos. En el monitoreo y control del presupuesto se encontró que el alcance de los proyectos no se planea de nuevo al finalizar cada iteración, por lo que es la re-planeación una de las partes fundamentales en los métodos ágiles (Cohn, 2005) para adaptarse al cambio. En la sección de motivación se muestra una propuesta que considera el cambio del alcance en el monitoreo y control de costos apegándose a la administración del valor ganado (EVM) y a los principios del manifiesto ágil. En la sección de análisis e interpretación se propone una técnica para estimar

el costo de un proyecto de desarrollo basado en los tiempos históricos para un conjunto de HU con el mismo esfuerzo.

La propuesta está dirigida a empresas que desarrollan *software* utilizando métodos ágiles, en las que se tenga un plan de estimación basado en métricas relativas y que ven la necesidad de monitorear el presupuesto de sus proyectos, además de ver la necesidad de tener una base histórica para la estimación de costos basada en hechos históricos dentro de la organización.

### Monitoreo y control del costo

Las deficiencias identificadas en el monitoreo y control de costos de la sección de monitoreo y control del presupuesto se solucionan en este punto del artículo.

Debido a que el plan de desarrollo en los métodos ágiles debe ser flexible y cambiante por la propia naturaleza de los proyectos de desarrollo de *software* (Cohn, 2005), este debe ser monitoreado y controlado para evitar poner en riesgo el éxito del proyecto, negociar con el cliente los cambios del alcance, reducir la incertidumbre del estado actual del proyecto, apoyar la toma de decisiones con el cliente y el equipo de desarrollo, generar confianza y transmitir información fiable a los involucrados.

En esta propuesta se utilizan tres gráficas para observar la información agrupada y que sea fácil de comunicar y entender por cualquier persona, aún sin conocimiento en administración de proyectos de desarrollo de *software*. Se identificaron dos actividades importantes para poder realizar el monitoreo, y fue la planeación el primer acercamiento en donde se generó un plan de trabajo y estimación de recursos necesarios para el proyecto; la segunda actividad necesaria para permitir el monitoreo y el control del proyecto conforme evoluciona es la replaneación, en donde se identifican y se administran riesgos para poder terminar el proyecto con éxito.

### Planificación

La planificación es la actividad donde surge el primer acercamiento detallado de las necesidades del cliente y donde se identifican los recursos necesarios para el desarrollo de un producto de *software*. En esta propuesta no se tratan la identificación de las necesidades o la definición de los requerimientos de cliente, pero sí es importante que el tamaño del producto utilice la métrica relativa de puntos de historia, ya que es la más utilizada en los métodos ágiles y no agrega esfuerzo

en la actividad de la estimación de esfuerzo requerido por HU, como en Kang y Choi (2010) donde se utilizan puntos de función.

En esta propuesta se realiza la planeación utilizando EVM, esta planificación va a ser cambiante durante cada iteración y se explica en el punto del objetivo de investigación.

Para planear un nuevo proyecto de *software* es necesario conocer las siguientes métricas base:

- **Tamaño base total del producto (TBTP):** es el tamaño del producto definido en puntos de historia. Para conocer este valor se puede aplicar la siguiente fórmula:

$$TBTP = \sum_{i=0}^n \text{Esfuerzo}_{HUi} \quad (1)$$

- Donde HUi es el conjunto de Historias de usuario del proyecto y Esfuerzo es la métrica con que se mide el tamaño de las historias de usuario definido por puntos de historia (SP).
- **Velocidad base del equipo (VB):** es la cantidad de puntos de historia pronosticada que el equipo de desarrollo entregará al usuario durante cada iteración. Esta métrica se obtiene comúnmente por juicio de expertos. En la sección de análisis e interpretación se hace una propuesta para obtener esta métrica de una forma más precisa, basada en información histórica de proyectos con atributos similares desarrollados en una organización.
- **Número de Iteraciones (I):** es la cantidad de iteraciones que el equipo necesita para desarrollar el producto. Para conocer este valor se aplica la siguiente fórmula:

$$I = \frac{TBTP}{VB} \quad (2)$$

- Para obtener un valor entero el resultado de I debe ser redondeado hacia su entero próximo.
- **Duración de iteración (DI):** indica la duración en tiempo de cada Iteración, en donde el rango común es entre 2 semanas y 4.
- **Costo base del producto (CBP):** define el costo a invertir para el desarrollo del producto. En los métodos ágiles no existe una práctica o método estándar para estimar el costo de un producto.
- **Costo base del equipo por iteración (CBI):** se define como el costo monetario que se espera gastar durante cada iteración. Para conocer este valor se utiliza la siguiente fórmula,

$$CBI = \frac{CBP}{DI} \quad (3)$$

- **Costo base de SP por iteración (CBSPI):** es el costo monetario por desarrollar un punto de Historia y se puede calcular con la siguiente fórmula,

$$CBSPI = \frac{CBI}{VB} \quad (4)$$

Además de las métricas base en esta actividad, se obtiene la gráfica *burn down* de planeación, en la que se grafica la cantidad de SP faltante por desarrollo durante cada iteración. Para obtener esta gráfica es necesario conocer la cantidad de SP que el equipo va a desarrollar durante cada iteración basándose en la información de la planeación. La función a graficar es la siguiente:

$$f(x) = TBTP - \sum_{i=0}^x SPI_i \quad (5)$$

en donde  $x$  es la iteración que se va a graficar y SPI son los SP a desarrollar en la iteración  $i$ .

Para ejemplificar esta actividad tomamos como ejemplo el proyecto “Sistema de gestión de contenidos” desarrollado en la empresa Compulogic, a través de su división de *software*, en donde el equipo de desarrollo está conformado por un líder de proyecto y 5 desarrolladores. El equipo levantó requerimientos con un total de 46 HUs, cada una con su respectivo esfuerzo, donde se obtuvo un TBTP de 486 SP. El líder del proyecto consideró una VB de 70 SP por iteración basándose en las velocidades logradas en proyectos anteriores; con el TBTP y VB se obtienen 7 iteraciones necesarias para desarrollar el producto con una duración de 2 semanas cada una. Además, el líder estimó el costo del producto basándose en los sueldos de los desarrolladores que forman parte del equipo y del tiempo que durará el proyecto, de donde obtuvo un CBP de \$252,000.00<sup>1</sup>. El equipo planea los SP a desarrollar en cada iteración y obtiene la información que se muestra en la tabla 1.

Tabla 1. SP planeados por iteración para el proyecto “Sistema de gestión de contenidos”

Iteración	SPI
1	66
2	71
3	71
4	65
5	65
6	68
7	80

1 Para conservar la confidencialidad con los costos dentro de la empresa, el costo referenciado se cambió.

Tabla 2. Métricas base para el proyecto “Sistema de gestión de contenidos”

Métrica	Valor obtenido
TBTP	486 SP
VB	70 SP/Iteración
DI	2 semanas
I	7
CBP	\$252,000.00
CBI	\$36,000.00
CBSPI	\$ 563.74

El resumen de la información se ilustra en la tabla 2. El *burn down* de planeación se muestra en la figura 6.

### Replanificación

La gráfica de *burn down* de planeación evolucionará durante cada revisión de iteración, en ella se mostrará que:

- **El avance base planeado (AB)** se grafica con la misma función utilizada en la planeación.
- **El avance real (AR)** mostrará el valor ganado durante cada iteración incluyendo los cambios en el alcance durante cada iteración. Los cambios en el alcance se mostrarán a la alza cuando se agreguen SP al producto y a la baja cuando se eliminen.
- **El avance replanificado (ARR)** mostrará los ajustes que el equipo toma durante cada planeación de una nueva iteración. Durante cada planeación de iteración el equipo decide qué HU va a desarrollar, se basa en la información de la planeación, pero esta es susceptible a cualquier tipo de cambio generado por petición del involucrado para mitigar riesgos o para eliminar restricciones.

Para controlar el cambio del alcance y mantenerlo visible durante cada iteración se propone la *gráfica de cambios en el alcance*, que mostrará el impacto de estos cambios en los SP agregados o eliminados del TBTP.

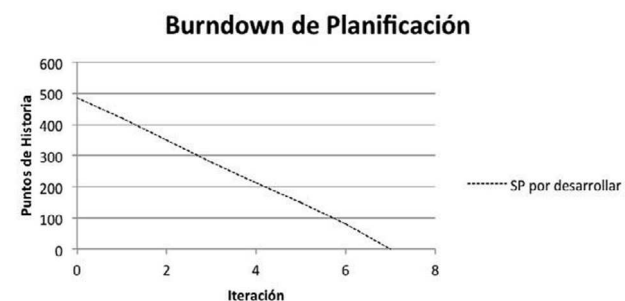


Figura 6. *Burn down* de planeación para el proyecto “Sistema de gestión de contenidos”

En esta actividad es necesario obtener las siguientes métricas:

- *Costo actual de la iteración (CAI)*: define el costo monetario de una iteración.
- *SP desarrollados en la iteración (SPAI)*: define los SP desarrollados en una iteración. Solo se contabilizan los SP aceptados por el cliente al finalizar una iteración. También se puede nombrar velocidad de iteración.
- *Costo actual de SP (CASPI)*: define el costo monetario por desarrollar un SP en una iteración. Para conocer este costo se utiliza la siguiente función:

$$CASPI = \frac{CAI}{SPAI} \quad (6)$$

- *Índice de desempeño del costo (CPI)*: esta métrica permite visualizar la fluctuación del costo por SP contra el costo por SP real durante cada iteración. Este índice permite conocer si el equipo desarrolla los SP bajo lo presupuestado en la planificación o está excediendo el presupuesto y pone en riesgo el éxito del proyecto. El CPI ideal debe tener un valor igual a 1. Cuando el CPI es mayor que 1 significa que el costo por SP es menor al planeado y si el CPI es menor que 1 significa que el costo por SP es mayor que el planeado. La fórmula para conocer este índice es la siguiente:

$$CPI = \frac{CBSPI}{CASPI} \quad (7)$$

- *Índice del desempeño del calendario (SPI)*: este índice es el que permite realizar el monitoreo del calendario, el equipo conocerá la fluctuación de la velocidad planeada respecto a la real durante cada iteración. Cuando el SPI obtenido durante una iteración es mayor que 1, el proyecto terminará antes de lo planeado; si el SPI es menor que 1, el proyecto terminará después de la fecha planeada. La fórmula para conocer este índice es la siguiente:

$$SPI = \frac{SPA1}{VB} \quad (8)$$

- *Cambios en el alcance (SPCA)*: esta métrica permitirá conocer la cantidad de SP agregados o eliminados al TBTP durante una iteración.
- *Total de cambios en el alcance (TSPCA)*: con esta métrica será posible monitorear el total de cambios realizados en el proyecto. En EVM se considera exitoso un proyecto que varió 30% su tiempo y costo (Rusk, 2009), es por ello que se recomienda que esta métrica nunca sea mayor que 30%. Para obtener esta métrica se utiliza la siguiente función:

$$TSPCA(n) = \sum_{i=1}^n SPCA_i \quad (9)$$

en donde se calculan el total de cambios de alcance en cada iteración hasta una iteración  $n$ .

El monitoreo y control de los cambios en el alcance se contabilizan al finalizar cada iteración en la reunión con el cliente para negociar los cambios, si es necesario, ya que es recomendable que el cambio del alcance no sea mayor que 30% del TBTP.

Para ejemplificar la actividad de replaneación continuaremos con el proyecto de "Sistema de gestión de contenidos" en donde se ilustrará el desarrollo del proyecto durante las siete iteraciones. En la figura 7 se muestra la gráfica de *burn down* de iteración obtenida al finalizar el proyecto. En esta gráfica es posible apreciar que en las iteraciones 1 y 4 hubo cambios en el alcance y repercutieron para que se hiciera una replanificación de las HU a desarrollar durante cada iteración.

Como se mencionó, los cambios en el alcance del proyecto también se monitorean en esta propuesta con la gráfica de cambios en el alcance; la figura 8 ilustra el TSPCA, el cual permaneció menor que 30% del TBTP.

En la figura 9 se puede ver la fluctuación de los índices CPI y SPI. En la tabla 3 se muestra la información recabada durante cada iteración para poder realizar la figura 9.

En la iteración 6 se integró al equipo de desarrollo un miembro más para colaborar y terminar en tiempo el proyecto. Se aprecia que el CPI en las iteraciones 6 y 7 es mayor que 1, por lo que el costo de SP fue mayor al planeado. En la columna de SPI se aprecia que en las iteraciones 6 y 7 la velocidad del equipo fue mayor a la planificada, lo que permitió terminar en tiempo.

El costo real por iteración del proyecto se muestra en la tabla 4. El precio real del proyecto fue de \$ 258,923.08 incrementándose en un 2.74%. Debido a que el proyecto terminó en tiempo y con un presupuesto mayor en un 2.74%, puede ser considerado como exitoso.

## Estimación del costo

En esta sección se presenta una técnica para estimar el costo de un proyecto basándose en datos históricos de la organización. Vale la pena aclarar que para planear un nuevo proyecto, los datos históricos deben provenir de proyectos con características similares en cuanto a la tecnología, que se utilicen SP o alguna métrica semejante para su planificación y que el equipo de desarrollo del nuevo producto sea maduro.



La estimación de costos para el desarrollo de un nuevo producto es una de las tareas más difíciles en ingeniería de *software*, por lo que no es novedoso que en métodos ágiles sea una tarea empírica y casi siempre basada en juicios de expertos quienes confían en su amplia experiencia.

La técnica propuesta en este artículo involucra el uso de la información generada de proyectos desarrollados en una organización, por lo que cada organización obtendrá resultados diferentes.

Para estimar el costo de un proyecto en esta propuesta la base es el tiempo de desarrollo necesario, por lo que el costo de un proyecto es directamente proporcional al tiempo necesario para terminarlo. El tiempo necesario para desarrollar una HU se obtiene de datos históricos que se procesan utilizando la desviación estándar y una gráfica con la campana de Gauss para identificar el tiempo más confiable para las HU que tengan un mismo esfuerzo, es decir, cada conjunto de las HU estimadas en 8 SP tendrán asignados inherentemente un tiempo estimado para su desarrollo, así como las estimadas en 2 SP, 4 SP, 13 SP, 20 SP, etcétera. Esta serie se considera, ya que se utiliza en la planeación de métodos ágiles (Project Management Institute, 2003; Sulaiman y Barton, 2006), pero cada organización puede utilizar su propia serie de valores relativos.

El tiempo de desarrollo de una HU se considera desde que un desarrollador la toma en la fase de desarrollo, hasta que el cliente la acepta. Los tiempos muertos durante el proceso de desarrollo no deben contabilizarse, solo el tiempo invertido en trabajo sobre una HU.

Conocer el tiempo real de desarrollo de cada HU es difícil y sería costoso que el equipo recabara esa información de forma manual, por lo que se sugiere obtener la información de herramientas desarrolladas con BPMN que permiten la automatización de los procesos de la organización y su completo monitoreo, herra-

mientas como BizAgi permiten obtener esta información de una forma fácil y sin invertir mucho esfuerzo para obtener información de calidad. Este artículo no propone una herramienta para extraer el tiempo preciso de desarrollo por cada HU.

Para realizar esta técnica de estimación de costos es necesario contar con la información de los tiempos históricos del desarrollo de HU de la organización agrupados según su esfuerzo, de forma similar a la que se muestra en la tabla 5.

En la tabla 5 se aprecian los tiempos gastados en horas y agrupados por esfuerzo para las HU de un proyecto. Se comprobó que tener un rango muy amplio entre los tiempos de desarrollo arroja menor precisión al graficar la distribución normal para la campana de Gauss, es por eso que se sugiere que los tiempos se capturen con valores que respeten una serie, con incrementos de un cuarto de hora, es decir, podremos representar el tiempo de las HU con la siguiente serie y sus combinaciones: 0.25 hora, 0.5 horas, 0.75 horas, 1 hora.

Con esta serie de tiempos se tiene la información más agrupada en la campana de Gauss lo que permite tener mayor precisión. En la tabla 6 se muestra un ejemplo de la forma en que se verían los tiempos invertidos para las HU con esfuerzo de 8 SP.

Con la información de los tiempos de desarrollo como en la tabla 6, el siguiente paso es calcular la media, la varianza ( $S^2$ ) y la desviación estándar ( $S$ ).

Calcular la varianza del conjunto de tiempos por cada conjunto de esfuerzos servirá para conocer su dispersión, la varianza se define como la media de las diferencias cuadráticas de  $n$  puntuaciones con respecto a su media aritmética y se obtiene con la siguiente fórmula:

$$S^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (10)$$

donde  $x_i$  es el tiempo para una HU,  $n$  es el total de datos de la muestra y  $\bar{x}$  la media del conjunto de datos.

La desviación estándar del conjunto de datos hará que la medida de dispersión sea de la misma dimensión que las muestras, en este caso horas, se obtiene sacando la raíz cuadrada de  $S$ .

$$S = \sqrt{S^2} \quad (11)$$

Los valores de la media, de  $S^2$  y de  $S$  para el conjunto de datos de la tabla 6 se muestran en la tabla 7.

Utilizando la distribución normal para los tiempos de un conjunto de HU con el mismo esfuerzo nos permitirá una mejor entropía entre los tiempos recolectados para el conjunto de HU con esfuerzos iguales.

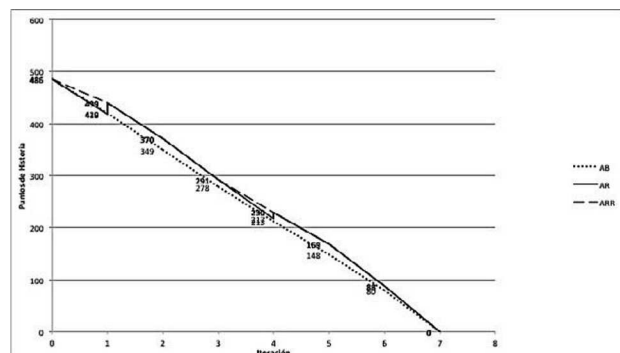


Figura 7. Burn down de iteración 8 para el proyecto: "Sistema de gestión de contenidos"

La distribución normal se calcula con la siguiente fórmula:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (12)$$

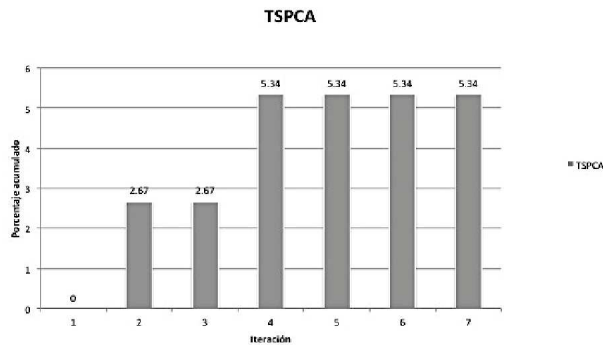


Figura 8. TSPCA para el proyecto: "Sistema de gestión de contenidos"

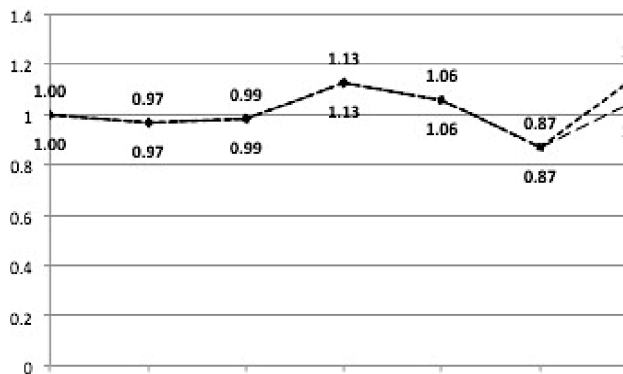


Figura 9. Índices CPI y SPI para el proyecto: "Sistema de gestión de contenidos"

donde  $x$  es un tiempo del conjunto de las HU con mismo esfuerzo,  $\mu$  es la media o  $x$  y  $\sigma$  es la desviación estándar o  $S$ .

En la tabla 8 se calcula la distribución normal para la  $x$  y la  $S$  obtenidas con la información de la tabla 6. La columna de grupos muestra el rango identificado de tiempos para las HU de 8 SPs de la tabla 6.

En la figura 10 se muestra la campana de Gauss o distribución normal para los tiempos de HU de un proyecto con esfuerzo de 8 SP. En ella se puede apreciar que para una  $x$  de 1 hora se tiene un valor muy cercano a 1, lo que significa que para una HU de 8 SP con  $x$  y  $S$  obtenidas de los históricos mostrados en la tabla 11 existe 91.3% de probabilidad de que el esfuerzo necesario para su desarrollo sea de 1 hora.

Con la campana de Gauss podemos conocer el tiempo con mayor precisión de la serie de valores históricos de las HU en un esfuerzo determinado (ej. 8 SP), identificando el valor preciso como la  $Y$  más alta en la curva. De esta forma, para estimar el costo de un nuevo proyecto podemos confiar en los datos históricos de los tiempos de desarrollo invertidos en las HU con esfuerzos iguales.

## Caso de estudio

El presente caso de estudio se diseña de tal forma que pueda presentar las discusiones respecto a dos proyectos de una empresa de *software* utilizando el método de medición propuesto. Primero, se da una motivación señalizando el foco del problema, el objetivo de la investigación y el problema. Después se presentan los resultados de las gráficas del método de medición y las interpretaciones de sus resultados, así como también las discusiones sobre las dificultades y problemas presentados en el estudio.

Tabla 3. CPI y SPI del proyecto: "Sistema de gestión de contenidos"

Iteración	Costo por SP planeado	Costo replaneado siguiente iteración	Costo real	CPI	SPI
0				1	1
1	514.2857143		580.3167421	0.971428571	0.971428571
2	514.2857143	467.5324675	521.7391304	0.985714286	0.985714286
3	514.2857143	461.5384615	455.6962025	1.128571429	1.128571429
4	514.2857143	486.4864865	486.4864865	1.057142857	1.057142857
5	514.2857143	404.494382	590.1639344	0.871428571	0.871428571
6	514.2857143	448.4265734	487.1794872	1.055639098	1.157142857
7	514.2857143		448.4265734	1.146867168	1.257142857

Tabla 4. Costo real por iteración en el proyecto

Iteración	Costo Real
1	\$ 36,000.00
2	\$ 36,000.00
3	\$ 36,000.00
4	\$ 36,000.00
5	\$ 36,000.00
6	\$ 39,461.54
7	\$ 39,461.54
	\$ 258,923.08

## Motivación

Para informar al lector del foco y el alcance del presente trabajo, a continuación se presenta el problema y los objetivos de investigación, además de los fundamentos para la comprensión del contexto del artículo.

### Establecimiento del problema

En este artículo se tomaron en cuenta los siguientes problemas para los *administradores de proyectos* y *expertos* en metodologías ágiles para el desarrollo de *software*:

1. Varios autores han encontrado una necesidad clara de los gestores de proyectos de *software* en métodos ágiles. Existe una *escasa gestión y monitoreo* de costos en métodos ágiles (Yap, 2006; Keaveney y Conboy, 2006; Sulaiman y Barton, 2006; Rusk, 2009).
2. Otros autores indican que hay *poca evidencia* acerca de la medición de los costos de un proyecto para los administradores de proyectos (Jones, 2004; Sulaiman y Barton, 2006; Rusk, 2009).
3. La *estimación de costos* en métodos ágiles se basa comúnmente en juicios de expertos y no en *procesos repetibles y mejorables* (Keaveney y Conboy, 2006; Pham A. y Pham P., 2011). Sin un proceso bien definido y medible, no es posible lograr una optimización ni la reducción de costos.

Tabla 5. Tiempos agrupados por esfuerzo para un proyecto

Conjunto esfuerzo 2	Conjunto esfuerzo 4	Conjunto esfuerzo 8	Conjunto esfuerzo 13	Conjunto esfuerzo 20
1.10	7.00	5.2	19.2	23.6
1.60	7.00	16	27.5	31.4
1.45	7.00	8.3	23	37.2
13.40	7.00	9.2	28.8	35.1
4.50	8	7.5	15.1	30.5
6.50	7.5	11.7	23.2	42
3.10	7	23.5		
3.60	8	17.2		
2.00	7	21.2		
	7	13.5		
	6	12.1		
	7.5	21.3		
	7	19.6		
	8	11.4		
	8	15.3		
	7	18.7		
	7	21.1		
	6	25.6		
	7	17		
	7.5	25.2		

En suma, el problema es *la falta de una efectiva gestión, monitorización, generación de evidencias y estimaciones basadas en procesos mejorables en costos para proyectos de software desarrollados con métodos ágiles*. Los principales afectados son los administradores de proyectos, los expertos en métodos ágiles y los dueños de las empresas de desarrollo de SW, con respecto a la pérdida de costos, precisión en los tiempos y compromisos con el cliente.

### Objetivo de investigación

El objetivo de esta investigación es analizar el *control, monitorización y generación de evidencias* por parte de los desarrolladores y los administradores de proyectos durante el desarrollo de dos proyectos de *software* en una pequeña empresa del ramo. Su propósito es descubrir las *dificultades o problemas* con respecto al uso de la propuesta en cuanto a la *pérdida de costos, precisión en los tiempos, imprevistos y compromisos con el cliente*, desde la perspectiva de un *administrador de proyectos* en el contexto del uso de métodos ágiles.

### Contexto

El estudio se llevó a cabo en una pequeña empresa de desarrollo de *software* basado en procesos ágiles, específicamente en *Scrum* y *eXtreme Programming*. Dos proyectos de *software* fueron parte del estudio:

- El **primer proyecto (P1)** fue el desarrollo de un *software* como servicio (SaaS, *Software as a Service*) para gestionar la red de colaboración entre la triple hélice (gobierno, industria y academia). Se registraron en el equipo un total de 7 desarrolladores, un total de 75 HU planificadas, 22 SP por iteración, un costo base de \$128.68 pesos mexicanos por cada SP (el costo no contiene impuestos ni ganancias para la empresa) y el periodo de desarrollo fue del 4 de junio al 8 de agosto. En este proyecto se utilizó la propuesta de estimación y control de costos.
- El **segundo proyecto (P2)** fue el desarrollo de una herramienta de diseño y seguimiento de indicadores, donde se definieron iteraciones de duración semanal, con 4 iteraciones planificadas. Se registró en el equipo un total de 8 desarrolladores durante el proceso, 47 historias de usuario planeadas, una estimación de 263 puntos de historia, una velocidad base de 69 SP por iteración y un costo base de \$128.68 pesos mexicanos por cada SP. En este proyecto no se utilizó la propuesta de estimación y control de costos del presente artículo. Por último, el

periodo de desarrollo fue del 25 de septiembre al 8 de noviembre.

## Análisis e interpretación

A continuación se presentan los resultados finales de los proyectos en las gráficas correspondientes.

El proyecto P1 no generó los suficientes datos para generar las gráficas de estimaciones de esfuerzos, ya que no se contaba con datos históricos suficientes. Sin embargo, la estimación de los expertos fue suficiente, ya que este era un diseño de desarrollo conocido por el cliente. Las gráficas generadas del método propuesto son la gráfica *burn down* al cierre del proyecto, como se puede apreciar en la figura 11, y la gráfica CPI/SPI también al cierre del proyecto en la figura 12.

Las series de datos de la gráfica *burn down* se describen como el avance planeado, que muestra el esfuerzo restante para su desarrollo, obtenido durante la etapa



Figura 10. Distribución normal para los tiempos de HUs con esfuerzo de 8 SP

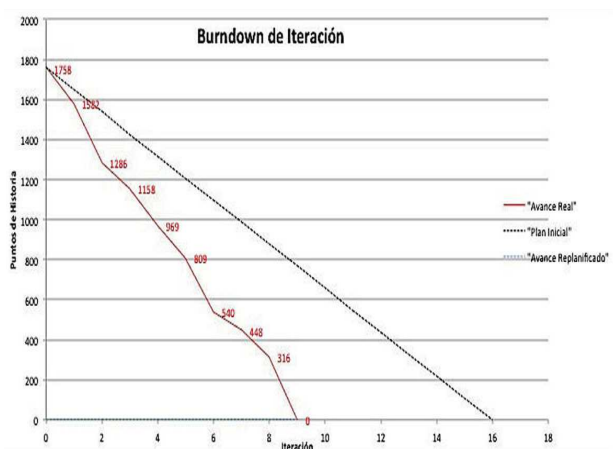


Figura 11. *Burn down* al cierre del proyecto P1

de planeación; el avance real, que muestra el valor ganado durante cada iteración, así como los cambios en el alcance, mostrando picos cuando el alcance crece o se agrega esfuerzo al desarrollo y el avance re-planificado, que muestra el avance replanificado para cada iteración. Como se puede apreciar en los resultados, el avance real fue más rápido de lo planeado debido al conocimiento de la solución del sistema, es decir, un sistema bastante conocido por los expertos respecto a las entrevistas con el cliente. Es por esto que no fue necesaria una replanificación. Este método de medición es de mayor utilidad para proyectos en donde se desconoce el problema para poder tomar decisiones respecto a las replanificaciones, como resultados esperados de esfuerzo.

Respecto a la gráfica de CPI/SPI en el cierre del proyecto (figura 12), no muestra resultados inesperados conforme al tiempo y al costo, fueron resultados controlados en todo momento. Los resultados fueron, por lo mismo, una solución conocida y entendida por las necesidades del cliente. Por último, no se generó una gráfica TSPCA debido a que no se creó un valor agregado a las historias de usuario durante el desarrollo del proyecto.

En el segundo proyecto, al igual que en el primero, aún no se tenían los suficientes datos históricos para generar las estimaciones sobre los esfuerzos. La gráfica de la que se parte para la toma de decisiones es la *burn down* en la que se muestran las dificultades de control, principalmente de la segunda iteración a la quinta; para

Tabla 6. Tiempos históricos del proyecto para HUs con esfuerzo de 8 SP utilizando la serie propuesta de tiempos

Conjunto de tiempos para esfuerzo 8
1.00
1.00
0.50
1.00
1.5
1
1.25
1
0.75
0.5
0.5
1.25
1.5
1
0.5
0.75
1.75
1.25
2
0.5
1

Tabla 7. Valores de  $\bar{x}$ ,  $S^2$  y  $S$  para el conjunto de tiempos de HUs con esfuerzo de 8 SP de la tabla 6

$\bar{x}$	1.03
$S^2$	0.190131579
$S$	0.4360408



Tabla 8. Distribución normal para la  $x$  y  $S$  para el grupo de tiempos de HUs con 8 SP, identificados en la tabla 6

Grupos	Distribución Normal
0	0.057739813
0.25	0.188545233
0.5	0.443194719
0.75	0.749915682
1	0.913417089
1.25	0.800874872
1.5	0.505474255
1.75	0.229653069
2	0.075107672

efectos de encontrar las dificultades y problemas del método de medición se discuten dichas iteraciones. Las gráficas utilizadas son las de la figura 13 con la *burn down* al cierre del proyecto, la figura 14, con gráfica CPI/SPI al cierre del proyecto, y la figura 15 con la gráfica TSPCA para el cierre del proyecto.

A continuación se describen las discusiones de los problemas y las dificultades de las iteraciones en el proyecto.

### Iteración 2

El esfuerzo que el equipo estimó desarrollar en esta iteración fue de 66 SP, pero realmente el *Valor Ganado* (EV) fue 50 SP. Este resultado en EV tiene un impacto directo en el costo; el índice del desempeño del costo (CPI) para esta iteración fue 0.73, lo que muestra que el costo por SP fue más caro que el valor planeado, teniendo un costo de \$175.00, 36% más caro para el desarrollo de un SP. Además, el índice del desempeño del calendario (SPI) fue 0.73, indicando que el equipo desarrolló más lento de lo planeado, es decir, 24.24% más lento. El problema fue que no se estimó de una forma precisa la velocidad de desarrollo y no se estimaron los riesgos de nuevas tecnologías adquiridas en la empresa.

Al cierre de la iteración 2, el equipo se reunió con el cliente para mostrar los avances y obtener alguna retroalimentación acerca de las tareas desarrolladas. Como resultado se agregaron 135 SP como descubrimiento de nuevas evidencias, que representaron 51.3% del total del producto planeado, lo que ponía en riesgo la terminación en tiempo y costo del desarrollo del producto, además de que la velocidad de desarrollo era más lenta. Por estas razones se decidió agregar 2 desarrolladores más al inicio de la tercera iteración.

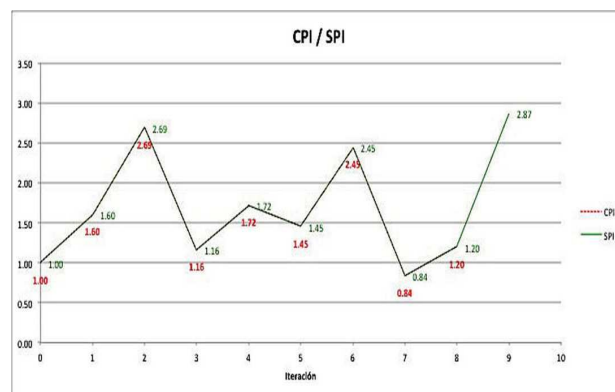


Figura 12. CPI/SPI al cierre del proyecto P1

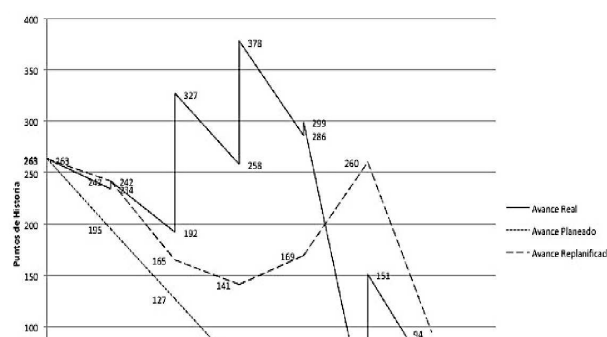


Figura 13. Burn down al cierre del proyecto P2

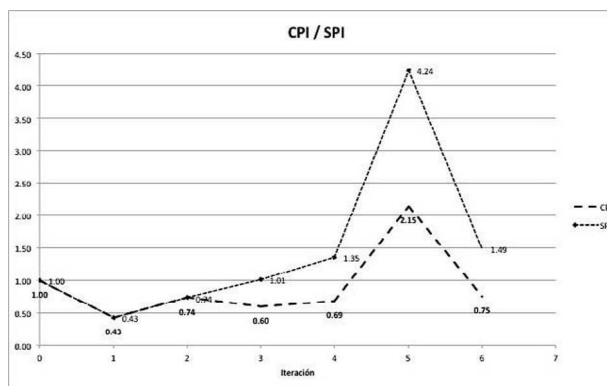


Figura 14. CPI/SPI Al cierre del proyecto P2

### Iteración 3

Al inicio de la iteración 3 el alcance del proyecto ya había aumentado 54 %, por lo que se agregaron dos desarrolladores más al equipo para aumentar la productividad de SP en el proyecto. El impacto en el costo de agregar a dos desarrolladores fue 68% más caro para esta iteración. En este proyecto trabajó el equipo 2; al término de la iteración la velocidad de desarrollo fue 69 SP, acorde a lo esperado en el plan, dicho de otra forma,

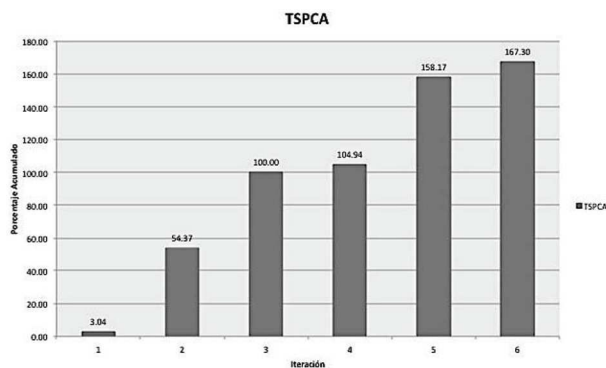


Figura 15. TSPCA para el cierre del proyecto P2

el SPI tuvo un valor igual a 1, ya que el equipo avanzó conforme a la planeación inicial. Aumentar el número de desarrolladores incrementó el costo de la iteración y el costo de desarrollo de un SP, el CPI fue 0.66, el cual evidencia el aumento en el costo, 66% más caro.

En el cierre de la iteración, al mostrar los avances al cliente, fue aumentado el alcance debido a nuevos hallazgos y características necesarias para el sistema, el total de SP agregados fue 120. Esta repercusión es notable visualizando la gráfica del *burn down* para este proyecto, ya que el equipo esperaba tener un restante de 59 SP pero subió a 378 SP el valor real. Tomando en cuenta estos valores el remanente al cierre de la iteración 3 se duplicó, en la figura 13 se observa un aumento de 100% más respecto al valor planeado.

#### Iteración 4

En esta iteración el encargado del desarrollo, agregó un desarrollador más, completando los 8 desarrolladores del equipo, esto ocasionó que subiera el costo por el número de integrantes. El valor ganado en esta iteración fue 92 SP, 35% más rápido en su desarrollo con un SPI equivalente a 1.35. El costo asumido para esta iteración fue mayor, el CPI obtenido fue igual a 0.68, 32% más caro.

Basados en la planeación inicial es en esta iteración cuando el proyecto debió terminar su fase de desarrollo, al estar aún 286 SP alejados de la meta, a la cual se agregaron otros 13 SP, con lo que el proyecto creció 104%. En suma, el equipo no terminó lo planeado a consecuencia de los cambios en el alcance, y fue más caro el desarrollo del proyecto. El problema fue aceptar 104% de cambios sobre lo planeado.

#### Iteración 5

En esta iteración el equipo de desarrollo a cargo fue el equipo 3. Al cierre, el valor ganado fue 288 SP, quedan-

do solo un restante de 2 SP. Acorde al SPI tenemos que la velocidad del equipo fue 4.23 mayor que la planeada (figura 14), 323% más rápido de lo planeado. El costo de desarrollar un SP en esta iteración fue más barato, con un CPI igual a 2.14, lo que significa que fue 54% más barato que lo planeado. Probablemente, los beneficios de la velocidad y el costo se debieron a la sobreestimación del esfuerzo planeado, por lo que es necesario realizar estimaciones más precisas.

Al finalizar la iteración se agregaron 149 SP, significando 53% del peso planeado para el proyecto y se obtuvo una suma de cambios en el alcance de 158.17% sobre el esfuerzo total planeado del proyecto. Nuevamente, el problema de este incremento fue aceptar más de 30% de cambios sobre el plan.

Conforme a las dificultades y problemas se concluye que es necesario crear un proceso de toma de decisiones utilizando las gráficas y los recursos humanos, de información y tecnológicos de que se disponga.

## Conclusiones y trabajo futuro

Se creó un método de medición para la estimación, control y gestión de costos y esfuerzos en equipos de desarrollo de *software* ágil con la finalidad de resolver los problemas encontrados en la literatura: una escasa gestión y monitoreo de costos, poca evidencia acerca de la medición de los costos de un proyecto para los administradores y falta de estimación de costos en métodos ágiles basada en procesos repetibles. La causa principal de estas deficiencias es que los métodos ágiles siguen los principios del manifiesto ágil. El principio de esta aclaración es: “El *software* funcionando es la medida principal de progreso”. Este principio se centra en la medida *software* del funcionamiento para medir un avance del proyecto, y el principio “A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia” se aleja de las medidas predictivas como son las estimaciones. Ambos principios restringen a los investigadores al crear propuestas precisas para las estimaciones de tiempos, esfuerzos y costos. Sin embargo, este trabajo ofrece tres tipos de gráficas para apoyar los problemas mencionados: la gráfica TSPCA para el control de cambios realizados en el proyecto y no sobrepasar el tamaño del proyecto de *software* en 30%, la gráfica CPI/SPI para el control de costo por SP y calendario, y la gráfica *burn down* que permite representar la replanificación de cada iteración, además de la técnica de estimación de esfuerzo con datos históricos de SPs.

A pesar de haber diseñado el método para resolver los problemas mencionados, la información de los da-

tos históricos de proyectos anteriores y actuales no fue suficiente para introducir datos y comprobar la eficacia de las evidencias de estimaciones. También se presentaron dificultades durante las iteraciones del segundo proyecto, por lo que los costos y esfuerzos no se controlaron correctamente. Como trabajo futuro es necesario crear un proceso de toma de decisiones para resolver las dificultades y problemas presentados en los proyectos de manera que se utilicen las tres gráficas para una toma de decisiones efectiva.

Para lograr realizar estas estimaciones sin afectar al manifiesto ágil, actualmente se está desarrollando una herramienta para generar las gráficas “Distribución normal para los tiempos de historias de usuario con esfuerzo en SP” y lograr realizar las estimaciones del esfuerzo de los proyectos con más precisión y poca interacción entre los desarrolladores y la herramienta. En un futuro, con el uso de la herramienta, se espera generar la información histórica automáticamente para lograr realizar estimaciones de esfuerzo precisas para nuevos proyectos.

Otro trabajo en un futuro se puede desarrollar a partir de calcular la cuenta de los tiempos gastados en una actividad dentro de un proceso, el cual suele ser difícil si se realiza de forma manual. Para ello, se realizará una propuesta basada en BPMN y en automatización de procesos, para conocer el tiempo invertido en cada HU dentro de un proyecto de *software*, utilizando la distribución normal para conocer la tendencia de un conjunto de HU con un mismo esfuerzo dado en SP.

## Anexo

### 1. Términos y definiciones comunes en métodos ágiles

Para entender mejor la forma en que los métodos ágiles monitorean y estiman el costo de los proyectos es necesario definir los conceptos utilizados en este tipo de administración. En este anexo se define qué es un punto de historia, qué es una historia de usuario, qué es una iteración (*sprint*), la velocidad de iteración y las gráficas *burn down*.

Las **historias de usuario (HU)** son la unidad más pequeña administrada en el proyecto y se definen en Jeffries *et al.* (2000) como: “Una promesa de conversación”. Una HU básicamente contiene cuatro características, una descripción corta de la funcionalidad a desarrollar, un tamaño relativo, un valor de negocio que define qué tan importante es esta HU para el negocio y uno o más criterios de validación para definir si está terminada. La manera de documentar los requerimientos de un pro-

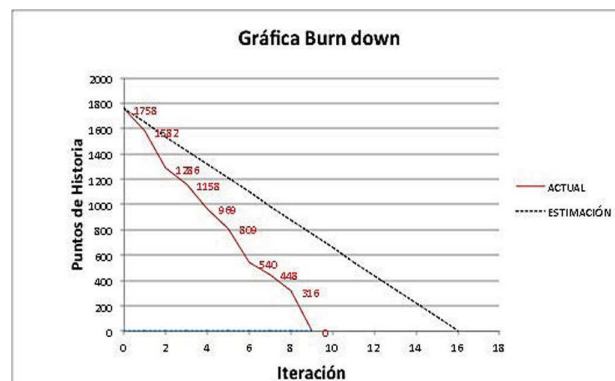


Figura 16. Gráfica burn down para mostrar los SPs restantes para el desarrollo

ducto en métodos ágiles es utilizar historias de usuario. Cohn (2004) contiene más información.

Para definir el **tamaño de una HU** la métrica que se utiliza son los **puntos de historia (SP, story points)** (Cohn, 2004), aunque también se utilizan otras métricas como los días ideales (Cohn, 2005), las horas de ingeniería perfecta y los puntos de función (Kang y Choi, 2010). Los **SP** definen qué tan grande es el esfuerzo necesario para desarrollar una funcionalidad del producto comparándola con otra funcionalidad del mismo producto y no qué tan largo es su desarrollo.

En los métodos ágiles se realizan **revisiones periódicas** con el cliente, que toman el nombre de **iteración** o en inglés **sprint**. Durante cada iteración se realiza una planeación para definir las HUs que serán desarrolladas y entregadas al cliente al terminar la iteración. El desarrollo de un producto está conformado por una o varias liberaciones (*releases*) y cada liberación tiene un conjunto de *n* iteraciones con duraciones de una a cuatro semanas.

Para **monitorear** que los compromisos establecidos en el proyecto se cumplan, el desempeño del equipo de desarrollo se mide basándose en el número de SP desarrollados y aceptados por el cliente durante cada iteración, esta métrica se llama **velocidad de la iteración**.

La herramienta más común en los métodos ágiles para monitorear el avance del proyecto son las **gráficas burn down** (figura 16); en esta gráfica se muestra el avance de desarrollo de un equipo durante cada iteración, mostrando los SP faltantes para su desarrollo, en donde el eje X presenta las iteraciones y el eje Y la cantidad de los SP.

## Agradecimientos

Este es un proyecto apoyado por el Consejo de Nacional de Ciencia y Tecnología (Conacyt) a través del proyecto: “Plataforma de colaboración para cadenas productivas en micro, pequeñas y medianas empresas

bajo el modelo de *software* como servicio” con número de proyecto 000000000181172.

## Referencias

- Anderson D.J. *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*, Coad Series, Prentice Hall Computer, EUA, 2003.
- Agarwal N., Rathod U. Defining Success for Software Projects: An Exploratory Revelation. *International Journal of Project Management*, volumen 24 (número 4), 2006: 358-370.
- Chow T., Cao D.B. A Survey Study of Critical Success Factors in Agile Software Projects. *Journal of Systems and Software*, volumen 81 (número 6), junio de 2008: 961-971.
- Cohn M. *Agile Estimating and Planning*, Prentice Hall, EUA, 2005.
- Cohn M. *User Stories Applied: For Agile Software Development*, Addison-Wesley Professional, EUA, 2004.
- Deemer P., Benefield G., Larman C. *The SCRUM Primer*, Scrum Training Institute, EUA, 2010, pp. 1-22.
- Jeffries R., Anderson A., Hendrickson C. *Extreme Programming Installed*, 2000. The XP Series.
- Jones C. Software Project Management Practices: Failure Versus Success. *The Journal of Defense Software*, (número octubre), 2004: 5-9.
- Kang S. y Choi O. Model-Based Dynamic Cost Estimation and Tracking Method for Agile Software Development, en: Science (ICIS), 2010 IEEE/ACIS 9th, agosto de 2010, pp. 743-748.
- Keaveney S. y Conboy K. Cost Estimation in Agile Development Projects, en: Proc. 14th European Conf. Information, 2006, pp. 1-15.
- Laanti M. y Kettunen P. Cost Modeling Agile Software Development. *International Transactions on Systems Science and Applications*, volumen 1 (número 2), 2006: 175-179.
- Miranda E. y Bourque P. Agile Monitoring Using the Line of Balance. *Journal of Systems and Software*, volumen 83 (número 7), julio de 2010: 1205-1215.
- Pham A. y Pham P. *Scrum in Action: Agile Software Project Management and Development*, Cengage Learning Center, Boston, ESUA 2011, pp. 17-31.
- Project Management Institute. Guide to the Project Management Body of Knowledge - PMBOK® Guide 2003 Edition, 2003.
- Rawsthorne D. Monitoring Scrum Projects with AgileEVM and Earned Business Value (EBV) Metrics. *Agile Journal*, volumen 0, 2008.
- Rusk J. Earned Value for Agile Development. *Software Tech News*, volumen 12 (número 1), 2009: 20-27.
- Sulaiman T. y Barton B. AgileEVM-Earned Value Management in Scrum Projects, en: Agile Conference, 2006.
- Yap M. Value Based Extreme Programming, en: Agile Conference, 2006.

## Este artículo se cita:

### Citación estilo Chicago

Mitre-Hernández, Hugo A., Edgar Ortega-Martínez, Cuauhtémoc Lemus-Olalde. Estimación y control de costos en métodos ágiles para desarrollo de *software*: un caso de estudio. *Ingeniería Investigación y Tecnología*, XV, 03 (2014): 403-418.

### Citación estilo ISO 690

Mitre-Hernández H.A., Ortega-Martínez E., Lemus-Olalde C. Estimación y control de costos en métodos ágiles para desarrollo de *software*: un caso de estudio. *Ingeniería Investigación y Tecnología*, volumen XV (número 3), julio-septiembre 2014: 403-418.

## Semblanza de los autores

**Hugo A. Mitre-Hernández.** Investigador en ingeniería del *software* en CIMAT, Unidad Zacatecas, México, afiliado en el grupo SEL-UC3M (*Software Engineering Lab*) de la Universidad Carlos III de Madrid y pertenece al Sistema Nacional de Investigadores (SNI) de México. En el 2010, finalizó su doctorado del Ministerio Español de Ciencia e Innovación (MICINN), realizó una estancia en el Centro de Investigación Fraunhofer USA (Maryland), para después obtener el grado de doctor en ciencia y tecnología informática de la Universidad Carlos III de Madrid. Sus áreas de interés son: gestión del conocimiento, medición de productos y procesos, gestión estratégica para organizaciones de ingeniería del *Software*, Gobierno de las TICs y gestión de Clusters.

**Edgar Ortega-Martínez.** Es desarrollador de *software* en Compulogic S.A. de C.V., además es SCRUM master certificado por la Scrum Alliance. Recién obtuvo su grado de maestría en ingeniería de *software* en el Centro de Investigación en Matemáticas A.C. Sus intereses son el desarrollo de *software* con métodos ágiles y la mejora de procesos de desarrollo de *software*.

**Cuauhtémoc Lemus-Olalde.** Ingeniero en sistemas computacionales por el Instituto Tecnológico y de Estudios Superior de Monterrey, Campus Monterrey (1986). Ahí estudió la maestría en ciencias computacionales (1988), para luego obtener doctorado en la misma materia por la Universidad de Tulane, Nueva Orleans (1996). En la Universidad de Houston-Clear Lake, realizó un postdoctorado en calidad de investigador visitante como parte del programa aeroespacial del Instituto para operaciones de sistemas espaciales, en coordinación con la NASA y el Centro Espacial Johnson. Es director del Centro de Investigación en Matemáticas, A.C. (CIMAT), Unidad Zacatecas. Sus intereses de investigación son: mejora de procesos de *Software*, medición de procesos (6 sigma), gestión de la innovación y métodos de innovación TRIZ aplicados a la industria.