

ASSE 2016, 17° Simposio Argentino de Ingeniería en Software

# Uso de Opciones Reales para evaluar la contribución de metodologías KANBAN en desarrollo de software

Pedro E. Colla

Maestría en Sistemas Embebidos  
Instituto Universitario Aeronáutico  
Av. Fuerza Aérea Km 8 ½  
(5000) Córdoba – Argentina  
pcolla@iua.edu.ar

**Abstract.** Este artículo aborda el estudio preliminar de los factores que aportan valor al utilizar una metodología ágil Kanban aplicada al desarrollo de software, en particular en el desarrollo de software asociado a sistemas embebidos. La adopción de esta metodología se hace mayormente en base a expectativas subjetivas e historias de éxito, pero el abordaje formal de la caracterización de los factores que contribuyen al valor de la organización otras que la gestión de la capacidad del proceso y su ciclo de trabajo es aún escaso. Se modelan estos factores, se estiman los principales parámetros del modelo y se valida el resultado por medio de simulación. Finalmente se obtienen conclusiones y avizoran líneas de trabajo futuro.

## 1 Antecedentes

Tradicionalmente el software ha sido un instrumento para que las organizaciones mejoraran su productividad interna a partir de acciones de automatización. Sin embargo, este enfoque está cambiando en la medida que el software es utilizado como plataforma para mejorar, o incluso producir, los ingresos y por lo tanto estar sujeto a constantes presiones competitivas para producir soluciones innovadoras en tiempos cortos. Este es un contexto muy volátil donde la metodología de desarrollo debe permitir cambios rápidos. El enfoque “*ligero*” (“*lean*” en la bibliografía de habla inglesa) y sus conceptos se han vuelto crecientemente populares para el desarrollo de software. El objetivo es poder entregar valor a los clientes en forma más efectiva y eficiente a partir de utilizar procesos que encuentren y eliminen el retrabajo, asimilado a desperdicio, que es el principal componente del deterioro de la calidad y la productividad en el desarrollo encontrado por las organizaciones (Liker J.K., 2008). En el ámbito de manufactura, donde se origina, fue desarrollado y aplicado en Toyota donde se lo denominó originalmente el *Toyota Production System* (TPS). El TPS fue establecido basado en dos conceptos (Taiichi O., 1988), el de “*automatización pero con intervención humana*”, implicando que cuando ocurre un problema el proceso debe detenerse impidiendo que el producto defectuoso pueda ser producido. El segundo concepto se denomina “*just-in-time*” (justo a tiempo) que significa que se produce solamente lo que necesita el siguiente paso del proceso en un esquema de flujo continuo. Kanban es un sub-sistema del TPS creado originalmente para controlar los niveles de inventario y el flujo de insumos al proceso productivo (Kumar C.S., 2007) (Ahmad M. O., 2013). Kanban ha trascendido su utilización en manufactura y es extensivamente utilizado en el desarrollo de software y servicios en muchos dominios industriales (Vallet, 2015). Kanban facilita la obtención de altos volúmenes de producción, alta utilización de la capacidad disponible, mejoras substanciales en el ciclo de trabajo y reducir colas de trabajo en proceso (WIP por sus siglas en inglés *work-in-progress*). Kanban se utiliza en el desarrollo de software desde el año 2004 cuando David Anderson introdujo la práctica mientras trabajaba en Microsoft (Anderson, 2010). Kanban permite también para el desarrollo de software visualizar el trabajo, limitar el trabajo en progreso e identificar restricciones del proceso para que se alcancen los objetivos de producción mientras que al mismo tiempo mantiene su foco en ítems individuales en cada momento (Wang X., 2012). En general, Kanban intenta proveer visibilidad sobre el trabajo en progreso y al hacerlo mejorar las comunicaciones, cooperación entre equipos e integración entre desarrolladores de software, probadores y

ASSE 2016, 17° Simposio Argentino de Ingeniería en Software  
equipos de soporte. En desarrollo de software el objetivo de implementar Kanban es poder visualizar y mejorar el flujo de valor mediante optimizar el ciclo de trabajo limitando al mismo tiempo el trabajo en progreso (Burrows M., 2014) (Al-Baik O., 2014). En desarrollo de software Kanban utiliza tarjetas (o equivalentes virtuales) para representar ítems bajo trabajo las que se colocan sobre un plano basado en la información dada por los sistemas de gestión técnica y de proyecto de software. Las prácticas fomentadas por la utilización de Kanban aportan metodologías maduras que incluso pueden asimilarse a la satisfacción de requisitos como los propuestos como objetivos a satisfacer en el modelo de referencia SEI-CMMI™ (Fritzche & Keil, 2007) (Glazer, et al., 2008) (Hurtado Alegría & Bastarrica, 2006). La metodología Kanban puede implementarse de muchas formas, pero usualmente se segmenta el ciclo de desarrollo en una primera etapa donde están los trabajos estimados y en espera, una última etapa donde se colocan al finalizar el ciclo y varias etapas intermedias. Una posible representación es mostrada esquemáticamente por la Figura 1



**Figura 1 Descripción conceptual de un muro Kanban**

La totalidad de los requerimientos a desarrollar, denominados *ítems de trabajo* (“work items”) son gestionados durante su ciclo de vida en función de su prioridad relativa para luego ser implementados en ciclos de esfuerzos relativamente cortos. Los miembros del equipo, en función de su disponibilidad y competencias son los responsables de mover ítems individuales de un estado al siguiente. Ante cualquier bloqueo (no representado en la figura) el ítem es colocado en un área especial de “en problemas” por lo que recibe inmediata visibilidad y atención. En este enfoque se siguen los principales criterios del *Manifiesto Ágil* (Manifiesto, s.f.) obteniendo liberaciones de los componentes incrementales del producto bajo desarrollo. Si bien la bibliografía muestra reportes consistentes de éxito en la aplicación de metodologías ágiles en general, y Kanban en particular, para la solución de proyectos de diferentes tamaños y grados de complejidad el principal marco teórico para justificar la metodología Kanban se basa en la denominada “*Ley de Little*” (Little J. D., 2008) donde se aborda el comportamiento sistémico vinculando la *capacidad de flujo del proceso* ( $\lambda$ ) con el *tiempo de proceso* o *ciclo de desarrollo* ( $t_r$ ) y el *WIP* (W) en un momento dado, estas variables están relacionadas entre sí por la ecuación:

$$WIP = \lambda t_r$$

**Ec. 1**

Es claro que la organización obtiene beneficios incrementando la capacidad de procesar requerimientos y reduciendo el ciclo de trabajo a través de la gestión del número simultáneo de ítems en progreso, lo que constituye uno de los principales objetivos de la metodología.

Ese enfoque, no obstante, centra su atención en lo que la metodología produce al sistema en su conjunto, pero no explora las contribuciones de valor que produce en el proceso de los ítems individuales que procesa. Pero al mismo tiempo el alto grado de visibilidad sobre los ítems bajo desarrollo, la inmediata atención sobre bloqueos o problemas y el foco en abordar los ítems que aportan más valor permiten a Kanban aportar elementos de decisión y gestión diferenciales. Es por lo tanto de interés estimar la magnitud de la contribución al valor de los ítems individuales. Si la capacidad de alterar el resultado de un proceso mediante acciones de gestión aporta valor en sí mismo; el instrumento denominado “*opciones reales*” es usualmente utilizado para estimar éste valor. Por lo tanto identificando las opciones generadas por el proceso utilizado es posible aislar el grado de valor adicional que estas permiten obtener. Este valor estará

asociado a la metodología misma y no solamente a la gestión de como fluyen en el sistema tal como es posible medir mediante la Ley de Little.

La contribución de éste artículo consiste en estudiar que magnitud de contribución al valor del proyecto realiza la visibilidad adicional y la gestión focalizada en los ítems de trabajo, proponiéndose mecanismos para poder medirlo.

La utilización del método Kanban es particularmente apropiado para el desarrollo de software como el que es habitualmente involucrado en el desarrollo de sistemas embebidos. Estos sistemas se caracterizan por requerir ciclos de producción muy cortos, típicamente asociado a estrategias de llegada inmediata al mercado, integración compleja entre software y hardware así como la necesidad de canalizar gran cantidad de requerimientos no necesariamente acoplados entre sí en el menor tiempo posible.

Dado que la creación de valor ocurre por la contribución de los aportes de muchos requerimientos independientes y aislados se propone validar la propuesta mediante simulación a partir de un modelo simple del ciclo de desarrollo y la utilización de valores tentativos para los principales parámetros involucrados basado en métricas organizacionales típicas para de esa forma obtener la magnitud preliminar de los resultados esperables.

## 2 Planteo de la propuesta

Para la determinación del valor creado (o destruido) por una determinada metodología de desarrollo de software es necesario identificar en primer término como medir tal valor y a continuación definir diferentes escenarios posibles que expongan las características a explorar. Si bien el valor obtenido a partir de los procesos utilizados por una organización desarrolladora de software es usualmente conformado por la suma de factores tangibles e intangibles se comienza por los primeros para a continuación establecer pautas consistentes que consideren los segundos.

Como factores intangibles de una organización, y en particular si es pequeña o mediana, resultará esencial considerar factores tales como la incertidumbre en la que operan, la fortaleza financiera, la capacidad de gestión, la volatilidad de los requerimientos y el riesgo técnico inherente a cualquier desarrollo tecnológico en general y de software en particular. Este enfoque permitirá considerar que los factores intangibles contribuirán a mejorar o deteriorar el perfil de riesgo de una organización.

Es usual comenzar por obtener la valuación dada mediante el *Valor Presente Neto* (NPV por sus siglas en inglés Net Present Value) del proyecto como un abordaje que le es familiar a la mayoría de las organizaciones para la determinación del valor de un proyecto y reconocido por considerar al mismo tiempo los resultados económicos tangibles, los premios al capital por tiempo y el impacto del riesgo<sup>(Brealey R.A., 2010)</sup>. Sin embargo, la valuación por NPV es a menudo criticada por asumir una visión estática de la ejecución de un proyecto y por lo tanto no capturar los comportamientos dinámicos que estos tienen y los efectos que tienen las decisiones que se toman en su respuesta para optimizar el resultado mediante la gestión activa de los factores involucrados.

Para abordar esta flaqueza se propone que el modelo incorpore a las estimaciones de valor obtenidas mediante los métodos tradicionales de NPV la valuación por opciones reales creadas durante la ejecución como una forma de capturar la capacidad de la organización para gestionar mejores soluciones en un marco de riesgo e incertidumbre en los resultados<sup>(Erdogmus, et al., 2005)</sup>.

Este abordaje no es totalmente novedoso, otros autores lo han realizado para metodologías ágiles como XP o *Pair Programming*<sup>(Erdogmus & Favaro, 2002)</sup>; incluso el producto anterior de éste esfuerzo de investigación pudo comprobar que las metodologías SCRUM también generan valor por el mismo mecanismo<sup>(Colla P.E., 2012)</sup>. Pero la misión permanente de los desarrolladores de software es alcanzar mejores resultados más rápidamente y Kanban facilita esos objetivos por lo que ha ganado momento en los últimos tiempos, principalmente por su sencilla y elegante aplicación del pensamiento “lean”. Sin embargo solo unos pocos estudios empíricos investigan la dinámica e impactos de Kanban en los proyectos y sus resultados. La implicancia clave es que el valor reside en su simplicidad para motivar a los desarrolladores a observar y controlar las actividades de proyecto<sup>(Ikonen M., 2011)</sup>.

### 2.1 Valuación de un proyecto de software

Si bien el enfoque tradicional de gestión del valor se basa en identificar los equilibrios sistémicos de cuatro variables fundamentales; recursos, tiempo, funciones y calidad que proporcionan el valor deseado.

A los efectos financieros son los *recursos* y el *tiempo* las variables cuya dinámica definen el valor resultante. Los recursos terminan re-expresándose mediante el costo de procesar un dado ítem de trabajo y su variación contribuye a crear o destruir valor del proyecto según disminuya o crezca respecto del planeamiento financiero que lo sustenta.

Por su parte el tiempo contribuye a aportar o deteriorar valor al hacer los flujos de fondos más cercanos o lejanos en el tiempo. Si partimos del supuesto que el proyecto de software típicamente habilita un activo subyacente que de él depende, entonces podemos también asumir que los flujos ingresantes asociados al resultante del proyecto no estarán disponibles hasta que se finalice la construcción del ítem y pueda ser utilizado para producir resultados para la organización, por lo tanto cualquier retraso deteriora el perfil financiero deseado justamente al retrasar la realización de éstos beneficios.

Los aportes de valor y riesgos derivados de factores funcionales y de calidad, las otras variables sistémicas que definen la naturaleza del desarrollo, se asumen capturadas mediante las características intrínsecas de los flujos de fondos, la duración del proyecto y el costo de oportunidad utilizado para descontar los flujos. En el caso de pequeñas y medianas empresas existirán un número apreciable de otros factores tales como dificultades de financiación, problemas de escala, retención de talento, acceso a mercados y capacidad de inversión que también se consideran incorporadas como incertidumbres o riesgos en la tasa de corte a la que descuenta sus flujos de fondos (Brealey R.A., 2010).

En forma simplificada podremos expresar el Valor Presente Neto Planeado (NPV) para un dado ítem de trabajo como la relación entre el costo para implementarlo y los ingresos que producirá según es indicado por la (Ec. 2)

$$NPV = \frac{I}{(1+r)^{t_r}} - \sum_{t=0}^{t=t_r} \frac{C_t}{(1+r)^t}$$

#### Ec. 2

En esta expresión los flujos de ingresos futuros están representados por el valor presente de los mismos al momento de la liberación (I) y los de *egresos* (C<sub>t</sub>) son descontados al *costo de oportunidad* (r) sobre el *plazo de liberación o release un determinado ítem de trabajo* (t<sub>r</sub>), se asume que salvo contadas excepciones las organizaciones abordarán con prioridad aquellos ítems que satisfagan la condición NPV>0.

Esta expresión permite ver que cuando el costo o el tiempo se incrementan el valor del proyecto se deteriora a ingresos constantes. Este factor claramente explica el foco en el tiempo (cycle time) que se tarda en implementar un determinado ítem de trabajo o requerimiento.

En rigor, dos organizaciones que operan a distinto nivel de riesgo deberían descontar sus proyecciones de valor a costos de oportunidad diferentes reflejando este hecho.

## 2.2 Valuación por opciones reales

Las opciones son instrumentos de evaluación financiera que reflejan el valor inherente a la capacidad de operar en un contexto de incertidumbre y cambiar los resultados de un determinado proyecto mediante decisiones de gestión activa sobre él. Las opciones son típicamente caracterizadas por su *valor inicial* (spot price), su *valor de ejercicio* (strike price), el *tipo de opción*, el *término* sobre el cual tiene validez y cuando son ejercitables *solo al final* (europea) o en *cualquier momento* de su término (americanas) entre otras.

Las opciones son usualmente definidas según la característica de la flexibilidad otorgada por su ejecución; típicamente serán relevantes para los procesos de construcción de software las opciones que nos permitan abandonar un proyecto, crecerlo, reducirlo, contratar o realizar elecciones de gestión en general. Las opciones reales tienen valor cuando hay incertidumbre en los resultados, esta incertidumbre afecta los resultados, existe cierta flexibilidad para gestionar los factores bajo incertidumbre a partir de acciones posibles y existe un ejercicio racional de las decisiones. En el análisis realizado en este artículo será de interés concentrar la atención en el *strike price* el que representará el valor neto para la organización de las opciones modeladas.

Es importante diferenciar en el análisis de opciones la noción de riesgo, el cual consiste en la distribución de probabilidad de los distintos resultados posibles de la noción de incertidumbre que se trata de la falta de confianza en obtener los resultados (Hung & So, 2010).

Cuando la incertidumbre se elimina del proyecto la valuación de las opciones tiende a cero y el valor global del proyecto tiende al pronosticado por métodos como el NPV previamente explorado. Black y Scholles (Black & Scholes, 1973) en su ya clásica formulación fundacional del campo propusieron la forma de

evaluar el valor entregado por una opción financiera pero requiere un proceso continuo de decisión que no siempre es válido asumir en desarrollo de software.

En tal caso se considera más adecuado el uso de métodos discretos de cálculo como el ofrecido por el *método de Lattice* (Mun, 2002) con una estructura como la mostrada en la Figura 2.

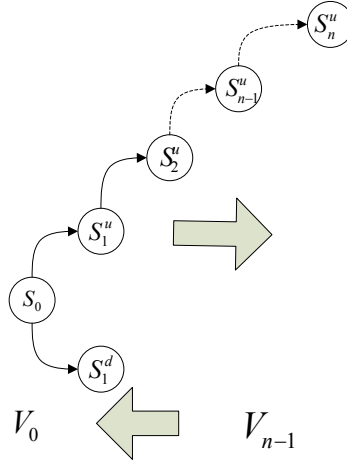


Figura 2 Arbol binario de decisión (Lattice)

Mediante su uso se calculará en forma recursiva el valor de la opción que será en definitiva el valor total aportado al proyecto mediante la gestión a partir de los mecanismos de opciones estudiados. Para su cálculo se empieza por trazar un árbol binario de decisión donde en cada nodo es posible tomar decisiones o ejercer opciones. En el caso de Kanban estos hitos se corresponden con los momentos donde se revisa el estado de los ítems de trabajo (por ejemplo frecuencia diaria). En cada nodo del árbol binario de decisión se calcula el valor ascendente (u) y el valor descendente (d) mediante (Ec. 3, Ec. 4, Ec. 5, Ec. 6 y Ec. 7):

$$S_{i+1}^u = u \cdot S_i \quad S_{i+1}^d = d \cdot S_i \quad u = e^{\sigma\sqrt{\delta t}} \quad d = e^{-\sigma\sqrt{\delta t}} = \frac{1}{u} \quad p = \frac{e^{r_f\delta t} - d}{u - d}$$

Ec. 3                      Ec. 4                      Ec. 5                      Ec. 6                      Ec. 7

Donde el *periodo considerado* ( $\delta t$ ) es la escala de tiempo discreta del ejercicio, la *tasa de retorno sin riesgo* ( $r_f$ ) el componente del premio por tiempo en el costo de oportunidad y la *volatilidad del proceso* ( $\sigma$ ) una magnitud adimensional que permite introducir el grado de incertidumbre en el que el opera el proyecto y cuya magnitud se estimará posteriormente.

$$V_{n-1} = S_n - S_0 \quad \forall S_n - S_0 \geq 0 \quad V_{n-1} = 0 \quad \forall S_n - S_0 < 0$$

Ec. 8

Una vez finalizado el cálculo del valor de todos los nodos finales se produce la *retropropagación* de los mismos mediante el cálculo recursivo (Ec. 8) para obtener el valor residual de la opción en cada nodo de decisión ( $V_i$ ).

Los restantes nodos culminando en el raíz ( $i=0$ ) se calculan mediante (Ec. 9)

$$V_{i-1} = (pV_i^u + (1-p)V_i^d)e^{-r_f\delta t}$$

Ec. 9

El valor de la opción será el que se encuentre al nodo raíz del árbol binario al finalizar el cálculo ( $V_0$ ), el que recibirá el nombre de *Option Price Value* (OPV) y representa el valor creado por las acciones de gestión realizadas durante el transcurso del desarrollo.

La metodología Kanban permite plantear los principales ingredientes de la valuación por opciones. El trabajo es en principio segmentado en ítems de trabajo que son priorizados y seguidos durante su ciclo de vida tomado decisiones sobre su progreso y cualquier factor que lo retrase. De tal manera que al hacerlo se crean opciones de *diferimiento*. Por otra parte se despliega un método riguroso mediante el cual entre ítems de trabajo simultáneos se puede reflexionar sobre el valor de continuar en una determinada dirección, alterar las pautas de diferimiento o cancelar determinados ítems de trabajo (opción de *diferimien-*

ASSE 2016, 17° Simposio Argentino de Ingeniería en Software  
*to/abandono*). Se asume legítimo considerar el valor de las opciones creadas por la metodología empleada como parte del valor global del proyecto

### 3 Modelado

A modo de validación del planteo se analizarán los resultados esperables por una organización ejecutando un flujo de requerimientos de tamaño y complejidad que se consideran típicos en una operación real mediante la utilización de un enfoque Kanban. Se asume que la organización se enfoca en mejorar su capacidad de producción y reducir el ciclo de implementación.

El valor obtenido no depende en principio de la “madurez” organizativa, según resulta medida por estándares de calidad, sino de su habilidad de mover en forma óptima un ítem de trabajo a través de un proceso definido.

Los costos y los ingresos de un determinado ítem de trabajo son en principio arbitrarios y dependientes de las características de cada requerimiento procesado.

Se modelarán los requerimientos como un conjunto de requerimientos pendientes, cada requerimiento será evaluado respecto al esfuerzo ( $E$ ) necesario para implementarlo, el esfuerzo total será entonces el involucrado en desarrollarlo más el necesario para evaluarlo ( $E_{plan}$ ) según la ecuación (Ec. 10), es natural esperar que las organizaciones tenderán a reducir el esfuerzo de planeamiento mediante la utilización de técnicas de estimación rápida.

$$E_{tot} = E + E_{plan}$$

Ec. 10

Cada etapa del proceso insumirá un esfuerzo que resulta de una proporción del esfuerzo total, proporción que dependerá de factores como tamaño o complejidad del ítem de trabajo, pero que en principio las organizaciones podrán caracterizar dentro de ciertos límites. El esfuerzo total será entonces típicamente (Ec. 11)

$$E_{tot} = E_{plan} + E_{req} + E_{descut} + E_{vv} + E_{rel}$$

Ec. 11

Donde los esfuerzos intervinientes se corresponden con las etapas típicas de Kanban, es decir requerimientos ( $E_{req}$ ), diseño, desarrollo y test unitario ( $E_{descut}$ ), validación y verificación ( $E_{vv}$ ) y liberación ( $E_{rel}$ ). El tiempo que insume cada etapa estará relacionada con la cantidad de horas que se le dedique en un período determinado, se puede evaluar éste factor como una productividad de tiempo ( $p_i$ ) dada por el tiempo efectivo dedicado por los miembros de la organización al esfuerzo de desarrollo, aspecto que naturalmente las organizaciones tratan de maximizar. Como nos enfocamos en el aporte de valor por las acciones de gestión sobre los ítems individuales no se le asigna importancia en una primera aproximación a la prioridad relativa de ese ítem respecto del resto que se están desarrollando, en otras palabras la prioridad definirá cuando el ítem entra al sistema, pero cuando lo haga se obtendrá el valor adicional modelado. Finalmente, existirán problemas durante el desarrollo, los que inducirán a la colocación del ítem en una zona especial de “ítem bloqueado” o “ítem con problema” de la cual usualmente se retirará luego de la realización de algún retrabajo. Este aspecto puede ser modelado mediante la introducción del *Costo de Calidad Pobre* (CoPQ por sus siglas en inglés Cost of Poor Quality) el que es típicamente un parámetro organizacional sujeto a gestión y optimización. El esfuerzo será entonces dado por (Ec. 12)

$$E_{tot} = E_{plan} + [(1 + CoPQ)(E_{req} + E_{descut} + E_{vv} + E_{rel})]$$

Ec. 12

Esta formulación proviene de la mera observación de cómo se agregan los esfuerzos en el sistema. El costo de implementar el ítem de trabajo normalmente estará mayormente determinado por el esfuerzo que insumirá y el costo promedio de los recursos necesarios para ejecutar ese esfuerzo, las organizaciones suelen medir el mismo como *costo por ingeniero* (CPE por sus siglas en inglés *cost per engineer*). Por lo tanto el costo total de un ítem de trabajo estará dado por (Ec. 13)

$$C_{tot} = CPE \times E_{tot}$$

Ec. 13

Es de esperarse que el ítem de trabajo representa una inversión, sobre la que se espera un rendimiento económico de alguna índole; en algunos casos se implementarán funciones sin tal rendimiento económico

pero que deban implementarse de todas formas por otros motivos tales como requerimientos mandatorios legales. Sin embargo se puede asociar normalmente el ingreso que generará un ítem de trabajo y su costo mediante la relación denominada *retorno de la inversión* (ROI por sus siglas en inglés return of investment); por lo tanto el ingreso podrá ser modelado como lo indica la (Ec. 14):

$$I = ROI \times C_{tot}$$

**Ec. 14**

Este ingreso ocurrirá a la finalización de la implementación o *tiempo de liberación* ( $t_i$ ) por lo tanto para obtener el valor transferido al comienzo del proyecto es necesario descontar el mismo usando el costo de oportunidad de la organización ( $r$ ) por lo que finalmente el valor a utilizar para la evaluación de la opción real será dado por la (Ec. 2):

$$S_0 = NPV$$

Con los elementos necesarios para obtener el valor de la opción usando el método Lattice previamente descrito es posible obtener también que proporción ésta representa del valor total del proyecto mediante las ecuaciones (Ec. 2) (Ec. 9) obteniendo (Ec. 15)

$$\kappa = \frac{OPV}{NPV}$$

**Ec. 15**

Esta relación de *valor agregado por la capacidad de gestión* ( $\kappa$ ) y medido asimilando éstas a opciones reales determinará cuanto valor agrega la metodología en función de las facilidades de visibilidad y gestión que provee.

## 4 Ejecución del modelo

El modelo es validado mediante una simulación por el método de MonteCarlo del cual se obtendrán probables distribuciones de los resultados. Se utilizan 20000 intentos, valor que se obtiene por prueba y error como un razonable equilibrio entre el tiempo de proceso y la exploración de suficientes alternativas para confiar en obtener un resultado significativo. Los valores utilizados como parámetros del modelo pueden verse en la Tabla 1 Valores Típicos Utilizados a continuación.

Variable de Simulación	Sím	UM	Min	Mediana	Max
Tasa anual de Rentabilidad	r	%	5,00%	10,00%	20,00%
Esfuerzo [CR]	E	Horas	12,00	40,00	600,00
Esfuerzo de Estimación	E <sub>est</sub>	Horas	1,00	4,00	16,00
Costo de Calidad Pobre	CoPQ	%	5,00%	10,00%	40,00%
Productividad Semanal	$\pi$	horas	35,00	40,00	44,00
Costo del Ingeniero	CPE	US\$/Hr	\$ 17,00	\$ 19,00	\$ 20,00
Retorno de Inversión	ROI	--	1,00	2,00	3,00
Tasa Libre de Riesgo [Anual]	r <sub>f</sub>	%	1,000%	2,000%	5,000%

Se utiliza como volatilidad media  $\sigma=0,26$ , como productividad semanal una distribución uniforme entre 35 y 44 horas por semana.

**Tabla 1 Valores Típicos Utilizados**

Los valores utilizados se obtienen a partir de datos históricos relevados de experiencia profesional, en los casos que no se puede asignar una distribución determinada se aproxima el modelado estocástico mediante una distribución triangular (Sargent, 1982.)

Estos valores pueden corresponder a un determinado contexto de desarrollo, pero cada organización podrá utilizar sus propios valores históricos para realizar la determinación del valor que recibe por el uso de Kanban.

Un elemento crucial necesario para la ejecución del modelo es estimar la *volatilidad media* ( $\sigma$ ), la cual es estimada mediante el *método del punto fijo* (Mun, 2002) a partir de considerar la variación de los valores netos obtenidos entre cinco simulaciones sucesivas de escenarios relacionados con los mismos valores de ejecución anteriores, esta aproximación preliminar da como resultado  $\sigma=0.26$ .

Los resultados de la ejecución del modelo serán entonces distribuciones de la variable de interés o sea la magnitud del valor aportado por las opciones al proyecto ( $\kappa$ ).



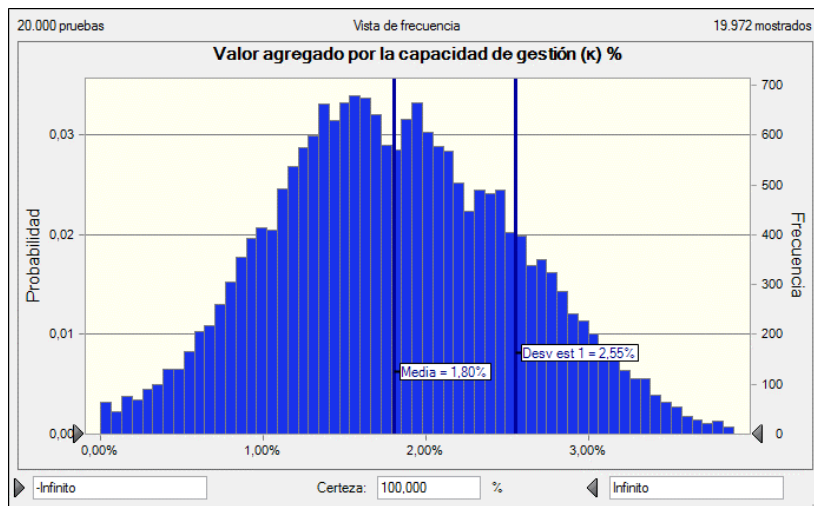


Figura 3 Distribución de valor adicional obtenido calculando opciones reales

Los resultados muestran una contribución del entorno del 2% del valor total neto del proyecto derivado de las opciones de gestión que la metodología produce medido a través del mecanismo del valor de las opciones que provee para tomar éstas acciones de gestión. Esta cifra puede parecer pequeña, pero en términos de valor neto descontado de riesgo es muy significativa, en particular en contextos donde éste factor es muy superior a la tasa libre de riesgo con la que opera la economía en un momento dado.

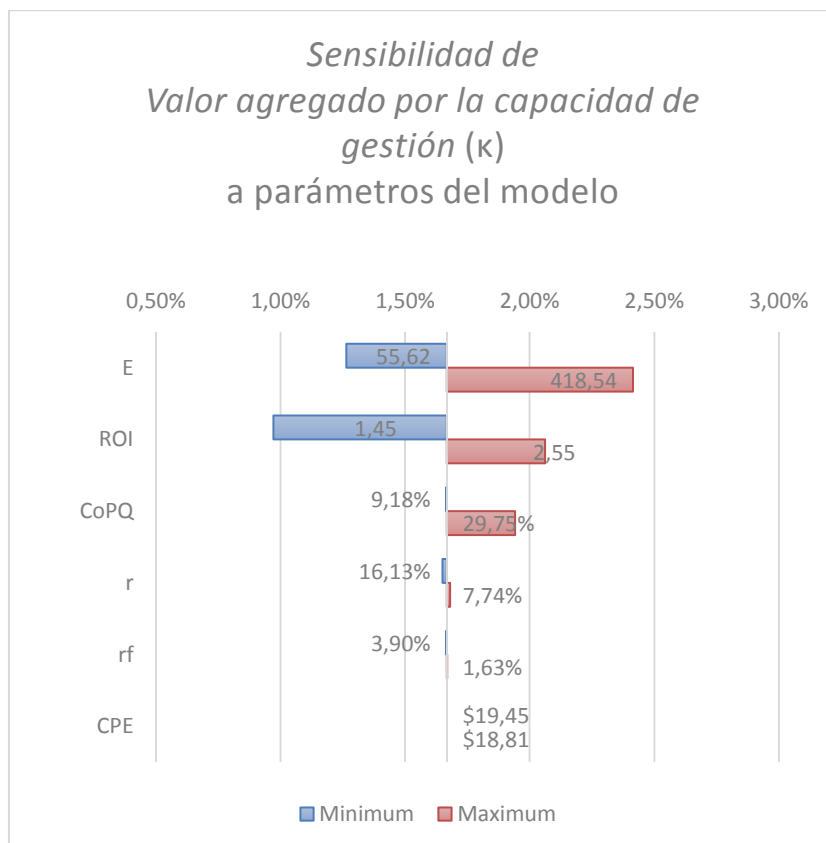


Figura 4 Grafico Tornado influencia de variables independientes

Al estudiarse en más detalle los factores que contribuyen al factor  $\kappa$  mediante un gráfico de tornado (Figura 4) pueden observarse las dependencias sistémicas con las variables independientes del modelo. La principal contribución proviene del tamaño del requerimiento, resultado no tan intuitivo, donde el valor se incrementa significativamente para ítems cuyo esfuerzo es grande. En segundo lugar el resultado es sen-



sible al retorno de la inversión del ítem en cuestión, siendo mayor para retornos atractivos y menor para los que no lo son. En tercer lugar en influencia se encuentra la relación de retrabajo (costo de calidad pobre) involucrada, resultado si razonable toda vez que el retrabajo deteriora significativamente tanto el ciclo de producción como la capacidad del proceso agregando incertidumbre al resultado; las acciones de gestión permiten focalizar problemas de éste tipo y minimizar su impacto. Finalmente el costo de oportunidad, la tasa libre de riesgo y el valor promedio de los recursos no evidencian una significativa contribución al resultado obtenido.

#### 4.1 Amenazas a la validez

Hay varias cuestiones que es necesario tener presentes como amenazas a la validez de los resultados obtenidos. La investigación futura debe ayudar a determinar la medida en que las acciones dentro de cada ítem de trabajo independientemente a la gestión de su flujo durante su proceso, por ejemplo la influencia de acciones de índole arquitectónica o técnica.

Seguramente existirán oportunidades de creación de valor mediante la consideración del acople entre distintos requerimientos y la conveniencia de procesarlos simultáneamente, pero en ésta primera aproximación al modelado del valor aportado por la gestión no se considera éste factor. No se considera en el modelado la opción de priorizar que la metodología inherentemente provee, y que preliminarmente se considera debe proveer valor adicional.

La hipótesis simplificadora que el proceso de decisión puede asimilarse a un árbol binario se considera apropiado para un abordaje inicial del estudio del problema, pero es necesario enfatizar en el análisis sobre la posibilidad de utilizar otras configuraciones posibles de red de decisión.

#### 4.2 Conclusiones

El artículo despliega una fundamentación del valor adicional obtenido por una organización cuando usa metodologías basadas en Kanban por sobre los beneficios de gestión de la cola de trabajo con foco en el ciclo de implementación y la capacidad de proceso del sistema. Este valor adicional parte de la hipótesis, razonablemente sustentada, que al hacerlo obtiene los beneficios de las opciones reales que están implícitas en la flexibilidad de gestión que le metodología permite.

Al utilizar técnicas de validación por simulación utilizando el método de MonteCarlo es posible observar cuales son las magnitudes provenientes de ambos factores que aportan al valor obtenido, así como que estas son distinguibles y diferenciables. Se utilizan datos de campo para obtener resultados cuantitativos preliminares indicativos de las magnitudes que son esperables y se utilizan para un análisis de riesgo utilizando técnicas estocásticas. Para formular el análisis fue necesario obtener valores preliminares de la volatilidad esperable en un proyecto de software cuya exploración en la bibliografía ha sido escasa, siendo un primer paso en el que habrá que perseverar con esfuerzos de refinamiento futuros.

Se identifica una fuerte contribución al valor en el caso de requerimientos e ítems de trabajo relativamente grandes y que requieran esfuerzos y tiempos de desarrollo mayores; conclusión que es en cierta forma sorprendente. También es remarcable la sensibilidad al valor de retorno de la inversión que el ítem provee. Es interesante también comprobar que valores elevados de retrabajo generan incertidumbres que destruyen valor y por lo tanto su gestión contribuye a mitigar éste efecto. No se identifican implicancias significativas desde el costo del dinero (libre de riesgo), el costo de oportunidad y el costo de los recursos involucrados.

Finalmente se concluye que siendo Kanban una metodología de desarrollo fuertemente anclada en la entrega de ítems de trabajo con foco en la gestión de su proceso individual y con una escala de inversión necesaria para su implementación relativamente reducida resultará de atractivo a organizaciones pequeñas induciéndolas al despliegue de prácticas que eleven su performance y obtengan mejoras en su competitividad a partir de ellas.

#### Bibliografía

- Ahmad M. O., M. J. (2013). Kanban in software development: A systematic literature review. *Proceedings of the 39th Euromicro Conference Series on Software Engineering and Advanced Applications*, pp. 9-16.
- Al-Baik O., M. J. (2014). The kanban approach, between agility and leanness: a systematic review. *Empirical Software Engineering*.

- Anderson. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.
- Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy*, *V81*(N3), págs. pp 637-654.
- Brealey R.A., M. S. (2010). *Fundamentos de Financiación Empresarial*. McGraw-Hill. .
- Burrows M. (2014). *Kanban from the inside*. Blue Hole Press.
- Colla P.E. (2012). Marco para evaluar el valor en metodología SCRUM. *13th Argentine Symposium on Software Engineering*, pp. 32-46.
- Erdogmus, H., & Favaro, J. (2002). Keep your options open: Extreme Programming and the Economics of Flexibility. (G. S. Michele Marchesi, Ed.) *NRC Publications Archive, Ch 43*(Extreme Programming Perspective, Addison Wesley), pp 503-552.
- Erdogmus, H., Favaro, J., & Halling, M. (2005). Valuation of Software Initiatives under Uncertainty: Concepts, Issues and Techniques.
- Fritzche, M., & Keil, P. (2007). Agile Methods and CMMI: Compatibility or Conflict? *VI*(N1).
- Glazer, H., Dalton, J., Anderson, D., Konrad, M., & Shrum, S. (2008). *CMMI or Agile: Why not embrace both!* Software Engineering Process Management.
- Hung, M., & So, L. (2010). The Role of Uncertainty in Real Option Analysis.
- Hurtado Alegría, J., & Bastarrica, M. (2006). Implementing CMMI using a Combination of Agile Methods. *V9*(N1).
- Ikonen M., P. E. (2011 ). On the Impact of Kanban on Software Project Work. *16th IEEE International Conference on Engineering of Complex Computer Systems*, pp. 305-314.
- Kumar C.S., P. R. (2007). Literature review of JITKANBAN system. *The International Journal of Advanced Manufacturing Technology*, *32*(3-4), pp. 393-408.
- Liker J.K., H. M. (2008). *Toyota culture: the heart and soul of the Toyota Way*. New York, USA : McGraw-Hill.
- Little J. D., G. S. (2008). Little's law. In *Building Intuition*. Springer, pp. 81-100.
- Manifesto, A. (s.f.). Manifesto for agile software development. Retrieved Feb,2012 <http://www.agilemanifesto.org>.
- Mun, J. (2002). *Real Options Analysis, Tools and Techniques for Valuing Strategic Investment and Decisions*. Hoboken, New Jersey: John Wiley & Sons.
- Sargent, R. (1982.). Verification and Validation of Simulation Models. En *Chapter IX in Progress in Modelling and Simulation* (págs. pp.159-169.). Academic Press London.
- Taiichi O., T. (1988). *Toyota production system: beyond large-scale production*. Productivity press.
- Vallet, B. (2015). *Kanban at Scale – A Siemens Success Story*. Recuperado el 10 de 05 de 2016, de <http://www.infoq.com/articles/kanbansiemens-health-services>
- Wang X., C. K. (2012). “Leagile” software development: An experience report analysis of the application of lean approaches in agile software development. *J. Syst. Software*, vol.85, pp.1287-1299.