

# Influencia de los riesgos sobre los métodos de estimación para los proyectos de software

## *Influence of risks on the assessment methods for software projects*

Dayami Rodríguez Brito, Miguel Ángel Monagas Reyes

Universidad de las Ciencias Informáticas

[drodriguez@uci.cu](mailto:drodriguez@uci.cu)

### **Resumen**

Es el proceso de gestión para la creación de un sistema o software, el que encierra un conjunto de actividades; una de las cuales es la estimación.

Estimar es proyectar, aceptando cierto grado de incertidumbre el futuro que depara un proyecto en cuestión, siendo además la base esencial de las restantes actividades de planificación. [1]

Para gestionar la incertidumbre, es necesario realizar una adecuada gestión y análisis de los riesgos que pueden incurrir durante el ciclo de vida de un proyecto. Teniendo bien en cuenta el riesgo como la posibilidad de que existan consecuencias indeseables o inconvenientes, de un acontecimiento, cuya aparición no se puede determinar con anterioridad [2, 3], se estarán teniendo en cuenta también una de las acciones esenciales para conseguir un alcance exitoso en un proyecto software.

El presente trabajo se centra en realizar un análisis relacionado con la influencia de los riesgos sobre los métodos más reconocidos por la bibliografía para estimar proyectos de software: Puntos de Función, COCOMO II, Puntos por Casos de Uso, con vistas a identificar deficiencias y proponer mejoras en estimaciones futuras.

**Palabras clave:** Estimación, método de estimación, métrica, riesgos.

### **Abstract**

*It is the management process for the creation of a system or software which contains a set of activities, one of which is the estimate.*

*Project estimate is accepting a degree of uncertainty that the future holds a draft in question is also an essential basis of all other planning activities. [1]*

*To manage the uncertainty, there is a need for proper management and analysis of risks that may arise during the life cycle of a project. Taking proper account of risk as the possibility of undesirable consequences or disadvantages of an event whose occurrence can not be determined previously [2, 3], are also taking account of the actions essential to reach a a successful software project.*

*This work focuses on analysis related to the influence of risks on the most recognized in the literature for estimating software projects: Function Points, COCOMO II, by Use Case Points, to identify deficiencias and propose improvements in future estimates.*

**Key words:** Estimation, estimation method, metrics, risks.

### **Introducción**

El objetivo fundamental de la planificación de proyectos de software, es proporcionar un marco de trabajo, que permita hacer estimaciones razonables de recursos, costos y una planificación temporal [3]. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente a medida que progresa el proyecto. Además las estimaciones deberán definir los escenarios del mejor y peor caso, de modo que los resultados sean los más verídicos

posible. Estimar es tener una visión al futuro y aceptar cierto grado de incertidumbre. Al estimar no solo se analiza el procedimiento técnico utilizado, sino que se toman en cuenta los recursos, costos, tiempo y planificación.

Cuando se planifica un proyecto de software, se deben realizar estimaciones de esfuerzo humano, la duración cronológica del esfuerzo humano, del proyecto y del costo. La planificación se logra entonces, mediante un proceso de descubrimiento de la información que lleve a estimaciones razonables [3]. La planificación se convierte en una herramienta de posicionamiento anticipado, pues permite saber con anterioridad las fortalezas y debilidades que puede tener el proyecto, así como adecuarse a los cambios que puedan surgir en el mercado o en la sociedad.

El futuro es lo que preocupa, ¿qué riesgos podrían hacer que el proyecto fracase? El cambio es la preocupación ¿cómo afectarán los cambios en los requisitos del cliente, en las tecnologías de desarrollo, el proyecto y todas las entidades relacionadas con él, al cumplimiento de la planificación temporal y al éxito en general? Y por último la elección ¿qué métodos y herramientas se deberían emplear, cuánta gente debería estar implicada, qué importancia hay que darle a la calidad?[3]

Es una buena idea identificar los posibles riesgos, evaluar su probabilidad de aparición, estimar su impacto y así establecer un plan de mitigación o contingencia en caso que ocurra el problema [3] con vistas a estar preparados y tomar las medidas pertinentes para evitar la mayor cantidad de complicaciones que puedan ocasionar retrasos durante el proceso de desarrollo.

Se ha considerado necesario en el presente trabajo, realizar un análisis sobre los métodos que se utilizan para estimar proyectos de software en la actualidad, debido a las consecuencias que puede traer basar la planificación de la producción en estimaciones poco fiables. Se resalta el análisis de la influencia de los riesgos sobre las estimaciones, como una de las principales acciones al gestionar proyectos y por el alto impacto de las probabilidades de ocurrencia al modificar constantemente los resultados de las estimaciones, previendo la importancia de cuantificar los riesgos dentro del propio método a aplicar y así obtener resultados mucho más flexibles y cercanos a la realidad.

## **Desarrollo**

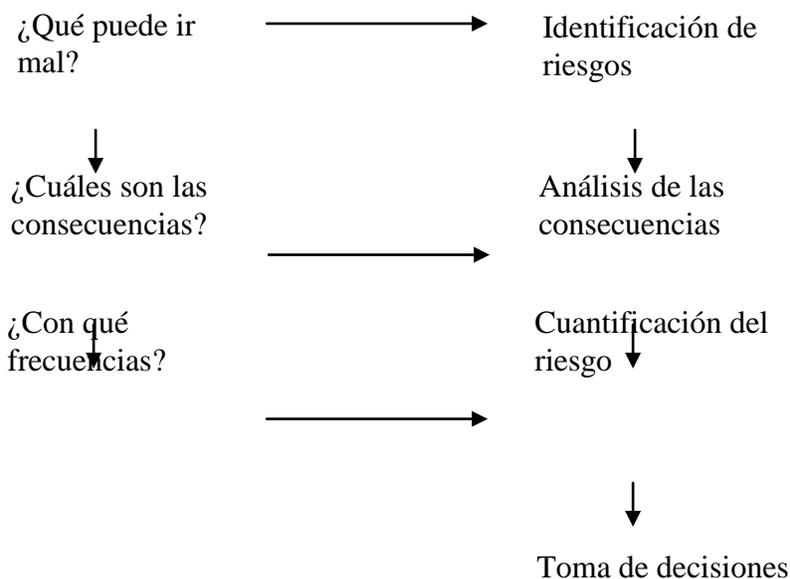
La estimación de la duración de las actividades que conforman el desarrollo de software es un tema que concierne a la gestión y control de proyectos para asegurar el éxito. En efecto, Ana María Moreno Sánchez Capuchino dice: La primera tarea en la gestión de proyectos es la estimación. [4]

“Cuando pueda medir lo que está diciendo y expresarlo con números, ya conoce algo sobre ello; cuando no pueda medir, cuando no pueda expresar lo que dice con números, su conocimiento es precario y deficiente: puede ser el comienzo del conocimiento, pero en tus pensamientos apenas estás avanzando hacia el escenario de la ciencia” [5].

Es una afirmación indiscutible que las mediciones son cruciales en el progreso de todas las ciencias. El progreso científico se logra a través de observaciones y generalizaciones basadas en datos y mediciones y la derivación de teorías como resultado de estas [6]. Sin la verificación a través de los datos y las mediciones, las teorías y las proposiciones permanecerían en un nivel abstracto.

La clave para manejar los riesgos es la medición y se sabe que negocios exitosos tienen mediciones exitosas. Una medida proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto.

El primer paso para gestionar los riesgos es reconocer que algo puede ir mal, luego analizar la probabilidad de ocurrencia y los efectos que puede traer en caso de que ocurra, para después darle prioridad al riesgo en función de la probabilidad de ocurrencia y del impacto sobre las estimaciones tanto de tamaño, coste y tiempo. Por último se desarrolla un plan para gestionar aquellos riesgos con gran probabilidad e impacto, que puede nombrarse Plan de Reducción, Supervisión y Gestión de Riesgos (RSGR) o un informe de riesgos[3].



Las dos características fundamentales que describen los riesgos son: incertidumbre y pérdida. Al analizar los riesgos es importante cuantificar el nivel de incertidumbre y el grado de pérdidas para cada riesgo en específico.

Se han considerado tres categorías para caracterizar los riesgos[3]:

Los riesgos del proyecto amenazan al plan del proyecto; si los riesgos se hacen realidad es probable que la planificación temporal del proyecto se retrase y que los costos aumenten. Los riesgos identifican los problemas potenciales de presupuesto, planificación temporal, personal, recursos, cliente y requisitos y su impacto en el proyecto de software.

Los riesgos técnicos amenazan la calidad y la planificación temporal el software que hay que producir. Si un riesgo técnico se convierte en realidad, la implementación puede llegar a ser difícil o imposible. Los riesgos técnicos ocurren porque el problema es más difícil de lo que se pensaba en un inicio.

Los riesgos del negocio amenazan la viabilidad del software a construir, poniendo a menudo en peligro el proyecto o el producto.

Los candidatos para los principales riesgos del negocio son [3]:

- Construir un producto o sistema excelente que no quiera nadie en realidad (riesgo de mercado).
- Construir un producto que no encaja en la estrategia comercial general de la compañía (riesgo estratégico).
- Perder el apoyo de una gestión experta debido a cambios de enfoque o a cambios de personal (riesgos de dirección).
- Perder presupuesto o personal asignado (riesgo de presupuesto).

Estas categorizaciones de los riesgos no siempre son las que se encuentran a diario, ya que algunos riesgos propios de la organización o del entorno de desarrollo simplemente son difíciles de predecir y clasificar.

Todo proyecto está expuesto a correr determinados riesgos que pueden ser identificados antes o en el transcurso de un proyecto determinado. Aún cuando se detectan los riesgos, debe existir la forma de corregirlos o asimilarlos para viabilizar el proyecto con vistas a la perfección de las estimaciones y una planificación mucho más realista y certera.

Considerando una de las características propias de un buen método de estimación, después de ser fácil de comprender, objetivo e usable, es precisamente que pueda ser flexible ante la influencia de posibles riesgos por los que pueda verse amenazado cualquier proyecto.

Existen numerosos métodos para estimar proyectos software entre los cuales se encuentran:

- Empíricos.
- Analógicos.
- Teóricos.

- Heurísticos.
- Las estimaciones global y detallada.
- Juicio del experto.
- Método Puntos de Casos de Uso.
- Modelo COCOMO.
- Puntos de Función.
- COCOMO II y los Puntos de Función.

Entre los métodos de estimación más recomendados por la amplia bibliografía y la experiencia de proyectos que los han implementado se encuentran:

### **Puntos de Función**

El método de puntos de función fue creado por Allan Albrecht y se basa principalmente en la identificación de los componentes del sistema informático en términos de transacciones y grupos de datos lógicos que son relevantes para el usuario en su negocio. A cada uno de estos componentes les asigna un número de puntos por función basándose en el tipo de componente y su complejidad; y la sumatoria de esto, da los puntos de función sin ajustar. El ajuste es un paso final basándose en las características generales de todo el sistema informático que se está contando.

Los objetivos de calcular Puntos de Función son:

- Medir lo que el usuario pide y lo que el usuario recibe.
- Medir atributos independientemente de la tecnología utilizada en la implantación del sistema.
- Proporcionar una métrica de tamaño que de soporte al análisis de la calidad y la productividad.
- Proporcionar un medio para la estimación del software.
- Proporcionar un factor de normalización para la comparación de distintos software.

El análisis de los Puntos de Función se desarrolla considerando cinco parámetros básicos externos del Sistema [3]:

1. Entrada (EI, External Input).
2. Salida (EO, External Output).
3. Consultas (EQ, External Query).
4. Ficheros Lógicos Internos (ILF, Internal Logic File).
5. Ficheros Lógicos Externos (EIF, External Interface File).

Con estos parámetros, se determinan los puntos de función sin ajustar (PFsA), luego se utiliza COCOMO para calcular el esfuerzo de desarrollo y otros indicadores a partir de la conversión de los puntos de función sin ajustar en líneas de código fuente o casos de uso. A este valor, se le aplica un Factor de Ajuste obtenido en base a unas valoraciones subjetivas sobre la aplicación y su entorno; es decir, las características generales del sistema. Se debe aplicar el coeficiente de conversión de acuerdo a la experiencia para obtener el esfuerzo de desarrollo.

Podemos decir entonces que los puntos de función aparecen con ventajas substanciales por sobre las líneas de código, para fines de estimación temprana del tamaño del software, y por ende, del esfuerzo de desarrollo. Además es una medida ampliamente utilizada, y con éxito, en muchas organizaciones que desarrollan software en forma masiva.

### **COCOMO II y los Puntos de Función**

Debido a la complejidad de los proyectos de software, el modelo original COCOMO, fue modificado, denominándose al modelo actual COCOMO II [7]. El nuevo modelo permite determinar el esfuerzo y tiempo de un proyecto de software a partir de los puntos de función sin ajustar, lo cual supone una gran ventaja, dado que en la mayoría de los casos es difícil determinar el número de líneas de código de que constará un nuevo desarrollo, en especial cuando se tiene poca o ninguna experiencia previa en

proyectos de software. Esto hace que ambos modelos, Puntos de Función y COCOMO sean perfectamente compatibles y complementarios. Su triunfo depende ampliamente de la adaptación del modelo a las necesidades de la organización, usando datos históricos; los cuales no siempre están disponibles.

### **Puntos de Casos de Uso**

Fue propuesto originalmente por Gustav Karner y posteriormente refinado por muchos otros autores. Este método exige la existencia de un modelo de casos de uso, por lo que se deberá comenzar a aplicar, una vez que se tenga algún entendimiento del dominio del problema o cuando se estén realizando las labores de arquitectura y dimensionamiento del tamaño del sistema. El método utiliza los actores y casos de uso identificados para calcular el esfuerzo que costará desarrollarlos [8]. A los casos de uso se les asigna una complejidad basada en transacciones, que son pares de pasos acción-usuario->respuesta-sistema de los escenarios de los casos de uso. A los actores se les asigna una complejidad basada en el tipo de actor, es decir, si son interfaces con usuarios o si son interfaces con otros sistemas (API o Protocolo).

Una vez asignada la complejidad, se calculan los puntos de caso de uso no ajustados, el TCF (factor de complejidad técnica) y el EF (factor del entorno). Con ellos, se calculan los puntos de caso de uso o UCP, que finalmente se traducen a esfuerzo en horas-hombre con un sencillo cálculo.

Una de las principales desventajas de este método es que no existe una teoría de cómo escribir o estructurar correctamente los casos de uso, por lo que todas las medidas de tamaño y estimación serán afectadas por la rigurosidad de los analistas.

Este método estima el esfuerzo de desarrollo de un producto de software a partir de los Casos de Uso y algunos factores de complejidad técnica y ambiente que influyen en el desarrollo, que de cierta forma y en dependencia de la criticidad de los especialistas pueden considerarse riesgos que afectan al proyecto y el método los cuantifica para afinar el resultado final.

Resulta importante destacar que aunque estos métodos de estimación de alguna forma tienen en cuenta la influencia de posibles riesgos (riesgos técnicos y de ambiente solamente), no brindan la posibilidad de cuantificar el resto de los posibles riesgos categorizados en un inicio, lo que puede incurrir a las incongruencias en las estimaciones iniciales que se encuentran a menudo los proyectos de software, que son pactados sin tener en cuenta la importancia de estas acciones en cuanto a las estimaciones.

Varios reconocidos autores que han trabajado sobre investigaciones relacionadas con las gestión de proyectos han documentado y aportado muchísima información referente a los riesgos, pero se considera que aún no se ha logrado vincular en un método de estimación, métricas que cuantifiquen el impacto de los riesgos sobre las estimaciones, aún cuando estas métricas existen.

Existen tres factores que afectan las consecuencias probables de un riesgo en caso de ocurrir: su naturaleza, su alcance y cuando ocurre [3].

La naturaleza del riesgo indica los problemas probables que aparecerán si ocurre. Por ejemplo, una interfaz externa mal definida para el hardware del cliente (un riesgo técnico) impedirá un diseño y pruebas tempranas y probablemente lleve a problemas de integración más adelante en el proyecto. El alcance de un riesgo combina la severidad (¿Cómo de serio es un problema?) con su distribución general (¿Qué proporción del proyecto se verá afectado y cuántos clientes se verán perjudicados?). Finalmente, la temporización de un riesgo considera cuándo y por cuánto tiempo se dejará sentir el impacto.

Estudios realizados por las Fuerzas Aéreas recomiendan primeramente determinar la probabilidad media de que ocurra un valor para cada componente (Rendimiento, Soporte, Coste, Planificación Temporal) de riesgo, determinar el impacto de cada componente basándose en criterios definidos con anterioridad y luego analizar los resultados.

Existe una métrica definida que mide la exposición al riesgo en general ER, utilizando la siguiente expresión [3]:

$$ER = P \times C$$

donde P es la probabilidad de que ocurra un riesgo y C es el coste del proyecto si el riesgo ocurriese.

Un ejemplo que ilustre la aplicación de esta métrica puede ser en un equipo de proyecto que define un riesgo para el proyecto de la siguiente manera [3]:

**Identificación del riesgo:** Solo el 70 % de los componentes del software planificados para reutilizarlos pueden, de hecho, integrarse en la aplicación. La funcionalidad restante tendrá que ser desarrollada de un modo personalizado.

**Probabilidad del riesgo:** 80% de probabilidad.

**Impacto del riesgo:** 60 componentes de software reutilizables fueron planificados. Si solo el 70% puede usarse, 18 componentes tendrán que desarrollarse improvisadamente (además de otro software personalizado que ha sido planificado para su desarrollo). Puesto que la media por componentes de 100 líneas de código (LDC) y los datos locales indican que el costo de la ingeniería del software para cada LDC es de \$14, 00; el costo global (impacto) para el desarrollo de componentes sería  $18 \times 100 \times 14 = \$ 25, 200$

**Exposición al riesgo:**  $ER = 0,80 \times 25, 200 = \$ 20, 200$

Debido a la importancia sobre las estimaciones de tamaño y costo que se le tributa a esta métrica es recomendable incluirla en el método de estimación; por la posibilidad que brinda de ajustar las estimaciones, previendo los riesgos en dependencia a las características propias del proyecto a desarrollar y cuantificándolos lo más pronto posible. Evidentemente el análisis de los riesgos sobre las estimaciones no se detiene en la evaluación del impacto del riesgo. En un segundo momento estos resultados serán analizados y evaluados, ya que los riesgos van sucediendo lo mismo a largo que a corto plazo, lo que conlleva a análisis reiterados a medida que progresa el proyecto, tomando medidas correctivas al respecto.

### Conclusiones

Una vez analizada la influencia de los riesgos sobre los métodos para estimar proyectos de software, se arriba a las siguientes conclusiones:

- Los métodos de estimación analizados no cuantifican todos los riesgos posibles a detectar en cualquier proyecto software. Solo el método de Puntos de Casos de Uso viabiliza factores de complejidad técnica y factores ambiente, pero no se consideran que todos los riesgos puedan incluirse dentro de estas clasificaciones de factores.
- La gestión de los riesgos no debe quedarse como una acción en paralelo a la aplicación del método de estimación a seleccionar, sino que para realizar estimaciones razonables deben aplicarse métricas que cuantifiquen el impacto de los riesgos sobre las estimaciones, para posteriormente realizar los análisis y las evaluaciones pertinentes sobre el desarrollo de la planificación temporal de los proyectos.

### Referencias Bibliográficas

1. Cao, M.I.J.I. "Principios para un método de estimación de proyectos de software basado en los escenarios principales".
2. Nuchera, A.H., "Una introducción a la gestión de riesgos tecnológicos". 2004.
3. Pressman, R.S., "Ingeniería de Software. Un enfoque práctico". 2005.
4. Capuchino, A.M.M.S., "Control y gestión de proyectos software, Unidad 4: Estimación de Proyectos Software". 1996.
5. Kelvin, L., "Capacitación de Vanguardia para Desarrolladores" 2000.
6. Kan., S.H., "Metrics and Models in Software Quality Engineering". 2000.
7. Ovejero, I.J.D. "Estimación de proyectos para sistemas basados en conocimiento".
8. Hernández, S.E.B. (2002) "Métricas de estimación de tamaño Puntos de Caso de Uso".