

Estimación de Software con COSMIC

—
Por Carlos Eduardo Vázquez

● Hacer una estimación “bottom-up” es inviable cuando no está disponible la estructura de proyecto, y hacer una estimación solamente basada en una analogía es muy subjetivo. Además, no se puede aprender de los errores cometidos. El objetivo de este artículo es introducir el método de medición de COSMIC y presentar una propuesta para derivar unidades de producto a partir de los requerimientos funcionales del usuario en diferentes representaciones.

La problemática de la estimación

Cuando se solicita una medición a un desarrollador para entregar un programa y él provee una estimación de 12 horas, lo más probable es que esté correcto porque se trata de una pieza cuya dificultad o probabilidad de error es pequeña.

Es así que para estimar proyectos de software típicamente se aplica una estrategia bottom-up que consiste en descomponer el proyecto en sus partes constituyentes, estimar cada una de las partes y sumar los resultados para una estimación total.

La falla en esta lógica es que al inicio no se conocen todos los programas y algunas actividades no están en función de esta cantidad, como por ejemplo la ingeniería de requerimientos y el diseño de arquitectura, cuyos productos son los insumos para la programación usada en el ejemplo de apertura. En otras palabras, el nivel de información disponible no permite usar la lógica de la estimación bottom-up como solución para los desafíos de la estimación.

¿Tiene caso estimar?

Ante esta problemática, hay quienes argumentan que este tipo de estimación bottom-up cuando no se conocen los programas requeridos es un esfuerzo inútil, y que es preferible mejor aplicar ese esfuerzo a ejecutar iteraciones de entre 15 y 30 días que nos permitan resolver los aspectos desconocidos, y entonces ya poder “saber” en lugar de simplemente creer.

El problema es que a nivel directivo se requiere tener estimaciones para poder tomar decisiones. La estimación permite tomar decisiones respecto al desarrollo o mejora de sistemas. ¿Tiene caso hacer este proyecto?, ¿su beneficio será mayor que su costo?, ¿cuándo es el momento para hacerlo?, ¿cuál es el 20% de los proyectos que ejecutaremos para atender el 80% de las demandas?

Contar con estimaciones previas también facilita que los equipos sean auto-administrados y que su desempeño sea planeado y monitoreado de acuerdo con las exigencias de transparencia y eficiencia del gobierno corporativo.

Unidad de producto de software

Algunos argumentan que el proceso de software es único y está más allá de cualquier medición. Sin embargo, existen similitudes de éste con la construcción civil en escala industrial. Cada edificio es único, a pesar de que pueda compartir los elementos arquitectónicos comunes en mayor o menor medida. El proceso de entrega de un inmueble involucra desde la concepción de un proyecto arquitectónico, pasando por varios proyectos de ingeniería, construcción y pruebas específicas, hasta la aceptación final por parte del propietario y la garantía por un periodo de transición.

Cuando se construye un edificio, es necesario tener un presupuesto disponible para su construcción. Una información clave en ese momento es el valor del costo por metro cuadrado de construcción. En base a eso podemos tomar varias decisiones: qué tan grande va a ser, cómo vamos a fondear su desarrollo, etc. En esta etapa no lidiamos con el costo detallado de cada componente del edificio. Después de todo, es mucho más costoso (proporcionalmente) construir un baño que una habitación, ya que el baño tiene un costo mayor de mano de obra y materiales. Posteriormente cuando sea necesario hacer la estimación de costo específica del baño, ya contaremos con suficiente información para hacer una estimación bottom-up.

En este punto surge un importante concepto: la productividad. Esta puede ser definida como la razón entre la cantidad de bienes o servicios producidos y las unidades de tiempo o costo para su entrega.

En el caso citado del edificio, la planeación de productividad se hace en dólares por metro cuadrado (USD/m²). Es decir que medimos nuestro producto en términos de metros cuadrados, y observamos el costo generado por cada unidad de producto.

En el caso del desarrollo de software, necesitamos identificar una unidad que nos permita medir el producto durante la planeación. Dicha unidad debe permitir aproximar el tamaño del software a

Carlos Eduardo Vázquez es fundador y responsable de investigación y desarrollo en Fatto Consultoría, empresa de origen brasileño especializada en consultoría para la estimación de proyectos de software. Tiene más de 20 años de experiencia en investigación, consultoría y entrenamiento en estimación con puntos de función. Fue uno de los tres primeros brasileños certificados expertos en puntos de función por el IFPUG en 1996 y uno de los primeros brasileños certificados por COSMIC en 2012. carlos.vazquez@fattocs.com

partir de sus requerimientos, y debe ser independiente del desarrollo técnico y decisiones de implementación.

Este tipo de unidad no elimina la necesidad de otras métricas que permitan cuantificar el desempeño técnico de productos y servicios a partir de su implementación, como por ejemplo: análisis de eficiencia del diseño, apoyo a la ingeniería de requerimientos, apoyo a la verificación y validación. Un ejemplo de métricas de este tipo son las que define ISO/IEC 25.000 (SQuaRE).

Considerando las características deseadas para esta unidad de producto, se deben tener como objetivo de medición los requerimientos funcionales del usuario, ya que están específicamente asociados a una tarea o servicio del usuario y describen lo que el software debe hacer independientemente de cómo sea desarrollado.

Contar los requerimientos funcionales parece ser una buena alternativa para representar las unidades de producto del software. Sin embargo, no todos los requerimientos son iguales y se corre el riesgo de no diferenciarlos. Para eso, la ISO/IEC definió un estándar para ese tipo de medición, denominado Medición del Tamaño Funcional, y cuyo método más moderno es supervisado por el Consorcio Internacional de Medición de Software en General o COSMIC.

COSMIC es la segunda generación de métodos de medición de tamaño funcional. Es aplicable a todo tipo de software, compatible con la ingeniería de software moderna, es de dominio público, su documentación no tiene costo, y es reconocido por ISO/IEC.

Proceso COSMIC

El método de medición consiste en un conjunto de modelos, principios, reglas y procedimientos que se aplican para determinar el valor de una magnitud para la funcionalidad entregada por el software. Dicha magnitud de la funcionalidad del software es expresada en puntos de función COSMIC.

A muy grandes rasgos, el proceso consiste en:

1. Establecer la frontera entre el sistema y los actores con los que interactúa (que pueden ser personas u otros sistemas).
2. Identificar los procesos funcionales que los actores reciben del sistema.
3. Para cada proceso funcional, identificar los movimientos de datos que genera. Cada que el usuario provee al sistema un conjunto de datos para realizar un proceso, hay un movimiento de datos. También es un movimiento de datos cuando el sistema provee grupos de datos al usuario, y cuando se acceden o almacenan grupos de datos en almacenamiento. Cada movimiento de datos contribuye con el equivalente a un punto de función COSMIC.

Medición vs aproximación del tamaño

Debemos recordar que en las primeras etapas del proyecto no es factible hacer estimaciones detalladas. El alcance estará definido en términos de macroprocesos de negocio y áreas funcionales. Estos elementos pueden ser contados y comparados con su correlación con el software entregado y medido en puntos de función COSMIC.

En los momentos posteriores en el ciclo de vida, cuando ya existe un alcance definido en términos de cuáles tareas el usuario deben ser parcialmente o totalmente transferidos para el software, es posible identificar procesos y aplicar la misma lógica en la extrapolación de la cantidad de puntos de función COSMIC a partir de la cantidad de procesos.

Al final del proyecto, una vez que se tiene el sistema implementado, se hace una medición (ya no es una estimación) del producto terminado. Esto se hace para:

- Evaluar el desempeño mediante la relación entre la cantidad de horas invertidas y la cantidad de puntos función COSMIC medidos.
- Revaluar los indicadores de productividad para que pasen a incluir el desempeño del proyecto que acaba de terminar.
- Revaluar la cantidad de puntos función COSMIC que corresponden en promedio a los procesos y a los conceptos de negocio conforme al nivel de información disponible en los diferentes puntos que se desea estimar.

Benchmarking

Los resultados reales del proyecto (tamaño, tiempo, esfuerzo y costo) nos sirven para compararlos con nuestra estimación inicial y en base a eso ajustar nuestro proceso de estimación para próximas estimaciones. De manera similar, debemos calcular cuál fue la productividad real (ej. horas hombre por punto de función) y tomarla en cuenta para futuras decisiones.

Con datos históricos de proyectos en la empresa podemos hacer benchmarking para comparar nuestra estimación. Por ejemplo, si estimamos un proyecto para el desarrollo en Java como si tuviera 150 puntos de función COSMIC y un esfuerzo de 1,050 horas hombre esto equivale a una tasa de entrega de 7 horas hombre por punto función (hh/cfp). Si tenemos una base de datos histórica de proyectos podríamos determinar qué tan factible es lograr una productividad de 7 hh/cfp.

Si no contamos con datos internos, podemos apoyarnos en bases de datos de la industria. Por ejemplo, el ISBSG (International Software Benchmarking Standards Group) provee herramientas y bases de datos para la estimación de proyectos de software. 🌐