

ESTIMACIÓN DE PROYECTOS DE SOFTWARE CON PUNTOS DE CASOS DE USO

Valero Orea, Sergio*

RESUMEN

Uno de los principales problemas a los que nos enfrentamos los desarrolladores de software al momento de planear nuestros proyectos es la estimación. Existen distintas técnicas que nos permiten estimar proyectos de software, cada una de ellas con sus ventajas y desventajas, pero la mayoría de ellas no ofrecen la flexibilidad de estimar software orientado a objetos, y se basan prácticamente en la experiencia del equipo de desarrollo. La técnica de estimación con puntos de caso de uso nos permite realizar estimaciones a partir de modelos orientados a objetos con una precisión bastante aceptable.

INTRODUCCIÓN

El proceso de gestión de proyectos de software comienza con un conjunto de actividades que se denominan *planificación del proyecto*. La primera de estas actividades es la *estimación* [1]. Estimar, o *cuantificar*, software no es una tarea fácil. Las metodologías de desarrollo de sistemas han evolucionado, desde los antiguos sistemas por lotes, la programación estructurada, hasta la orientación a objetos, pero las técnicas de estimación no lo han hecho. Por un lado, se conocen algunas técnicas como COCOMO, Puntos de Función y otras, pero la mayoría de ellas se basan en criterios muy poco efectivos de aplicar como líneas de código, experiencia previa sobre sistemas similares, etc. De esta forma, hemos podido notar un cambio en las metodologías para el desarrollo de software y de la misma manera, existen técnicas adecuadas que nos permiten realizar estimaciones, como la técnica de *puntos de casos de uso*, la cual se basa en metodologías orientadas a objetos, específicamente en el *modelo de casos de uso*. Este trabajo se enfoca en la descripción de esta técnica, la cual es muy fácil de comprender y su aplicación no es muy difícil de llevar a cabo como podremos ver a continuación.

DESARROLLO

Históricamente, los egresados de las Universidades, los profesionales de TI, las empresas, y todo aquel que le han pedido cuantificar los costos y los tiempos estimados de desarrollo de un proyecto de software han sido víctimas de una lucha para la cual no han sido preparados. En las Universidades se les prepara a los futuros profesionistas a programar, modelar, pero muy pocas veces a *planificar*. La planificación de proyectos es una tarea esencial para poder hacer cuantificaciones y las estimaciones adecuadas.

Hoy en día, los desarrolladores de software frecuentemente se apoyan de los casos de uso para describir procesos de negocio en proyectos orientados a objetos [2]. En el

*Universidad Tecnológica de Izúcar de Matamoros

análisis orientado a objetos, los modelos de casos de uso describen los requerimientos funcionales de un sistema de software.

Modelo de casos de uso

El modelo de casos de uso es un artefacto que surge como producto de la fase de requerimientos de la metodología de desarrollo y gestión de proyectos llamada Proceso Unificado de Rational (RUP, por sus siglas en inglés). Dicha metodología, propone una serie de prácticas para el desarrollo de proyectos de software basado en fases, a través de una serie de disciplinas que nos permitirán ir generando artefactos en cada una de las iteraciones. Esta metodología es cíclica, es decir, por cada ciclo se generan documentos entregables (artefactos) que nos permitirán medir el avance de nuestros proyectos, inclusive desde las etapas iniciales.

El modelo de casos de uso es una amplia y aceptada técnica que captura el proceso de negocio y los requerimientos de un proyecto de desarrollo de software. Esta es la base para las estimaciones, ya que contiene la especificación detallada de los actores y casos de uso.

Técnicas de estimación

La estimación del costo y del esfuerzo del software nunca será una ciencia exacta. Son demasiadas variables -humanas, técnicas, de entorno, políticas- que pueden afectar el costo final del software y del esfuerzo aplicado para desarrollarlo. Sin embargo, la estimación del proyecto de software puede dejar de ser un arte oscuro para convertirse en una serie de pasos sistemáticos que proporcionen estimaciones con un grado de riesgo aceptable [1].

Existen distintas técnicas de estimación, de las cuales podemos mencionar brevemente al modelo COCOMO, los puntos de función, y por supuesto, los puntos de casos de uso. Cada una de ellas presenta una serie de ventajas y desventajas, pero su discusión está fuera del alcance de este trabajo y nos enfocaremos a estudiar principalmente los *puntos de casos de uso*.

Puntos de casos de uso

Este método de estimación de proyectos de software fue desarrollado en 1993 por Gustav Karner de Rational Software y está basado en una metodología orientada a objetos, dándole el nombre de “estimación de esfuerzos con casos de uso”. Surgió como una mejora al método de puntos de función pero basando las estimaciones en el modelo de casos de uso, producto del análisis de requerimientos. Según su autor, la funcionalidad vista por el usuario (modelo de casos de uso) es la base para estimar el tamaño del software.

El objetivo de la técnica

Estimar las horas necesarias para ejecutar un conjunto de casos de uso. Es decir, necesitamos predecir cuánto tiempo llevará el desarrollo de software y cuántas personas se requieren para realizarlo. Para ello, es necesario cuantificar la complejidad del sistema y el tiempo necesario para producir una unidad de complejidad.

Al inicio, el método depende de casos de uso bien estructurados y bien escritos, con un nivel conveniente de detalle textual. Al final, se pretende obtener un número único que caracterice completamente al sistema y que se correlacione con la productividad observada del ingeniero.

Tabla 1. VENTAJAS Y DESVENTAJAS DE LA TÉCNICA DE PUNTOS DE CASOS DE USO	
Ventajas	Desventajas
Trabaja bien con diferentes tipos de software	No existe un estándar para escribir casos de uso lo que dificulta la aplicación del método.
Muestra buen rendimiento en proyectos pequeños, medianos y grandes.	Las herramientas en esta área son caras y se enfocan en la evaluación del proyecto

Vale la pena aclarar que un caso de uso por sí solo **no** permite efectuar una estimación de esfuerzos ni de tiempos, solamente son una herramienta para el análisis. La idea central es estimar el tamaño (cuantificar) del software a partir de los *requerimientos de los casos de uso*.

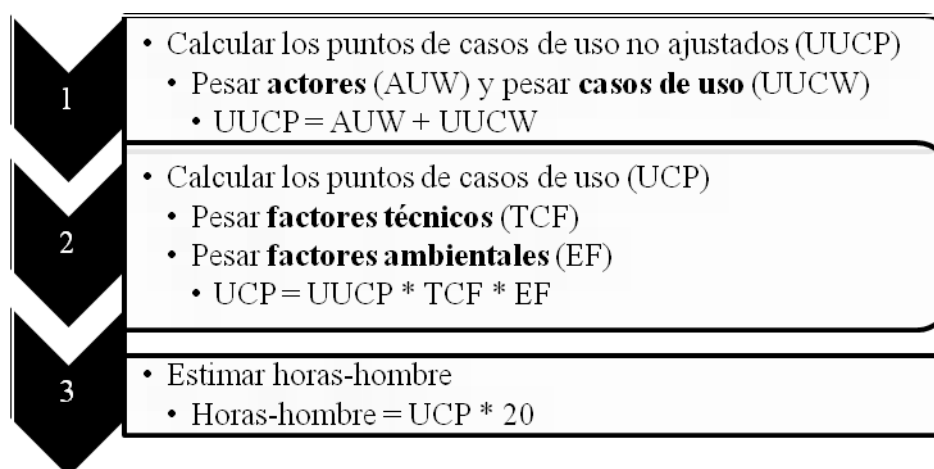


Fig. 1. Pasos para el desarrollo de la técnica de puntos de casos de uso

Desarrollo de la técnica de puntos de casos de uso

Como se puede apreciar en la figura 2, el proceso de desarrollo de la técnica de puntos de casos de uso esta formulado por 3 pasos bien definidos, los cuales se describen a continuación.

1. Calcular los *puntos de casos de uso no ajustados* (UUCP). Para lo cual, hay que definir un peso tanto para actores (AUW) como para casos de uso (UUCW) y sumar el resultado de esos pesos.
 - a. Para determinar el peso de los actores, nos vamos a apoyar de la siguiente tabla para determinar el tipo de actor:

Tabla 2. PESO DE LOS ACTORES		
Tipo de actor	Descripción	Factor
Simple	Otro sistema con una API definida	1
Medio	Otro sistema interactuando con algún protocolo (TCP) o una persona interactuando a través de una interfaz en modo texto	2
Complejo	Una persona interactuando a través de una interfaz gráfica de usuario	3

Finalmente, hay que contar el número de actores que hay en el sistema, multiplicar cada tipo por su factor de peso y sumar esos productos para obtener el total (AUW).

- b. Para determinar el peso de los casos de uso, nos apoyaremos de la siguiente tabla para determinar el tipo de caso de uso:

Tabla 3. PESO DE LOS CASOS DE USO		
Tipo de caso de uso	Descripción	Factor
Simple	3 transacciones o menos	5
Medio	4 a 7 transacciones	10
Complejo	Más de 7 transacciones	15

De la misma forma que con los actores, hay que contar el número de casos de uso que hay en el sistema, determinar su factor, multiplicar cada tipo por su factor de peso y sumar esos productos para obtener el total (UUCW).

2. Calcular los *puntos de casos de uso* (UCP). Para lo cual, hay que realizar el producto del peso de los puntos de casos de uso no ajustados (UUCP), el peso de los factores técnicos (TCF) y el peso de los factores ambientales (EF).
 - a. Para determinar el peso de los factores técnicos, nos apoyaremos de la siguiente tabla, la cual nos indica aspectos relacionados con la complejidad de los módulos a desarrollar:

Tabla 4. FACTORES TÉCNICOS		
Factor	Descripción	Peso
T1	Sistema distribuido	2
T2	Objetivos de performance o tiempo de respuesta	1
T3	Eficiencia del usuario final	1
T4	Procesamiento interno complejo	1
T5	El código debe ser reutilizable	1
T6	Facilidad de instalación	0.5
T7	Facilidad de uso	0.5
T8	Portabilidad	2
T9	Facilidad de cambio	1
T10	Concurrencia	1
T11	Objetivos especiales de seguridad	1
T12	Acceso directo a terceras partes	1
T13	Facilidades especiales de entrenamiento a usuarios	1

Para pesar los factores técnicos (TCF), hay que recorrer la siguiente tabla y estimar cada factor entre cero y cinco (0 y 5), donde

Tabla 5. ESCALAS DE ESTIMACION TCF	
Descripción	Valor
Irrelevante	De 0 a 2
Medio	De 3 a 4
Esencial	5

Posteriormente, hay que multiplicar la estimación de cada factor por su peso y sumar esos productos para obtener el total (*TFactor*). Finalmente, para obtener el peso de los factores técnicos hay que multiplicar el *TFactor* por la fórmula constante propuesta por el creador de la técnica de la siguiente forma:

$$TCF = 0.6 + (0.01 * TFactor)$$

Obteniendo así, el peso de los factores técnicos.

- b. Para determinar el peso de los factores ambientales, nos apoyaremos de la siguiente tabla, la cual nos indica aspectos relacionados con la habilidades y experiencias del grupo o equipo de trabajo:

Tabla 6. FACTORES AMBIENTALES		
Factor	Descripción	Peso
E1	Familiaridad con el modelo del proyecto utilizado	1.5
E2	Experiencia en la aplicación	0.5
E3	Experiencia en orientación a objetos	1
E4	Capacidad del analista líder	0.5
E5	Motivación	1
E6	Estabilidad en los requerimientos	2
E7	Personal de medio tiempo	-1
E8	Dificultad en el lenguaje de programación	-1

De la misma manera que los factores técnicos, para pesar los factores ambientales (EF) hay que recorrer la siguiente tabla y estimar cada factor entre cero y cinco (0 y 5), donde

Tabla 7. ESCALAS DE ESTIMACION EF	
Descripción	Valor
Sin experiencia, sin motivación, estabilidad	De 0 a 2
Promedio	3
Amplia experiencia, motivación, estabilidad	De 3 a 5

Igualmente, hay que multiplicar la estimación de cada factor por su peso y sumar esos productos para obtener el total (*EFactor*). Finalmente, para obtener el peso de los factores ambientales hay que multiplicar el *EFactor* por la fórmula propuesta por el creador de la técnica de la siguiente forma:

$$EF = 1.4 + (-0.03 * EFactor)$$

Obteniendo así, el peso de los factores ambientales.

3. El paso 3 de esta técnica nos indica que debemos estimar las horas hombre mediante la fórmula:

$$HH = UCP * 20$$

El autor de la técnica sugiere usar 20 horas hombre por UCP. Por ejemplo, para un sistema de *60 UCP * 20 hrs/hombre* nos da un total de *1200 hrs/hombre*. Lo que equivale a *30 semanas* (40 hrs por semana para una persona), de esta forma, un *equipo de 5 personas* desarrollarían el sistema en **6 semanas**. En resumen:

Tabla 8. RESUMEN EJEMPLO		
Actividad	Porcentaje de tiempo	Horas-Hombre
Análisis	10%	300
Diseño	20%	600
Programación	40%	1200
Pruebas	15%	450
Sobrecarga	15%	450
TOTAL	100%	3000

Es decir, 75 semanas a 40 hrs por semana para una sola persona o 15 semanas para un equipo de 5 personas de tiempo completo.

Ejemplo completo

Para demostrar los pasos en los que se basa la técnica, realizaremos un caso práctico para el caso de uso “retirar dinero de cajero automático”. El proceso se explica con el fin de lograr una mayor comprensión de la técnica y los pasos asociados.

- *Actor*: cliente.
- *Propósito*: realizar un retiro de una cuenta desde un cajero automático.
- *Visión general*: un cliente llega a un cajero automático, introduce la tarjeta, se identifica y solicita realizar un retiro de dinero. El cajero le da el dinero solicitado tras comprobar que la operación puede realizarse. El cliente toma el dinero y se va.

Curso típico de eventos

- 1.- Este caso de uso empieza cuando un cliente introduce su tarjeta en el cajero.
- 2.- El sistema pide la clave de identificación.
- 3.- El cliente introduce la clave.
- 4.- El sistema presenta las opciones disponibles.
- 5.- El cliente selecciona la operación de Retiro.
- 6.- El sistema pide la cantidad a retirar.
- 7.- El cliente introduce la cantidad requerida.
- 8.- El sistema procesa la petición y, eventualmente, da el dinero solicitado.
- 9.- El sistema devuelve la tarjeta y genera un recibo.
- 10.- El cliente recoge el dinero, el recibo y la tarjeta. Procede a retirarse.

Después de tener el flujo básico de eventos en la especificación de casos de uso, procedemos a calcular los **puntos de casos de uso no ajustados (UUCP)**:

Tabla 9. PESO ACTORES EJEMPLO		
Actores	Tipo	Valor
Cliente	Complejo	3
	UAW	3

Tabla 10. PESO CASOS DE USO EJEMPLO		
Caso de uso	Tipo	Valor
Retirar dinero cajero	Promedio	10
	UUCW	10

$$UUCP = UAW + UUCW = 3 + 10 = 13$$

A continuación, calcularemos los *puntos de caso de uso (UCP)*, primero obtendremos los factores técnicos con la tabla propuesta:

Tabla 11. FACTORES TÉCNICOS EJEMPLO				
Factor	Descripción	Peso	Nivel	Pes*Niv
T1	Sistema distribuido	2	3	6
T2	Performance o tiempos de respuesta	1	5	5
T3	Eficiencia del usuario final	1	1	1
T4	Procesamiento interno complejo	1	1	1
T5	Código reutilizable	1	0	0
T6	Facilidad de instalación	0.5	1	0.5
T7	Facilidad de uso	0.5	2	1
T8	Portabilidad	2	0	0
T9	Facilidad de cambio	1	1	1
T10	Concurrencia	1	5	5
T11	Seguridad	1	5	5
T12	Acceso directo a terceras partes	1	0	0
T13	Facilidades especiales	1	0	0
TFactor = \sum (nivel * peso)				25.5

El peso de los factores técnicos será:

$$TCF = 0.6 + (0.01 * TFactor) = 0.6 + (0.01 * 25.5) = 0.855$$

Como siguiente paso, calculamos los factores ambientales, y nos basaremos en la tabla propuesta anteriormente:

Tabla 12. FACTORES AMBIENTALES EJEMPLO				
Factor	Descripción	Peso	Nivel	Pes*Niv
E1	Familiaridad con el modelo	1.5	2	3
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en orientación a objetos	1	5	5
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	5	5
E6	Estabilidad en los requerimientos	2	5	10
E7	Personal de medio tiempo	-1	5	-5
E8	Dificultad en el lenguaje de programación	-1	0	0
EFactor = \sum (Peso * Nivel)				22

El peso de los factores ambientales lo calculamos de acuerdo a la fórmula propuesta:

$$EF = 1.4 + (-0.03 * EFactor) = 1.4 + (-0.03 * 22) = 0.74$$

Los puntos de casos de uso (UCP) para el caso de uso “retirar dinero de cajero automático” son:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{EF} = 13 * 0.855 * 0.74 = \mathbf{8.225}$$

Tomando en consideración la propuesta del creador de la técnica descrita en este trabajo, asignamos **20 hrs-hombre por punto de casos de uso**, dando un total de **164.50 Hrs-Hombre**, es decir, 4.11 semanas para una sola persona trabajando tiempo completo. Los resultados finales los mostramos en la siguiente tabla:

Tabla 13. HORAS-HOMBRE TOTALES EJEMPLO		
Actividad	Porcentaje	Horas Hombre
Análisis	10%	41.12
Diseño	20%	82.25
Programación	40%	164.50
Pruebas	15%	61.68
Sobrecarga	15%	61.68
Total esfuerzo	100%	411.25

Analizando la tabla 12, podemos ver que el proceso total de desarrollo del caso de uso de este ejemplo son **411.25 hrs**, el equivalente a **10.28 semanas** para una sola persona (trabajando tiempo completo a 40 hrs por semana). Para un equipo de desarrollo de **5 personas** con trabajo de tiempo completo, demoraría **2.05 semanas** en crearse un sistema con dichas características.

Los costos de producción de sistema se calcular en función del número de horas trabajadas, multiplicando el costo deseado por hora y el total de ellas. Por ejemplo, si el costo por hora es de \$300, el costo total del sistema sería:

$$\text{Total} = 411.25 \text{ hrs} * 300 = \mathbf{\$123,376.5}$$

Y el plazo de entrega sería de 2 semanas, con equipo de trabajo de 5 personas de tiempo completo.

Los tiempos estimados por etapa (análisis, diseño, programación, etc.), así como los costos por hora, son criterio del equipo de desarrollo o de la empresa encargada de crear el sistema y dependen en gran medida de la experiencia de ellos.

CONCLUSIONES

Como pudimos observar, la técnica de puntos de casos de uso nos permite cuantificar eficientemente las estimaciones de los proyectos de software. Es una técnica basada en metodologías orientadas a objetos, las cuales son las principales herramientas de desarrollo.

La planeación es una fase importante de todo proyecto, y conocer este tipo de herramientas resulta imprescindible para todo líder de proyecto. Gestionar adecuadamente nuestros recursos, nos permitirán sentar unas bases sólidas y adecuadas en nuestros proyectos de software.

Considero importante que todo universitario o profesionalista debe conocer esta técnica de estimación, para realizar su gestión correctamente, y, como pudimos ver en este trabajo, la técnica de puntos de casos de uso nos ofrece una metodología bastante aceptable y adecuada para los proyectos de TI.

REFERENCIAS

- [1] Roger S. Pressman. (1998). *Ingeniería del software, un enfoque práctico*, España, Ed. McGraw-Hill.
- [2] Roy K. Clemmons. (2006). *Project estimation with Use Case Points*. EEUU, Crosstalk: The Journal of Defense Software Engineering.
- [3] Metodología de la Red Nacional de Integración y Desarrollo de Software Libre, *Gobierno Bolivariano de Venezuela*, [en línea]; 2009 [consulta 10 Septiembre 2009] Disponible: <http://merinde.rinde.gob.ve/>