

El peligro de un Caso de Uso muy largo. Mitos y realidades

The risk of writing very long Use Cases. Myths and realities

Susel Ruiz Durán¹, Yenisleidy Piloto Lastra¹, Reynaldo Roselló Núñez¹

¹Universidad de las Ciencias Informáticas (UCI)

{[sruiz](mailto:sruiz@uci.cu), [ypiloto](mailto:ypiloto@uci.cu), [rosello](mailto:rosello@uci.cu)}@uci.cu

Resumen

Hasta la fecha, la producción en la Universidad de las Ciencias Informáticas (UCI) ha estado mayormente orientada hacia las aplicaciones informáticas “a la medida”. La experiencia acumulada en este sentido es relativamente amplia y es recomendable utilizarla en función del mejoramiento de los procesos de producción, especialmente el tema de la productividad y la eficiencia. Gran cantidad de proyectos fracasan en el mundo por una mala estimación de esfuerzo, se incrementan los tiempos y los presupuestos. De ahí la importancia de la estimación del tamaño y la complejidad del software y del esfuerzo de desarrollo, que nos permita estimar recursos, administrar y evaluar la productividad y la calidad y administrar eficientemente los contratos, entre otras ventajas. En función de la realidad de la UCI, y tomando como ejemplo el uso de diferentes patrones de Casos de Uso entre los proyectos, se hace un análisis de las debilidades existentes, enfatizando en las necesidades crecientes de contar con métricas que soporten la producción en el centro, desde todos los puntos de vista, que van desde la negociación hasta el desarrollo de aplicaciones informáticas y la evaluación de los proyectos.

Palabras clave: estimación de tamaño y complejidad, patrón CRUD de casos de uso, tamaño del software.

Abstract

To the present, the production in the UCI (in Spanish: Universidad de las Ciencias Informáticas) has been mainly oriented towards “custom-made” software applications. The accumulated experience in this sense is relatively ample and is recommendable to use it in order to improve the production processes, specially the subject of the productivity and the efficiency. Great amount of projects fails in the world due to a bad estimation of effort, the times and the budgets are increased. That is why it is important to estimate size and complexity of the software and the effort of development, and allows estimating resources, managing and evaluating the productivity and the quality and managing contracts efficiently, among other advantages. Based on the reality of the UCI, and taking as example the use of different Use Cases patterns in different projects, an analysis of the existing weaknesses is made, emphasizing in the increasing necessities to count on a metric that supports the production in the institution, from all the points of view, that go from the negotiation to the development of software applications and the evaluation of projects.

Keywords: CRUD use case pattern, size and complexity estimation, software size.

Introducción

La Universidad de las Ciencias Informáticas centra su razón de ser en la formación de profesionales de la rama de la informática en un proceso que integra de manera cohesiva la docencia y la producción de productos y servicios informáticos, que aporten ingresos considerables al país.

Hasta la fecha, la producción en el centro ha estado mayormente orientada hacia las aplicaciones informáticas “a la medida”, es decir, las aplicaciones desarrolladas para cubrir las necesidades específicas de un cliente determinado. La experiencia acumulada en este sentido es relativamente amplia y ha llegado el momento de utilizarla en función del mejoramiento de los procesos de producción, especialmente el tema de la productividad y la eficiencia.

Estos aspectos son, en términos de producción de software, temas de una complejidad considerable, por lo que ameritan un estudio profundo y sistemático, que se nutra de la propia experiencia de los proyectos desarrollados hasta el momento.

Este trabajo aborda algunos elementos relativos al tema, referidos al tamaño del software, la necesidad de una herramienta adecuada de estimación y los riesgos que trae consigo el hecho de desconocer determinados factores. Específicamente se centra en la conveniencia de utilizar un patrón de Casos de Uso en específico (Patrón CRUD) siempre y cuando estén dadas una serie de condiciones objetivas.

Materiales y Métodos

Los métodos utilizados incluyen métodos empíricos de observación, así como métodos teóricos como el Analítico Sintético.

Resultados y Discusión

El tamaño del Software

En la actualidad, una gran cantidad de proyectos fracasan debido a una mala estimación del esfuerzo de desarrollo, que conlleva a que se agote el tiempo y/o el presupuesto fijado inicialmente para el mismo. Es por ello que la estimación del tamaño del software y del esfuerzo de desarrollo se ha convertido en uno de los aspectos más relevantes en el mundo de la ingeniería de software y la gestión de proyectos informáticos.

Estimar el tamaño de una aplicación informática, sobre todo en la fase de inicio del proyecto, es una tarea bien difícil debido al gran número de factores desconocidos que existen en esta etapa, así como los factores conocidos complejos.

Sin embargo, el hecho de tener una estimación lo más real posible del tamaño del sistema que se desea construir permite, entre muchas otras cosas:

- Estimar recursos (personal, tiempo, medios de trabajo, presupuesto).
- Administrar y evaluar la productividad de los recursos (horas/funcionalidad).

- Administrar y evaluar la calidad de los productos (defectos/funcionalidad).
- Comparar proyectos que hayan sido estimados bajo las mismas reglas.
- Administrar los contratos, basándose en los requerimientos funcionales, tamaño de la funcionalidad, productividad esperada y costo por unidad de tamaño de la funcionalidad.

Uso de métricas para estimación de tamaño de las aplicaciones informáticas

Existen diversas métricas usadas internacionalmente para la estimación de tamaño, complejidad y esfuerzo necesario para la realización de un proyecto informático.

Los resultados de aplicar cada una de ellas difieren entre sí, así como los resultados de aplicar una misma métrica con valores de calibración diferentes.

De ahí la importancia de seleccionar y calibrar adecuadamente la métrica a utilizar en el marco de una empresa de producción de software, generalizar su uso y aplicarla estrictamente, como forma de planificar, administrar, evaluar y comparar los diferentes proyectos que se emprendan. Así mismo, el propio proceso de producción y sus resultados medibles proporcionan una retroalimentación muy valiosa para la concepción de la métrica en sí. De esta forma, la métrica seleccionada no puede verse como una ley estática e inmutable, sino como una herramienta perfeccionable a lo largo de la vida de la empresa.

¿Cuándo puede decirse que una métrica es adecuada?

Esta es una pregunta compleja, cuya respuesta completa va más allá del alcance de este artículo.

Sin embargo, pueden mencionarse algunos factores que no deben escapar de la atención a la hora de concebir, seleccionar o calibrar la métrica a utilizarse.

Por ejemplo, si se usan los Casos de Uso como unidad de medida de las aplicaciones y como guía del proceso de desarrollo, es importante tener en cuenta para estimar el tamaño la cantidad de escenarios, la cantidad de entidades de dominio y sus atributos, la comunicación con dispositivos de hardware y con otros sistemas informáticos.

Así mismo, para estimar el esfuerzo y el tiempo de desarrollo es importante contemplar la experiencia del equipo de desarrollo en el uso de las tecnologías a utilizar, la madurez de los procesos de la empresa, el nivel de cohesión del equipo, entre otros factores.

En esencia una métrica es mejor cuanto más se acerquen los resultados de las estimaciones que la utilicen a la realidad comprobada de los proyectos. Pero es casi imposible asegurar su efectividad sin haberla aplicado con anterioridad.

Por tal motivo se hace necesario para la UCI, como empresa productora de software, la selección y aplicación de una métrica lo más completa posible, así como su posterior refinación a partir de los resultados de los proyectos productivos.

Criterios para la identificación de los Casos de Uso. Patrones

Uno de los problemas a los que se enfrenta un equipo de desarrollo en las etapas tempranas de un proyecto, es el referido al criterio de selección de los Casos de Uso que conforman la aplicación a desarrollar. Existen en el mundo de la ingeniería de requerimientos una amplia gama de patrones de Casos de Uso que pueden ser utilizados, pero la pregunta que salta a la vista es: ¿Cuál es más conveniente? ¿Cuál se adecua mejor a las características del producto que se desea construir y a las peculiaridades de todos los involucrados, cliente incluido?

Un ejemplo bastante común, en el ámbito del desarrollo de aplicaciones de gestión, es la disyuntiva entre utilizar el patrón CRUD (Create, Read, Update and Delete, por sus siglas en inglés) o no. Es decir, la elección entre concebir Casos de Uso que reúnan todas las acciones básicas sobre una entidad de dominio en uno solo, o concebir un Caso de Uso para cada una de estas acciones básicas para cada entidad de dominio.

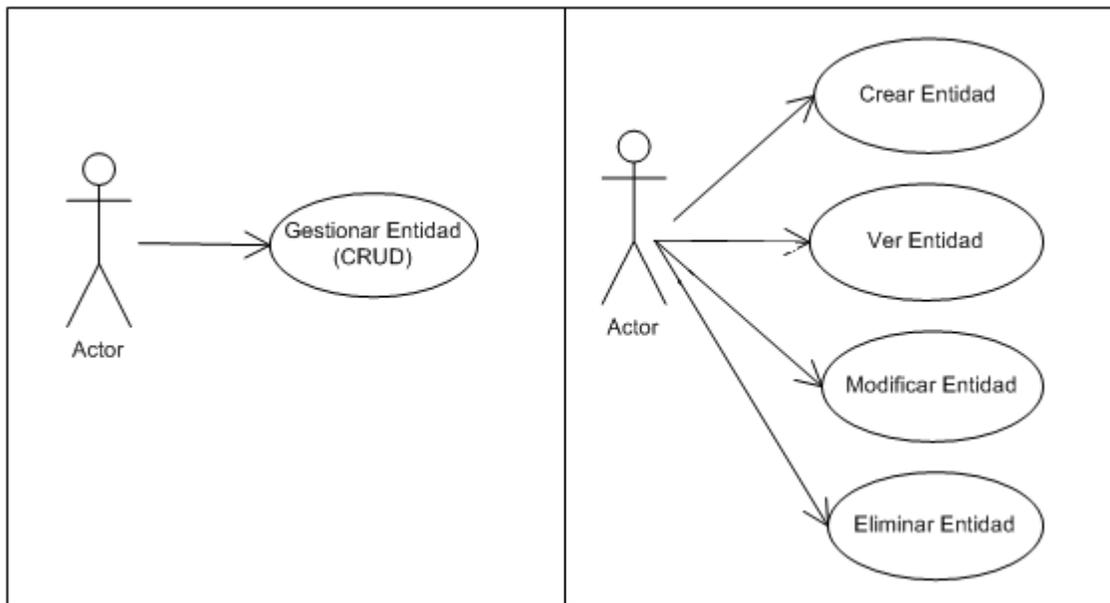


Fig. 1. CRUD VS no CRUD.

Cualquiera de las variantes es válida, siempre y cuando exista una métrica que vaya más allá del Caso de Uso como unidad de tamaño, y nos permita estimar la complejidad del mismo.

De otra forma, la planificación, evaluación y comparación de proyectos tratada con anterioridad, carece de sentido práctico: un Caso de Uso concebido bajo el patrón CRUD es equivalente a 4 Casos de Uso concebidos sin usar el patrón, y no pueden ser considerados como equivalentes de ninguna forma.

Ventajas y desventajas de la utilización del patrón de Casos de Uso CRUD

Entre las ventajas de la utilización del patrón CRUD pueden mencionarse:

- Se reúnen en un solo elemento de configuración del software todas las acciones básicas que se realizan sobre una entidad de dominio.
- Se facilita la comprensión por parte del cliente de la funcionalidad del sistema.
- Se facilita la especificación de los casos de uso, logrando un alto nivel de detalle sin tener que invertir esfuerzo en describir aspectos generales de funcionalidad más de una vez.
- Se facilita la reusabilidad del código, a partir de identificar relaciones entre los Casos de Uso, con un mínimo de esfuerzo.

La desventaja principal radica en que, si no existe una métrica completa que permita estimar la complejidad de los Casos de Uso, y la estimación y planificación de los proyectos permanece al nivel del Caso de Uso como unidad, casi seguramente el proyecto incurrirá en atrasos y sobrepasará su presupuesto, en comparación con otros que no usen dicho patrón.

Desde el punto de vista de las actividades de análisis, diseño, implementación y pruebas, la decisión de usar este patrón carece de implicaciones negativas: es equivalente desarrollar 4 funcionalidades si están especificadas en 4 Casos de Uso diferentes o en uno solo.

Una mirada desde el ángulo comercial

Como se planteó con anterioridad, una de las ventajas de tener una estimación adecuada del proyecto es el impacto positivo que se tiene en la administración de contratos.

Es imposible garantizar productividad en la esfera de la producción de software sin una herramienta que asegure la misma desde la etapa más temprana de un proyecto: la negociación, especialmente en el caso de las soluciones “llave en mano” que incluyen generalmente el desarrollo de aplicaciones informáticas “a la medida”.

El estudio inicial de procesos de negocio y la determinación, a través de métricas, del tamaño, complejidad y esfuerzo de desarrollo, pasan a convertirse en esta herramienta.

Volviendo al tema de los Casos de Uso que aplican el patrón CRUD, es evidente que no puede ser igual el costo de un Caso de Uso CRUD al de un Caso de Uso que solo contenga la funcionalidad referida a una sola acción básica sobre una entidad de dominio. Pero esto es parte de la negociación y de los aspectos que se deben pactar en el marco de un contrato, para evitar que posteriormente el proyecto se vuelva inmanejable dentro del tiempo y presupuesto estimado en sus inicios.

Conclusiones

Se hace de suma necesidad la selección y calibración o el desarrollo propio de una métrica o conjunto de métricas que soporten la negociación de contratos, planificación de tiempos y presupuestos, selección de personal, evaluación de la calidad y productividad de los proyectos y comparación de estos en el marco de la producción en la Universidad de las Ciencias Informáticas.

Esta métrica debe contemplar una serie de factores que, dada la experiencia alcanzada en los proyectos desarrollados hasta la fecha en el centro, se ha comprobado que tienen un peso importante y gran impacto en los tiempos de desarrollo, ajuste a presupuestos preestablecidos y calidad de los productos que se desarrollan.

De esta forma se puede estandarizar la producción y a la vez diversificar los modos de hacer las cosas, sin riesgo de fracaso del proyecto, porque se contará con las herramientas necesarias para que, independientemente de decisiones técnicas específicas de cada proyecto, se pueda estimar con la mayor exactitud posible en etapas tempranas la magnitud de lo que se pretende construir y actuar en consecuencia.

Agradecimientos

A Félix Rodríguez Figueroa, que nos motivó e instó a escribir este trabajo.

Referencias Bibliográficas

Colectivo de Autores. Estimación de esfuerzo en proyectos de software, 2007. [Disponible en: <http://www.slideshare.net/guestd0e1ff/s212-pf-pcu/>]

La Torre, Ludisley; Cepero, Mariela. Propuesta de métrica de perfeccionamiento de gestión de la calidad en el proceso de desarrollo, 2007. [Disponible en: <http://www.monografias.com/trabajos55/proceso-de-desarrollo-software/proceso-de-desarrollo-software.shtml>].