

Fundamentos de Ingeniería de software

Gestión

José Luis López Martínez

Universidad Autónoma de Yucatán

1. Introducción
2. Definiciones
3. Parámetros principales
4. Importancia
5. Recomendaciones
6. Etapas de la gestión de proyectos
7. Categorías
8. Estimación mediante juicio de expertos

Introducción

Sacar del equipo a una persona que rinde a un mal nivel puede ser más productivo que añadir a otra muy buena -Tom DeMarco

Muchos de los problemas que afectan a la Ingeniería de Software están relacionados con la:

- Gestión de recursos (personal)
- Licitación de proyecto
- Ejecución del proyecto
- Estimación de esfuerzo y plazos
- Correcta elección de la metodología de desarrollo
- Selección de la tecnología adecuada

Definiciones

Gestión de proyectos

Es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para cumplir los requisitos del mismo.

Gestión de proyectos en la Ingeniería de software

Se puede definir como la aplicación de las actividades de gestión (planificación, coordinación, medición, monitorización, control y realización de informes) para asegurar que el desarrollo y el mantenimiento del sw se realiza de una forma sistemática, disciplinada y cuantificable.

Costo de desarrollo

Determina la viabilidad y el éxito de un proyecto de desarrollo de software. Un proyecto se considera exitoso, si los ingresos obtenidos del proyecto completado son más que el costo incurrido. La estimación del costo es necesaria para determinar la viabilidad del proyecto. También ayuda a cotizar el costo del proyecto o cuánto dinero se debe cobrar para desarrollar el software para un cliente.

Esfuerzo

El costo del desarrollo de software es una función del esfuerzo requerido para desarrollar el software.

El esfuerzo se expresa en términos de persona-meses (PM). La ejecución de cualquier trabajo generalmente requiere el esfuerzo de varias personas. Suponga que tres personas dedican 3 meses, 2 meses y 1 mes de su tiempo, respectivamente, para hacer un trabajo, entonces el esfuerzo total utilizado en el trabajo es de 6 PM.

Recurso

El recurso humano es el principal recurso de un proyecto de desarrollo de software.

El desarrollo de software es trabajo de conocimiento. Requiere el esfuerzo de varias personas, como ingeniero de software, analista de sistemas, programador, tester, etc.

Cada uno de estos miembros puede tener diferentes tipos y niveles de experiencia / habilidades. La estimación de los recursos implica la determinación del número de personas de diferentes categorías en función de los niveles y tipos de habilidades / conocimientos informáticos necesarios para el desarrollo de software.

Como a las personas altamente calificadas se les paga más, el costo del proyecto de software depende no solo de la cantidad de recursos humanos sino también de su nivel de habilidad.

Tiempo

La duración de un proyecto es un criterio importante para el éxito de cualquier proyecto.

La duración del proyecto tiene relación con el costo, el esfuerzo y los recursos. A veces, al asignar más personas al proyecto o al incurrir en más costos, es posible reducir la duración del proyecto hasta cierto punto. Sin embargo, la relación no es lineal. El tiempo necesario para completar cualquier trabajo de conocimiento no se reduce necesariamente al agregar más personas.

Agregar más personas a un proyecto tardío lo retrasará

- Barry W Boehm

Estimación

Es el proceso de determinar una variable no conocida a partir de otras conocidas.¹

Buena estimación

Una buena estimación es aquella que proporciona una visión lo suficientemente clara de la realidad del proyecto como para permitir que el liderazgo del proyecto tome buenas decisiones sobre cómo controlar el proyecto para alcanzar sus objetivos.²

¹Ingeniería de software, Sánchez, S., et. al, (2012)

²Software estimation. McConnell S. (2006)

Parámetros principales

Los gestores deben de mantener un equilibrio de tres parámetros principales:

- Funcionalidad que el sistema en construcción ha de proporcionar
- El plazo en que se debe de desarrollar
- Los recursos de los que dispone (económicos, humanos, herramientas, métodos, etc)

Parámetros principales

Estos parámetros no son independientes sino que se influyen mutuamente, los cuales forman lo que se conoce como el triángulo mágico en la gestión de proyectos.



Figura 1: Triángulo mágico en la gestión de proyectos

Importancia

- Estimación de costes, plazos y esfuerzo

Parte esencial de la planificación de cualquier tipo de proyectos

No es una tarea fácil ya que es una tarea crítica

- Subestimar supondrá pérdidas para la organización y en un caso extremo, el proyecto tendrá que ser cancelado
- Sobreestimación, por el contrario, lleva a lo que se conoce humorísticamente como la ley de Párkinson -El trabajo se expande hasta rellenar todo el volumen disponible-

Recomendaciones

Recomendaciones

- Para llevar a cabo una correcta estimación de coste y esfuerzo es necesario establecer con claridad los objetivos del proyecto, procurar que los requisitos estén bien especificados y recabar toda la información disponible.
- En cuanto los métodos de estimación, existen varios y no puede decirse que unos sean mejores que otros o que haya uno que aporte los mejores resultados.
- Es habitual aplicar más de uno en un mismo proyecto, unos métodos pueden complementar otros.
- La mayoría de los métodos de estimación utilizan datos del proyecto anteriores para calcular las nuevas predicciones.

Etapas de la gestión de proyectos

La estimación correcta de los parámetros anteriores es la base de la planificación del proyecto. La estimación incorrecta es una de las principales causas de falla del proyecto de software.

La gestión de proyectos se lleva a cabo a través de estos procesos:

1. Iniciación y alcance del proyecto
2. Planificación del proyecto
3. Seguimiento del proyecto
4. Revisión y evaluación del proyecto
5. Cierre del proyecto

Categorías

A continuación se presentan las principales categorías para realizar técnicas de estimación:

- Estimación basada en juicio de expertos (basada en experiencia y conocimiento de las personas)
- Modelos algorítmicos o paramétricos (basadas en Ecuaciones)
- Otros (basadas en IA) .

Estimación mediante juicio de expertos

Estas técnicas suelen utilizarse como confirmación de los resultados de otras técnicas de estimación o como factor de corrección. Funciona como una caja negra, pues no se obtiene detalladamente cómo se ha llegado a la estimación.

- Delphi de banda ancha (Wideband Delphi)
- Reseñas grupales (Group Reviews)
- Estimación por analogía (Estimation by Analogy)

Wideband Delphi

- Se basa del método Delphi el cual fué desarrollado a finales de 1940 y fué utilizado en la predicción de tendencias tecnológicas
- En 1981 Barry Boehm amplió la técnica de Delphi a Delphi de banda ancha (mayor interacción y comunicación entre los participantes). Es una técnica estructurada.
- El equipo de estimación está compuesto por el lider del proyecto, el moderador, los expertos y los representantes del equipo de desarrollo, lo que constituye un equipo de 3 a 7 miembros³
- Se puede utilizar en estimaciones iniciales de un proyecto donde se carece de datos históricos para realizar las estimaciones de forma más precisa, y donde varias disciplinas estarán involucradas en el proyecto.

³<https://www.tutorialspoint.com>

Técnica de Delphi de banda ancha⁴

1. El coordinador Delphi presenta a cada estimador la especificación y un formulario de estimación.
2. Los estimadores preparan estimaciones iniciales individualmente.
3. El coordinador convoca una reunión de grupo en la que los estimadores discuten cuestiones de estimación relacionadas con el proyecto en cuestión. Si el grupo acuerda una sola estimación sin mucha discusión, el coordinador asigna a alguien para que haga de abogado del diablo.
4. Los estimadores entregan sus estimaciones individuales al coordinador de forma anónima.
5. El coordinador prepara un resumen de las estimaciones en un formulario de iteración y presenta el formulario de iteración a los estimadores para que puedan ver cómo se comparan sus estimaciones

⁴Software estimation.McConnell S. (2006)

Técnica de Delphi de banda ancha⁵

6. El coordinador hace que los estimadores se reúnan para discutir las variaciones en sus estimaciones.
7. Los estimadores votan de forma anónima sobre si quieren aceptar la estimación promedio. Si alguno de los estimadores vota “no”, regresa al paso 3.

La estimación final es la estimación de un solo punto resultante.

⁵Software estimation.McConnell S. (2006)

Wideband Delphi

Wideband Delphi Estimation Sheet						
Project: <Project Name>			Estimation Units: Person Hours			
Estimation Team Member: <Name>			Date: <MM-DD-YY>			
Task	Initial Estimate	Change 1	Change 2	Change 3	Change 4	Final
Task1	n_1					
Task2	n_2					
Task3	n_3					
Task4	n_4					
Task5	n_5					
Task6	n_6					
Task7	n_7					
Task8	n_8					
Net Change						
Total		$\sum n_i$				

Figura 2: Hoja de estimación ⁶

⁶<https://www.tutorialspoint.com/>

Wideband Delphi

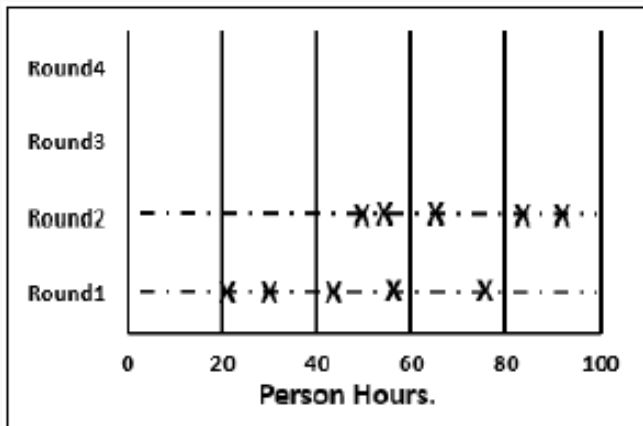


Figura 3: Rango de estimación ⁷

⁷<https://www.tutorialspoint.com/>

¿Cuáles son las ventajas y desventajas de esta técnica?

Revisiones grupales

Revisiones grupales

Una estrategia sencilla para aumentar la precisión de las estimaciones hechas por individuos es involucrar a un grupo en su revisión. La estrategia sigue tres pasos

1. Solicite a cada miembro del equipo que realice cálculos individuales de las partes del proyecto y luego organice una reunión para comparar sus estimaciones.
2. No se limite a calcular un promedio de las estimaciones y darlas por válidas.
3. Busque alcanzar una estimación consensuada que sea aceptada por todo el grupo. En caso de un estancamiento, evite tomar decisiones mediante votación; en su lugar, fomente la discusión de las diferencias y busque la aceptación unánime de todos los miembros del grupo.

Estudios indican que se sugiere utilizar de 3 a 5 expertos con diferentes perfiles o roles. Esta estrategia es útil para mejorar la precisión de las estimaciones.

Estimación por analogía

Estimación de expertos por analogía

Esta técnica consiste en realizar la estimación del proyecto actual comparando dicho proyecto con otros anteriores., de características y ámbitos similares.

1. Obtenga datos detallados sobre tamaño, esfuerzo y costos de un proyecto previo similar. Si es factible, descomponga por por características específicas, por ejemplo EDT (Estructura por desglose de trabajo).
2. Compare el tamaño de cada componente del nuevo proyecto con el proyecto anterior.
3. Estime el tamaño del nuevo proyecto como un porcentaje del tamaño del proyecto anterior.
4. Calcule una estimación del esfuerzo basada en la relación entre el tamaño del nuevo proyecto y el tamaño del proyecto anterior.
5. Asegúrese de que las suposiciones utilizadas en ambos proyectos, tanto el nuevo como el anterior, sean consistentes y coherentes.

- Una de las desventajas de este método es que la organización debe tener una base de datos historicos de proyectos realizados
- En posible utilizar un repositorio comercial en el cual se encuentren gestión de proyectos en diversos sectores como son telecomunicaciones, gobierno, etc., y en el cual se pudiese encontrar proyectos similares a los nuestros en relación a tamaño, tipos de base de datos, etc y por lo tanto predecir el esfuerzo.⁸
- International Software Benchmarking Standards Group (ISBSG) ⁹

⁸Ingeniería de software, Sánchez, S., et. al, (2012)

⁹<https://www.isbsg.org/>

Modelos algorítmicos o paramétricos

Modelos de estimación algorítmicos o paramétricos

Estos modelos de estimación establecen relaciones, a menudo en forma de ecuaciones, entre variables, típicamente el esfuerzo y el tamaño del software, pero también pueden abarcar otros aspectos del proyecto, como la complejidad o factores vinculados a la estructura organizativa del proyecto.

- IFPUG-UFP (Unadjusted Function Points)-Puntos de función no ajustado
- COSMIC-FFP (COSMIC Full Function Points)
- COCOMO (COConstructive COSt MOdel)-Modelo Constructivo de costos
- SLIM (Software Life Cycle Model)

Puntos de función (Function Point)

- Los Puntos de función (PF) es una de las técnicas más utilizadas en la estimación del software¹⁰
- Creado por Allan Albretch en 1979 en IBM
- Es una unidad de medida para llegar al tamaño de una aplicación de SW
- Mide la funcionalidad de un sistema de software desde la visión del usuario
- Independiente de la tecnología y lenguaje de programación
- Proporciona un indicador fiable del tamaño del software en etapas tempranas
- En 1990, IFPUG (Grupo Internacional de Usuarios de Puntos de Función) realiza una variación del modelo, surge UFP.
- Otra variante de este modelo es el propuesto por COSMIC
(Common Software Measurement International Consortium)

¹⁰Estimating Software-Intensive Systems: Projects, products and processes, Stutzke, R. (2005)

- Entradas externas (EI-external Input) : Entradas de usuario, ejemplo datos ingresados desde una pantalla, datos obtenidos por un sensor, formulario de registro
- Salidas externas (EO-external output) : Información generada por el sistema y que se muestra al usuario, ejemplo un informe de ventas mensuales, mensajes de error, mensajes de confirmación
- Consultas (EQ-external query): Entradas interactiva que requieren una respuesta del sistema, por ejemplo consultar registros de una base de datos
- Archivos internos (ILF-internal logical file): Los archivos que son manejados por el sistema, por ejemplo, una base de datos que almacena información de un inventario
- Archivos externos (EIF-external interface file): Interfaces con otros sistemas, por ejemplo un sistema que importa o exporta datos a un sistema de contabilidad externo

<i>Function Types</i>	<i>IFPUG Function Points (1984)</i>			<i>SPR Feature Points (1985)</i>
	<i>Low</i>	<i>Average</i>	<i>High</i>	
Input	3	4	6	4
Output	4	5	7	5
Inquiry	3	4	6	4
Internal logical files	4	10	15	7
External interface files	5	7	10	7
Algorithm	None			3

Figura 4: Factor de ponderación según la complejidad ¹¹

$$PF_{NoAjustados} = \sum_{i=1}^{15} numElementos_i.peso_i$$

¹¹Estimating Software-Intensive Systems: Projects, products and processes, Stutzke, R. (2005)

	Low	Average	High	Suma
Input(EI)	2x3	+ 4x4	+ 1x6	= 28
Output(EO)	0x4	+ 6x5	+ 2x7	= 44
Inquiry(EQ)	0x3	+ 8x4	+ 0x6	= 32
Internal logical files(ILF)	0x4	+ 2x10	+ 0x15	= 20
External Interface files(EIF)	0x5	+ 2x7	+ 0x10	= 14
Total de PF no ajustados				138

Figura 5: Ejemplo de puntos de función no ajustados

Posteriormente se realiza un factor de corrección, llamada PCA(Processing Complexity Adjustment)

$$PCA = 0.65 + (0.01 \sum_{i=1}^{14} F_i)$$

1	Comunicaciones de datos	8	Actualización <i>on-line</i>
2	Datos o procesamiento distribuido	9	Procesamiento complejo
3	Objetivos de rendimiento	10	Reutilización
4	Configuración usada masivamente	11	Facilidad de operación
5	Tasa de transacción	12	Facilidad de instalación y conversión
6	Entrada de datos <i>on-line</i>	13	Puestos múltiples
7	Eficiencia para el usuario	14	Facilidad de cambio

Figura 6: Factores de complejidad técnica ¹²

A cada factor se le asigna un valor entre 0 y 5, donde el 0 significa que es un factor es irrelevante para el proyecto y 5 que es un factor esencial para el proyecto. Supongamos que para nuestro proyecto de ejemplo se obtuvo un factor de complejidad de 52. Utilizando la ecuación de PCA.

$$PCA = 0.65 + (0.01 \sum_{i=1}^{14} F_i) = 1.17$$

¹²Ingeniería de software:Un enfoque desde la guía del SWEBOK,Sánchez.S,et. al.(2012)

$$PF_{ajustados} = PF_{NoAjustados} \times PCA$$

$$PF_{ajustados} = 138 \times 1.17 = 161.46$$

A partir de los PF se úeden calcular las líneas de código (LoC) y el esfuerzo basado en la productividad del equipo de desarrollo.

<i>Lenguaje de programación</i>	<i>Media LoC/PF</i>
Lenguaje ensamblador	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
Java, C++	53
Generadores de código	15
Hojas de cálculo	6
Lenguajes gráficos	4

Figura 7: Lenguajes y LoC promedio por PF ¹³

Supongamos que utilizaremos Java, y nuestros $PF = 161.46$, entonces

$$LoC = 53 \times PF = 53 \times 161.46 = 8,557$$

líneas de código en nuestro proyecto

¹³Ingeniería de software:Un enfoque desde la guía del SWEBOK,Sánchez.S,et. al.(2012)

Supongamos que la productividad del equipo de desarrollo es de 12 PF/personas-mes, y el costo promedio de tener el equipo de desarrollo es de 50 mil pesos mensuales.

Cada Punto de función tiene un costo de $(50,000/12) = 4,166$ pesos

El costo del proyecto sería $161 \times 4,166 \approx 670,000$ pesos

El esfuerzo es

$$esfuerzo = PF/productividad = 161/12 = 13.41$$

personas-mes

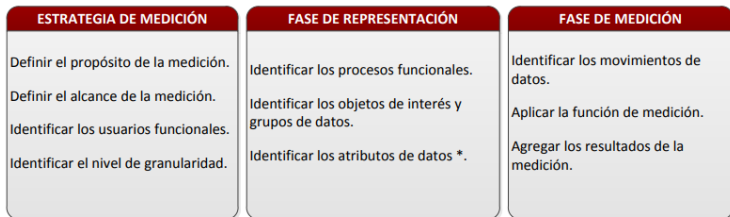
Cosmic

Cosmic FFP (Full Function Points)

Cosmic es una metodología estructurada que consta de tres fases

1. Estrategía de medición
2. Fase de representación o mapeo
3. Fase de medición

COSMIC fue diseñado para abordar requisitos funcionales en cualquier nivel de la arquitectura de software y en cualquier grado de descomposición de componentes.



(*) Opcional

Figura 8: Fases de Cosmic-FFP ¹⁴

¹⁴https://leda-mc.com/wp-content/uploads/2016/11/De_IFPUG_a_COSMIC.pdf

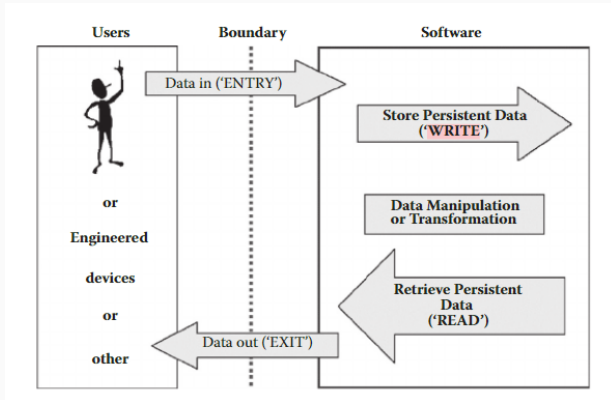


Figura 9: Modelo COSMIC FFP¹⁵

¹⁵COSMIC Function Points, Dumke,R.,Abran, A. (2016)

Ejemplo

RF-003-Facturación de Pedidos pendientes¹⁶

Desde la pantalla de pedidos pendientes, el usuario podrá seleccionar uno o varios pedidos para ser facturados. Se podrá facturar un pedido individual o un lote de pedidos. El usuario podrá escoger entre emitir una factura por pedido o unificar las facturas por cliente. Una vez facturado, se cambiará el estatus del pedido a “facturado”, de esta forma, al refrescar la lista de pedidos pendientes este no será mostrado.

¹⁶<http://www.pmoinformatica.com/2018/03/ejemplos-de-estimacion-de-costos-de-un-proyecto-de-software-COSMIC.html>

Proceso funcional : Facturar pedidos pendientes Movimientos de datos con 10 CFP:

- Entrada: Seleccionar pedido a facturar.
- Entrada: Seleccionar facturación agrupada por cliente o individual por pedido.
- Entrada: Iniciar proceso de facturación por medio de botón.
- Lectura: Leer de la base de datos los datos de pedidos seleccionados para facturación.
- Lectura: Leer las líneas de pedido.
- Escritura: Crear un registro de factura para cada pedido (Facturación individual).
- Escritura: Crear registro de líneas de factura individual.
- Escritura: Agrupar pedidos de un mismo cliente y crear registro de factura (Facturación por cliente).
- Escritura: Crear registro de líneas de factura por cliente.
- Escritura: Cambiar el estatus de pedido a facturado

Cocomo

COCOMO (Constructive Cost Model) Modelo Constructivo de Costos
Propuesto por Barry W. Boehm en 1981 y actualizado en 1995 con el
nombre de COCOMO II.

COCOMO se basa en la premisa de que el esfuerzo y el costo de un
proyecto de software están directamente relacionados con el tamaño del
software y otros factores, como la complejidad, la tecnología utilizada y el
entorno del proyecto.

En 2012, se desarrolla el COCOMO III, el cual considera aspectos
importantes, como la reutilización de componentes, la madurez del
proceso de desarrollo, la experiencia del personal y la flexibilidad en la
asignación de recursos.

COCOMO I se basa en la cantidad de líneas de código de software (LoC) como un indicador clave del tamaño del software. En el modelo inicial se distinguen tres tipos de proyectos:

- Orgánico : Proyectos estables, pequeños y relativamente simples
- Semiacoplado : Proyectos de tamaño medio
- Incrustados: Proyectos grandes y complejos con requisitos cambiantes

El modelo COCOMO se compone de tres submodelos : Básico., Intermedio , Detallado

- Básico

$$Esfuerzo = a.KLOC^b$$

(KLOC miles de líneas de código)

$$Tiempo = c.Esfuerzo^d$$

$$Personal = Esfuerzo / Tiempo$$

Ejemplo

Se requiere realizar un sistema de facturación del cual se tiene experiencia (por lo tanto, lo clasificamos como orgánico).

Adicionalmente, se estima que el número de LoC es de 50,500. Se tiene la siguiente tabla de parámetros

	a básico	a intermedio	b	c	d
Orgánico	2.4	3.2	1.05	2.5	0.38
Semiacoplado	3	3	1.12	2.5	0.35
Incrustado	3.6	2.8	1.2	2.05	0.32

Figura 10: Parámetros de COCOMO¹⁷

$$Esfuerzo = a.KLOC^b = 2.4 \times 50.5^{1.05} = 147.45 \text{ personas} - \text{mes}$$

$$Tiempo = c.Esfuerzo^d = 2.5 \times (147.45)^{0.38} = 16.67 \text{ meses}$$

$$Personal = Esfuerzo / Tiempo = 147.45 / 16.67 \approx 9 \text{ personas}$$

¹⁷Ingeniería de software:Un enfoque desde la guía del SWEBOK,Sánchez.S,et. al.(2012)

Modelos basados en la Inteligencia Artificial (IA)

En los últimos años se ha utilizado técnicas de IA en la estimación de proyectos en la Ingeniería de software

- Redes neuronales
- Sistemas basados en reglas
- Árboles de decisión
- Otros

Evaluación de modelos

- Magnitud media del error relativo

$$MMER = \frac{1}{n} \sum_{i=1}^n \left| \frac{e_i - \hat{e}_i}{e_i} \right|$$

- Técnicas estadísticas de correlación