

Análisis Comparativo de Estimación de Esfuerzo en el Desarrollo de Software

Cristian A. Remón¹, Pablo Thomas²

¹ Dpto. I+D Maker Electrónica, Mar del Plata, Argentina
cremon@makerelectronica.com.ar

² III-LIDI, Instituto de Investigación en Informática
Facultad de Informática, Universidad Nacional de La Plata, La Plata, Argentina
pthomas@lidi.info.unlp.edu.ar

Resumen. Las diversas metodologías de estimación de esfuerzo en el proceso de desarrollo de software han surgido como una necesidad temprana de poder evaluar la factibilidad económica de un proyecto. La evolución de los métodos de estimación no han sido como consecuencia de su invalidez o resultados erróneos, sino de la misma evolución de la tecnología, del acotamiento de los márgenes de error y de las exigencias del mercado en busca de productos de mayor calidad. Este trabajo propone modificar la metodología de Puntos de Casos de Uso de Gustav Karner para estimar el esfuerzo en el desarrollo de un producto de software, utilizando la Especificación de Requerimientos en vez de Caso de Uso. En este sentido, se presentan casos de estudio con resultados concretos.

Palabras Claves: Casos de Uso, Puntos de Caso de Uso, Estimación de esfuerzo, Especificación de Requerimientos.

1 Introducción

La calidad del software es medida, primordialmente, por el cumplimiento de los requerimientos definidos en etapas iniciales del ciclo de vida de desarrollo y por el proceso aplicado durante ese ciclo de vida. Para lograr este objetivo, todo el proceso de desarrollo de software debe ser administrado bajo un efectivo plan de proyecto [1], el cual debe garantizar que se concluya en costo y tiempo de acuerdo a los requerimientos definidos previamente.

El desarrollo de software es una actividad económica, y por lo tanto está sujeto a restricciones económicas además de las inherentemente técnicas, las cuales se encuentran explícitamente dentro del plan de proyecto [2].

Las actividades o tareas que se deben ejecutar para lograr el producto estipulado, requieren de la inversión de esfuerzo, el cual se estima en función de los requerimientos, y a que a partir de este valor como parte del costo del proyecto, se deriva el margen de ganancia que se obtiene por el producto terminado.

Uno de los métodos utilizados para estimar el esfuerzo de desarrollo de software

se basa en modelos de Casos de Uso [3]. Gustav Karner [4] tomó el modelo de Casos de Uso para mejorar la técnica de Puntos de Función en la estimación de esfuerzo en proyectos de software.

En este trabajo, continuación de la línea de investigación iniciada en [5], se propone determinar la factibilidad de utilizar la Especificación de Requerimientos para el cálculo de *Puntos No Ajustado de Caso de Uso (UUCP)* en vez del modelo de Casos de Uso, contabilizando entidades u objetos de análisis.

En el punto 2 se presenta el método de Gustav Karner para estimación de esfuerzo en el desarrollo de software. El punto 3 analiza el criterio de contabilización de casos de uso (transacciones, escenarios, entidades u objetos de análisis) para determinar el valor de UUCP; en el punto 4 se aplica el método de Karner en 10 casos de estudio contabilizando transacciones, escenarios y entidades u objetos de análisis para determinar el desvío de los resultados. En el punto 5 se propone un método de extracción y contabilización de entidades u objetos de análisis a partir de la Especificación de Requerimientos para obtener el valor de *Puntos de Requerimiento Funcional (FRP)* necesario para la estimación del esfuerzo. En el punto 6 se aplica el método propuesto a los mismos casos de estudio. Finalmente en el punto 7 se presentan las conclusiones y en el punto 8 trabajos futuros.

2 Método de Estimación de Esfuerzo de Gustav Karner [4]

El modelo de *Puntos de Caso Uso (UCP)* comienza “midiendo” la funcionalidad del sistema basada en el modelo de Casos de Uso en una cantidad denominada *Puntos No Ajustado de Caso de Uso (UUCP)*. El factor técnico involucrado es similar al método utilizado en Puntos de Función propuesto por Albrecht [4]. Se incluye un nuevo factor denominado *Factor de Entorno (EF)*. Los UCP representan una estimación del esfuerzo para desarrollar el sistema, el cual puede ser mapeado en horas/hombre para completar varias fases de Objectory [4].

2.1 UUCP - Puntos No Ajustados de Caso de Uso

Se identifican los actores involucrados en los casos de uso para calificarlos de acuerdo al siguiente criterio: *Simple* (peso=1) es aquel que representa una interfaz de programación o API (ej.: capa de abstracción); un actor *Medio* (peso=2) es aquel que interactúa mediante un protocolo (ej.: TCP/IP, HTTP, FTP) y un actor *Complejo* (peso=3) es aquel que interactúa por medio de una interfaz gráfica. Karner no especifica el origen de los pesos en ninguno de los casos.

Luego se clasifican los casos de uso de acuerdo al siguiente criterio: *Simple* si posee 3 o menos transacciones incluyendo cursos alternativos o casos de uso con menos de 5 objetos de análisis (peso=5), *Medio* si posee de 3 a 7 transacciones incluyendo cursos alternativos o casos de uso entre 5 y 10 objetos de análisis (peso=10), y *Complejo* si posee más de 7 transacciones incluyendo cursos alternativos o casos de uso con más de 10 objetos de análisis (peso=15).

Finalmente se suman los pesos de los actores y de los casos de uso para obtener el valor de Puntos No Ajustado de Caso de Uso (UUCP, por su sigla en inglés):

$$UUCP = \sum_{i=1}^{n} A_i * w_i + \sum_{i=1}^{n} CU_i * w_i$$

n =cantidad de actores o casos de uso

A = actor identificado

CU = caso de uso

w = peso del actor o caso de uso respectivamente

2.2 TCF – Factor de Complejidad Técnica

Determina la dificultad que conlleva la construcción del software. Cada factor técnico posee un grado de complejidad, que oscila entre 0 y 5; 0 significa un valor irrelevante o nulo y 5 determina un valor con alto grado de influencia. Cada factor técnico posee un valor de *peso*. El peso total de ese factor de influencia técnica se obtiene con el producto entre el valor de complejidad asignado y el peso que le corresponde al factor. La fórmula que permite calcular el TCF es:

$$TCF = 0,6 + 0,01 * \sum_{i=1}^{13} F_i * W_i$$

2.3 EF – Factor de Entorno

Indica la influencia del grupo humano involucrado en el proyecto sobre el sistema a desarrollar. De manera similar a los factores técnicos, los factores de entorno poseen un grado de influencia que oscila entre 0 y 5, donde 0 significa un valor irrelevante o nulo y 5 determina un valor con alto grado de influencia. Cada factor de entorno posee un valor de *peso*. El peso total de este factor se obtiene con el producto entre el valor de influencia asignado y el peso que le corresponde al factor de entorno. El cálculo final es de acuerdo a la fórmula:

$$EF = 1,4 + (-0,03 * (\sum_{i=1}^{8} E_i * W_i))$$

2.4 Cálculo de Puntos de Casos de Uso - UCP

El *Punto de Caso de Uso (UCP)* se calcula de la siguiente manera:

$$UCP = UUCP * TCF * EF$$

Karner [4] propone un factor de 20 horas / hombre por punto de caso de uso, resultado del valor medio obtenido en su investigación, para la estimación temprana del esfuerzo requerido para el desarrollo de un sistema de software.

$$Total\ horas/hombre = UCP * MR (recursos\ necesarios\ por\ UCP) = UCP * 20$$

3 Criterio de contabilización de casos de uso: transacciones, escenarios u objetos de análisis

El modelo de casos de uso propuesto por Ivar Jacobson [6] ha producido un alto impacto en el desarrollo de software. La aceptación de este modelo dio origen a variadas opciones para modelar un caso de uso. En algunos casos se modela un caso de uso por cada tipo de funcionalidad, otros modelan un caso de uso general con la idea de proyectar esta generalidad en la implementación de un control o librería que la soporte, y también existen los que agrupan estas funcionalidades en solo un caso de uso. Ninguna de esas formas son erróneas, sino que poseen ventajas y desventajas.

La clasificación de los casos de uso para determinar su valor de complejidad se basa en la cantidad de escenarios, transacciones o entidades (u objetos de análisis) que involucra. Si se toma el modelo de flujo principal y flujos alternativos, se deben incluir todas las transacciones que indiquen un estímulo/respuesta por parte del usuario al sistema, evitando el nivel de detalle en la descripción del flujo principal.

Si se toma el modelo de escenarios principales y alternativos, se contabiliza sólo la cantidad de escenarios del caso de uso. Un escenario se considera una transacción si se cumple con éxito toda la secuencia de pasos que la componen [7].

La elección del modelo a utilizar tendrá influencia directa en la estimación como consecuencia del conteo de transacciones u escenarios. Si se decide contar entidades u objetos de análisis, es importante considerar el modo de escritura de los casos de uso, respetando una línea gramatical:

sujeto + acción + objeto/sustantivo, ejemplo:
El usuario^[sujeto] ingresa^[acción] los datos de un Cliente^[objeto/sustantivo]

Si el diagrama de clases surge a partir del modelo de casos de uso, se puede identificar en esta etapa las entidades u objetos candidatos que formarán parte del modelo de objetos.

4 Aplicación del método de Karner en 10 casos de estudio

En esta sección se aplica el método de Karner en 10 casos de estudio: 8 tesinas de grado de proyectos de software de la Facultad de Ingeniería de la Universidad FASTA de la ciudad de Mar del Plata, Argentina; y en 2 proyectos de la industria de software de la empresa Maker, Argentina (www.makerelectronica.com.ar). El método aplicado sigue el lineamiento de contabilizar: transacciones que representen un estímulo del actor al sistema, contabilizar escenarios y entidades u objetos de análisis.

4.1 Descripción de Casos de Estudio

A continuación se enumeran los proyectos utilizados:

CE1- Gestión de cuentas corrientes en cooperativa de remises.

CE2- Gestión Técnica y Panel de Control de equipos de seguimiento satelital.

CE3- GAMA, Sistema informático que contribuye a la gestión de la atención médica ambulatoria.

CE4- SCRUM, Solución informática para la Unión de Rugby de Mar del Plata.

CE5- DO-RE-MI: Sistema de Gestión para un conservatorio de Música.

CE6- SIARER: Sistema Informático de Análisis de Recursos Energéticos.

CE7- MARATHON, Sistema de gestión deportiva y planificación de entrenamiento.

CE8- ZONDA, Sistema Informático de Análisis de Recursos Energéticos Renovables.

CE9-C.T.E.N.P, Sistema Informático para gestionar las tareas de la Escuela Nacional de Pesca de la ciudad de Mar del Plata.

CE10-GySP: Sistema Informático para gestionar y supervisar proyectos de software.

4.2 Resultados obtenidos

En la tabla 1 se presenta el cálculo de la estimación de esfuerzo utilizando los valores medios de contabilizar transacciones, escenarios y entidades, y luego se compara dicha estimación con el valor medio del esfuerzo real.

Tabla 1. Cálculo de estimación de esfuerzo utilizando los valores medios

	UUCP	TCF	EF	UCP	Media del Esf. Real	Esf. Estimado UCP * 20	%Desvío
Transacciones	475	0,979	1,153	536,17	2182,30 hh	10723,47 hh	391,38
Escenarios	424,30	0,979	1,153	478,94	2182,30 hh	9578,88 hh	338,93
Entidades	116,50	0,979	1,153	131,50	2182,30 hh	2630 hh	20,51

Se observa que el menor porcentaje de desvío se manifiesta en la contabilización de entidades; es decir, la utilización de entidades muestra una menor diferencia entre la cantidad de horas hombre estimadas y las realmente utilizadas.

4.3 Origen de las desviaciones

En función del análisis de los resultados obtenidos en la sección anterior se puede afirmar que las desviaciones tienen origen en el valor de UCP, debido a la ausencia de una norma o estándar que indique la manera de contabilizar la clasificación de los casos de uso, por transacciones, por escenarios o por entidades. Como causa de las múltiples formas existentes para describir un caso de uso, el resultado del valor de UCP puede variar considerablemente aplicando contabilización por transacciones u escenarios; en cambio, si se cuentan entidades u objetos de análisis, el porcentaje de desvío disminuye considerablemente.

Se opta por contar entidades u objetos de análisis para clasificar la complejidad del caso de uso, y se plantea aplicar el mismo método pero a partir de la Especificación de Requerimientos, clasificando cada requerimiento funcional de acuerdo a la cantidad de entidades u objetos de análisis involucrados, y así extraer el valor *UUCP*. De esta manera, es posible realizar una estimación temprana del esfuerzo necesario para construir un producto de software, ahorrando además el tiempo de diseño del modelo de casos de uso para aplicar el método de UCP de Karner.

5 Extracción de entidades u objetos de análisis a partir de la Especificación de Requerimientos

5.1 Método de Booch para identificar objetos

El método que se propone utilizar es el propuesto por Booch [8] que originó el método gramatical, el cual parte de una descripción textual del dominio del problema para identificar los objetos, sus atributos y comportamiento, con el objetivo de diseñar las clases que estarán involucradas en el producto de software a construir.

El método consta de los siguientes pasos:

- a. identificar los objetos y sus atributos
- b. identificar las acciones o comportamiento que involucran a cada objeto
- c. establecer la visibilidad de cada objeto y la relación entre otros objetos
- d. establecer la interfaz de cada objeto
- e. implementar cada objeto

El primer paso es *identificar los objetos y sus atributos*, los cuales se derivan de los sustantivos que se utilizan en la descripción del dominio. En algunas situaciones puede suceder que existan referencias a objetos similares; en estos casos lo importante es identificar aquellos objetos que son mencionados en reiteradas oportunidades. El primer paso de este método es el que se utiliza en este trabajo para identificar los objetos o entidades candidatas en la Especificación de Requerimientos.

Se pone especial énfasis sobre la sección de Requerimientos Funcionales, a los cuales se le aplica el método.

A continuación se aplica el método de Booch a una Especificación de Requerimientos del CE1 descrito en la sección 4.1, para identificar y extraer los objetos o clases candidatas que determinarán la complejidad de un requerimiento funcional. Se subrayan los objetos o entidades candidatas identificadas según la metodología mencionada:

RF 5. El sistema deberá contemplar la generación de una liquidación al momento del ingreso en servicio de una nueva licencia en la cooperativa. Dicha liquidación es proporcional a los días que la licencia prestará servicios hasta el final del mes en que ingresa. Es decir que si la licencia ingresara en la cooperativa el día 10 / 08, se

le generará un débito por el importe de la liquidación mensual sobre los días que presta servicios en el mes, por ejemplo, suponiendo que el importe mensual es de 100\$, al momento del ingreso se le debita en su cuenta corriente un importe de $(\$100 / 30 \text{ días}) \times 20 \text{ días restantes que restan para finalizar el mes}$.

Una vez identificadas las entidades candidatas, se determina la complejidad del requerimiento de acuerdo a la clasificación de la tabla 2. La clasificación utilizada es la propuesta por Karner en la sección 2.1, a la cual se le disminuye el tope de entidades en la clasificación *Complejo*, con el objetivo de subdividir el requerimiento cuando éste supera la cantidad de 10 entidades identificadas.

Tabla 2. Clasificación de complejidad

Complejidad	Definición	Peso
Simple	Cantidad de entidades ≤ 3	5
Medio	Cantidad de entidades ≤ 7	10
Complejo	Cantidad de entidades ≤ 10	15

En la tabla 2 se observa que en la cantidad de entidades que determina el peso de complejidad, se introdujo un valor tope de 10 con respecto a la clasificación original propuesta por Karner. Esta modificación tiene origen en que le correspondería un peso de 15 tanto a un requerimiento que posea 9 entidades identificadas como a uno que posea más entidades. En estos casos se recomienda analizar el requerimiento y aplicar una subdivisión.

La sumatoria de los pesos de las clasificaciones de los requerimientos funcionales se transforman en un valor denominado *Pesos de Requerimiento Funcional (FRP)*.

El valor de *FRP* se suma al valor de la sumatoria del peso de los actores identificados para formar la variable *Puntos No Ajustado de Requerimiento Funcional (UFRP)*, una analogía de la variable Puntos No Ajustado de Caso de Uso (UUCP) [4] definida por Karner. Los valores y la forma del cálculo de la complejidad técnica (TFC) y de entorno (EF) no son modificados. La denominación de la variable *Punto de Caso de Uso (UCP)* es reemplazada por *Puntos de Requerimiento Funcional (FRP)*, donde su valor se calcula de la siguiente manera:

$$\text{Puntos de Requerimiento Funcional (FRP)} = \text{Puntos No Ajustado de Requerimiento Funcional (UFRP)} * \text{Factor de Técnico de Complejidad (TFC)} * \text{Factor de Entorno (EF)}$$

Se puede observar que la metodología de Karner ha experimentado una modificación sólo en la parte del cálculo de complejidad en la definición de funcionalidades, reemplazando los Casos de Uso por la Especificación Requerimientos, con el objetivo de obtener un valor de estimación en una etapa temprana en el desarrollo del producto de software.

El siguiente paso es obtener las horas/hombre estimadas que se consumirán:

$Esfuerzo\ de\ Proyecto\ (EP) = FRP * 20$, donde 20 es la unidad asignada por Karner a cada punto de caso de uso.

6 Aplicación del método de Booch en la Especificación de Requerimientos de los Casos de Estudio

En esta sección se analizan los resultados obtenidos de aplicar el método de Booch para identificar entidades en cada uno de los casos de estudio.

Tabla 3. Valores medios de entidades identificadas en los casos de estudio

	Total de entidades según Karner	Total de entidades según Booch	% Desvío
Media	26,50	32,40	-18,21

En la tabla 3 se presenta la identificación de entidades a partir de los casos de estudio utilizando el método de Karner, y la identificación de entidades a partir de la Especificación de Requerimientos utilizando el método de Booch.

La diferencia entre ambos métodos de identificación produjo un desvío del 18%. Lo cual significa que contar entidades a partir de una Especificación de Requerimientos no posee una diferencia significativa en relación a contar entidades a partir de Casos de Uso.

Tabla 4. Comparación de valores de UCP y FRP a partir de contabilizar entidades

UCP	FRP	% Desvío
131,50	115,02	14,32

En la tabla 4 se comparan los valores de Puntos de Casos de Uso (UCP) y de Puntos de Requerimiento Funcional (FRP). El resultado de contabilizar entidades aplicando el método Booch, a partir de los Requerimientos Funcionales, arrojó sólo un 14% de desvío con respecto al método de UCP, lo cual indica la validez de contabilizar entidades a partir de requerimientos por sobre los Casos de Uso.

Tabla 5. Estimación de esfuerzo utilizando valores medios con método de UCP aplicados en los casos de estudio

UUCP	TCF	EF	UCP	Esfuerzo Estimado HH	Esfuerzo Real HH	% Desvío
116,5	0,979	1,153	131,50	2630,07	2182,3	20,52

Tabla 6. Estimación de esfuerzo utilizando valores medios con método de FRP aplicados en

los casos de estudio

UFRP	TCF	EF	UFR	Esfuerzo Estimado HH	Esfuerzo Real HH	% Desvío
101,9	0,979	1,153	115,02	2300,47	2182,3	5,41

En la tabla 5 y 6 se comparan los valores medios tomados de contar entidades y clasificar casos de uso, y los valores medios de contar entidades y clasificar requerimientos funcionales. En los porcentajes de desvío con respecto a la media del esfuerzo real invertido, se puede observar que si se utilizan los requerimientos funcionales para contabilizar entidades y clasificar la complejidad funcional del proyecto, es posible disminuir el margen de desvío con respecto al esfuerzo real.

Finalmente en la figura 1, se presenta una comparación entre la estimación de horas/hombre utilizando el método de Booch para calcular el valor de FRP a partir de la Especificación de Requerimientos, la estimación de horas/hombre utilizando el método de UCP y las horas/hombre reales invertidas.

7 Conclusiones

Lo expuesto en este trabajo indica que la aplicación del método de Puntos de Caso de Uso propuesto por Gustav Karner, sobre diez casos de estudio reales, genera desviaciones excesivas en la estimación de esfuerzo de un proyecto de software al contabilizar transacciones o escenarios, debido a múltiples formas existentes para describir un caso de uso. El resultado del valor de UCP puede variar aplicando contabilización por transacciones u escenarios; en cambio, si se cuentan entidades u objetos de análisis, el porcentaje de desvío disminuye considerablemente, lo cual no sucede cuando se contabilizan transacciones u escenarios, como fue mostrado en el punto 4. Así como es posible contabilizar entidades en los casos de uso, también es factible contabilizar las entidades a partir de la Especificación de Requerimientos, a fin de lograr una estimación temprana del esfuerzo para construir un producto de software, con el ahorro del tiempo de diseño del modelo de casos de uso.

La clasificación de los requerimientos sumado a la clasificación de los actores genera el valor de *Puntos No Ajustado de Requerimiento Funcional (UFRP)*, que multiplicado por los factores técnicos y de entorno genera el valor de Puntos de Requerimiento Funcional (*FRP*), lo cual indica los puntos de requerimientos funcionales calculados. Los valores medios calculados en la contabilización de entidades u objetos de análisis a partir de Casos de Uso y luego a partir de la Especificación de Requerimientos, arrojaron resultados que indican la viabilidad de utilizar este método. En resumen, es factible estimar esfuerzo a partir de una Especificación de Requerimientos con resultados aceptables en relación a estimar a partir de Casos de Uso. Esto permite realizar una estimación temprana, sin la necesidad de construir Casos de Uso.

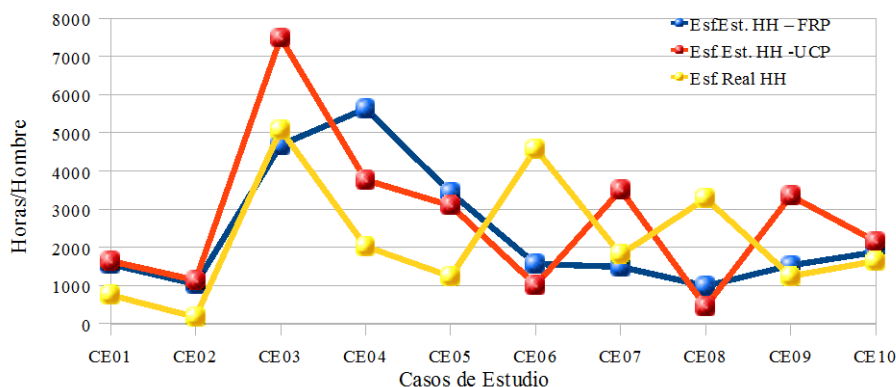


Figura 1. Comparación entre las horas-hombre reales invertidas y las estimaciones realizadas

8 Trabajos Futuros

Se considera necesario aplicar una serie de ajustes en la clasificación cuantitativa de Requerimientos Funcionales, así como también en la inclusión de un factor de complejidad que contemple la reutilización de código de otros proyectos.

También se prevé normalizar el formato de escritura de los requerimientos funcionales y realizar un análisis de los valores en la clasificación de los requerimientos.

Bibliografía

1. Kusumoto, S., Matsukawa, F., Inoue, K., Hanabusa, S., Maegawa, Y.: Estimating Effort by Use Case Points: Method, Tool and Case Study. In: Proceedings of the 10th International Symposium on Software Metrics, pp. 292--299 (2004)
2. Sommerville, I.: Ingeniería de Software, 6ta. Edición. Addison, Wesley (2002)
3. Anda, B., Dreiem, H., Sjøberg, D.I.K., Jørgensen, M.: "Estimating Software Development Effort Based on Use Case – Experience from Industry. In: M. Gogolla, C., Kobryn, C. (Eds.) UML 2001. LNCS, vol. 2185, pp. 487-502. Springer-Verlag (2001)
4. Karner, G.: *Resource Estimation for Objectory Projects*, Objectory Systems, Suecia (1993)
5. Remon C., Thomas P., Análisis de Estimación de Esfuerzo aplicando Puntos de Caso de Uso, CACIC 2010, Octubre 2010, Universidad de Morón, Argentina.
6. Jacobson I,Booch G., "The Unified Modeling Language User Guide" - Editorial Addison-Wesley. ISBN:0201571684
7. Caroll, E. 'Estimating Software based on use case points', OOPSLA 2005, San Diego , California, USA.
8. Booch G., *Object Oriented Development*, 1986, IEEE Transaction On Software Engineering, Vol. SE-12, No. 2, February 1986