

Tema 5: Estimación

GESTIÓN Y MANEJO DE PROYECTOS INFORMÁTICOS
MÁSTER EN INGENIERÍA INFORMÁTICA

Fernando Berzal Galiano



Índice

- El ámbito del proyecto
- ¿Qué es una estimación?
- Cono de incertidumbre
- Realización de estimaciones
- Técnicas de estimación
 - Estimación por descomposición
 - Estimación del tamaño
 - Estimación con herramientas software
 - Modelos empíricos
 - COCOMO II
 - Modelo de Putnam
- Comentarios finales

El ámbito del proyecto

- La parte más difícil de construir un sistema de software es decidir precisamente qué construir.
- El ámbito del proyecto debe establecerse de forma no ambigua y comprensible para los distintos stakeholders del proyecto:
 - Gestores
 - Técnicos
 - Clientes
- Stakeholder
 - Interesado o parte interesada.
 - Personas u organizaciones afectadas por las actividades y las decisiones de una empresa.

El ámbito del proyecto

Preguntas que nos ayudan a establecer el ámbito del proyecto:

- **Contexto**

- ¿Cómo encaja el sistema en su entorno?

- **Objetivos:**

- ¿Qué resultados se esperan del proyecto?
- ¿Qué queda fuera del proyecto?

El ámbito del proyecto

Preguntas que nos ayudan a establecer el ámbito del proyecto:

- **Restricciones:**

- ¿Qué requisitos debe satisfacer el sistema?
- ¿Algún requisito no funcional (seguridad, rendimiento, fiabilidad...)?

El ámbito del proyecto

Una vez delimitado el ámbito del proyecto, el problema se descompone en partes:

- Casos de uso.
- Subsistemas / módulos.
- Funciones.
- ...

La descomposición inicial del proyecto ayudará y facilitará las tareas de planificación (entre ellas, la estimación de costes).

El ámbito del proyecto

El principio W⁵HH

- Permite la definición de las características clave del proyecto y el plan del proyecto resultante.
- Consta de las siguientes preguntas:
 - Why is the system being developed?
 - What will be done?
 - When will it be done?
 - Who is responsible for a function?
 - Where are they located organizationally?
 - How will the job be done technically and managerially?
 - How much of each resource is needed?

El ámbito del proyecto

Descripción del ámbito del proyecto

Una vez que los stakeholders están de acuerdo sobre cuál es el ámbito del proyecto se elabora un documento que recoja lo acordado en uno de los siguientes formatos:

- Una narración descriptiva (en pocas páginas)
- Un conjunto de casos de uso “esenciales”.

¿Qué es una estimación?

Estimación puntual

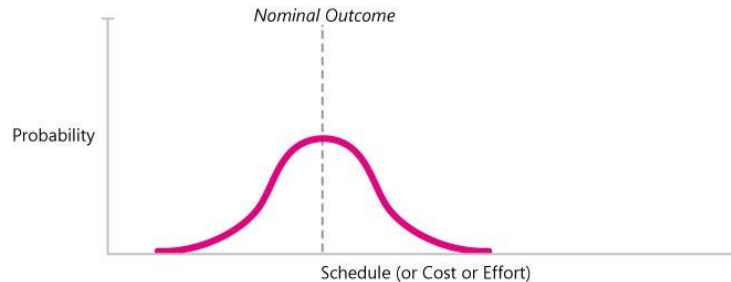


Inútil en estos casos:

Asume un 100% de probabilidad de que el resultado real sea exactamente igual que el esperado.

¿Qué es una estimación?

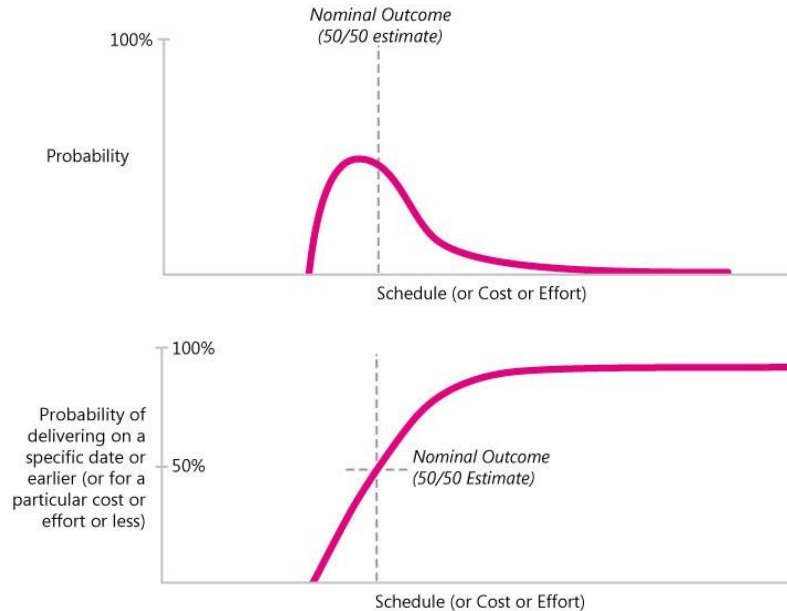
Estimación normal / gaussiana



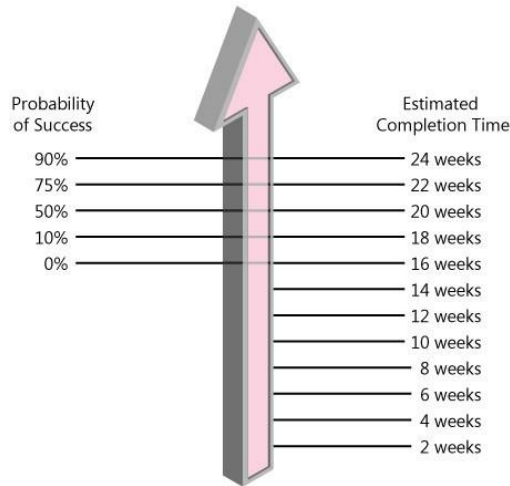
Basada en suposiciones incorrectas: existen límites en cuanto a la eficiencia con la que un equipo puede completar cualquier tarea que se le encomiende.

¿Qué es una estimación?

Estimación realista



¿Qué es una estimación?



Todas las estimaciones puntuales están asociadas a una probabilidad, explícita o implícitamente.

¿Qué es una estimación?

Consejos de Steve McConnell

- Distinga entre estimaciones, objetivos y compromisos.
- Cuando le pidan una estimación, determine si se espera de usted que realice una estimación o descubra una forma de lograr un objetivo.
- Cuando se encuentre con una estimación puntual, dude de que se trate de una estimación y compruebe si realmente se trata de un objetivo.
- Cuando vea una estimación puntual, su probabilidad no es del 100% (averigüe cuál es).

¿Qué es una estimación?

¿Qué podemos esperar de una buena estimación?

Una precisión del $\pm 10\%$ es posible,
pero sólo en proyectos bien controlados.

-- Capers Jones: *Estimating Software Costs*, 1998

Una precisión del $\pm 25\%$ el 75% del tiempo.

-- Conte, Dunsmore & Shen:
Software Engineering Metrics and Models, 1986

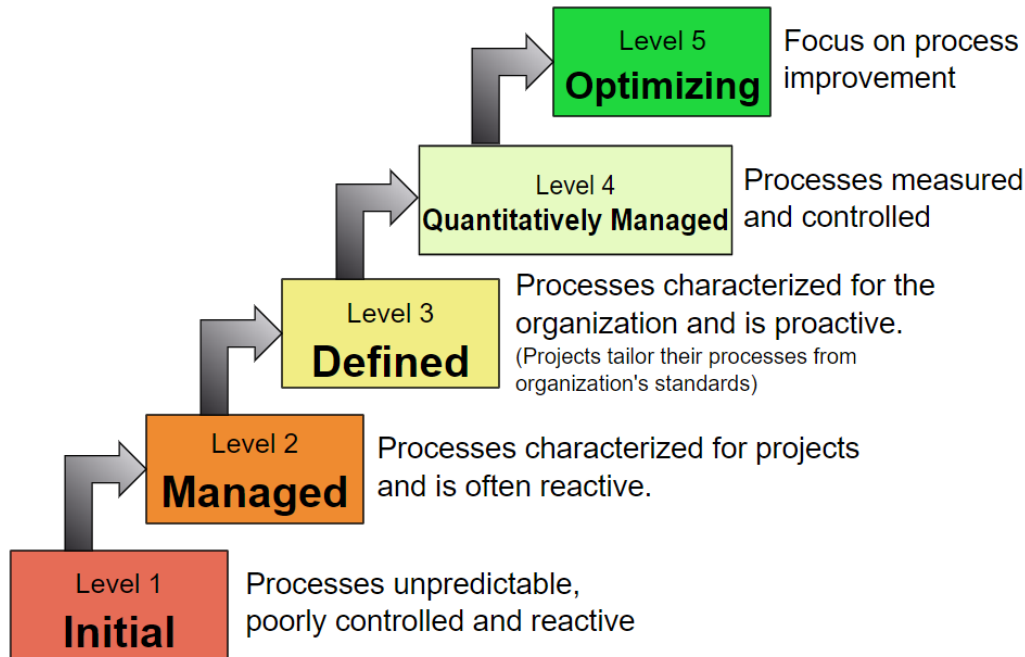
¿Qué es una estimación?

Capability Maturity Model (CMM)

- Administrado por el Instituto CMMI, una subsidiaria de ISACA, fue desarrollado en la Universidad Carnegie Mellon (CMU).
- Requerido en muchos contratos del Gobierno de los Estados Unidos, especialmente en el desarrollo de software.
- Guía la mejora de los procesos a través de un proyecto, una división o una organización entera.

¿Qué es una estimación?

Characteristics of the Maturity levels

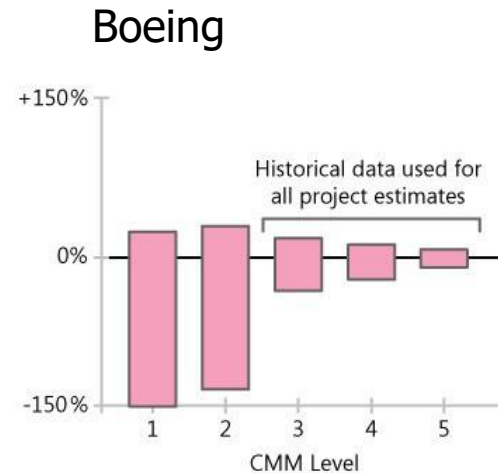
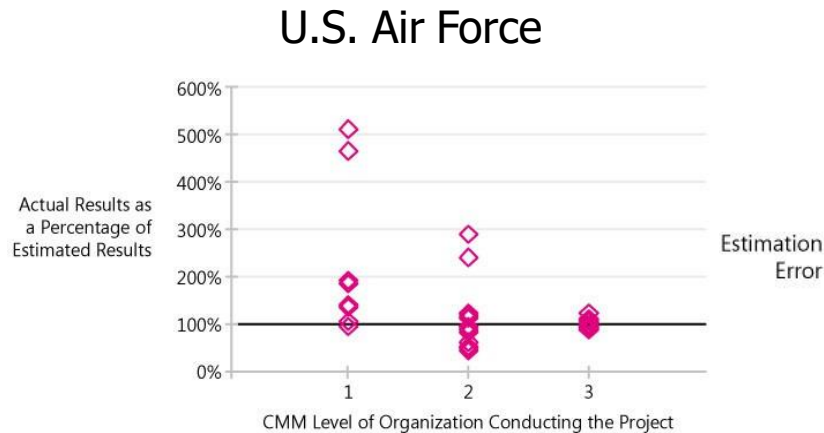


¿Qué es una estimación?



¿Qué es una estimación?

■ Resultados estimados vs nivel CMM



Source: "A Correlational Study of the CMM and Software Development Performance" (Lawlis, Fiowe, and Thordahl 1995).

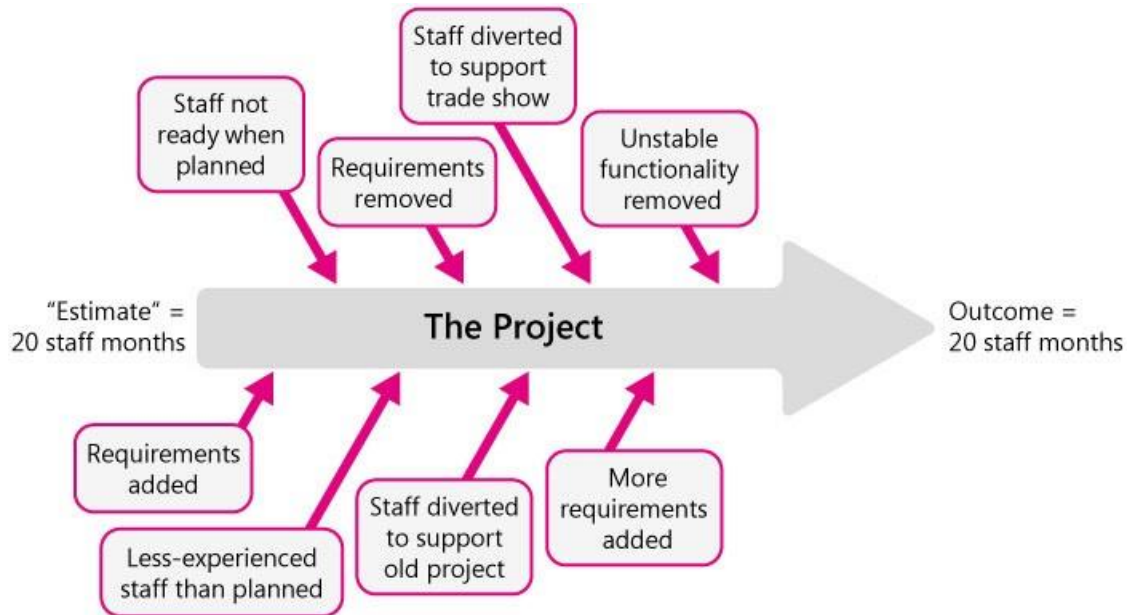
Steve McConnell:

Software Estimation: Demystifying the Black Art

Microsoft Press, 2006. ISBN 0735605351

¿Qué es una estimación?

Factores que influyen en la precisión de una estimación

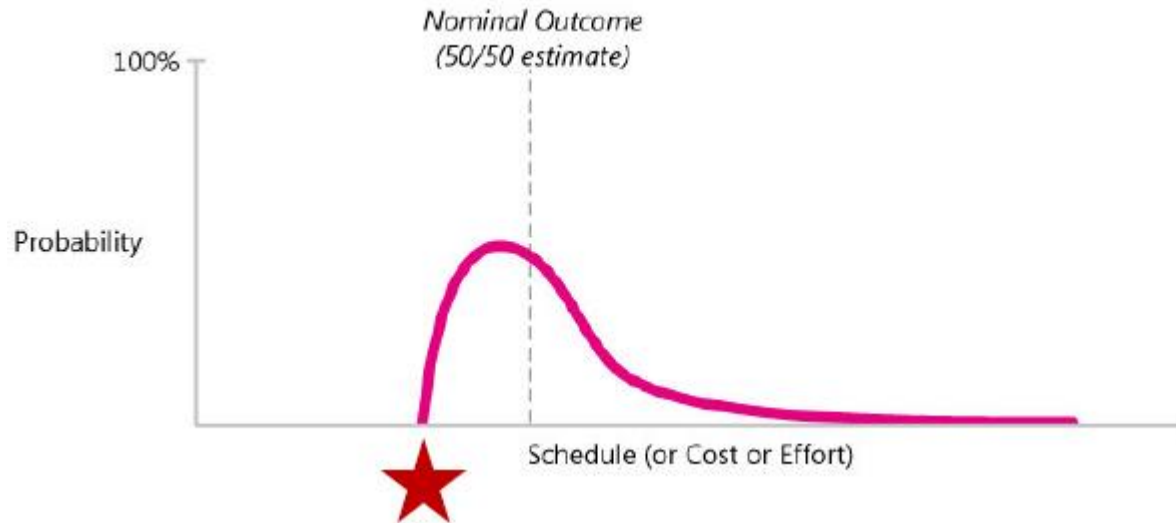


Algunos los podemos controlar, otros no...

¿Qué es una estimación?

- Definición más común de estimación:
 - La predicción más optimista que tiene una probabilidad no nula de hacerse realidad.

¿Qué es una estimación?



¿Qué es una estimación?

¿Qué es una buena estimación?

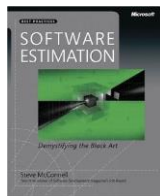
Una estimación que proporciona una vista suficientemente clara de la realidad del proyecto como para permitir al gestor tomar buenas decisiones sobre cómo controlar el proyecto para lograr sus objetivos.

-- Steve McConnell: Software Estimation, 2006

Test de autoevaluación

Estimación (intervalo)

[_____,_____]	Temperatura de la superficie del Sol
[_____,_____]	Latitud de Shanghai
[_____,_____]	Superficie de Asia
[_____,_____]	Volumen de los Grandes Lagos
[_____,_____]	Longitud de la línea costera del Pacífico
[_____,_____]	Libros publicados al año
[_____,_____]	Taquilla global de la película Avatar
[_____,_____]	Dólares en circulación en 2004
[_____,_____]	Dólares en circulación en 2019
[_____,_____]	Gasto público en España en 2018



Adaptado de...

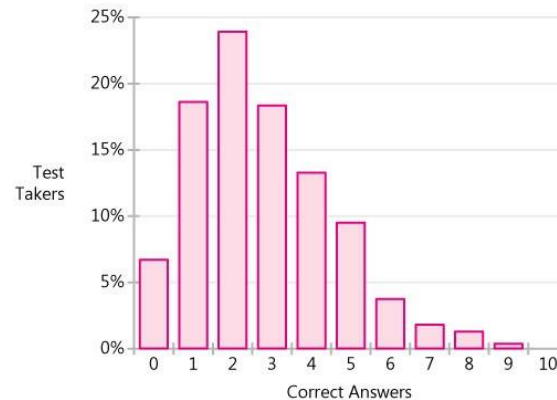
Steve McConnell:

Software Estimation: Demystifying the Black Art, 2006.

Test de autoevaluación

RESULTADOS

Si realmente estuviese realizando estimaciones con un 90% de confianza, debería haber acertado 9 de las respuestas...



Número medio de respuestas correctas: 2.8 → mala estimación

Test de autoevaluación

Consejos de Steve McConnell

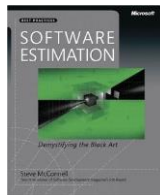
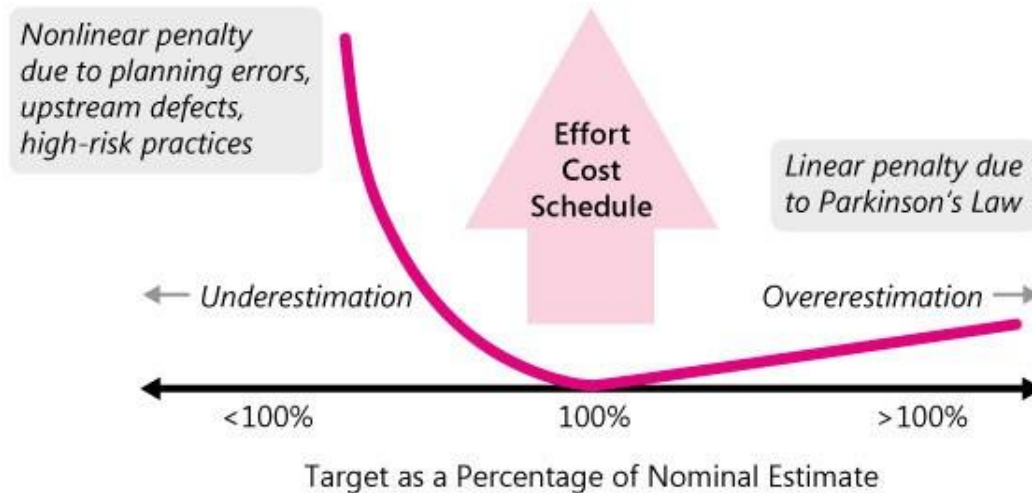
- No proporcione un porcentaje de confianza para sus estimaciones (p.ej. "90% de confianza") si no dispone de un **mecanismo cuantitativo** para derivarlo.
- Evite utilizar intervalos artificialmente estrechos: los intervalos deben representar su confianza en las estimaciones.
- Si siente presión para estrechar los intervalos de sus estimaciones, asegúrese de que la presión proviene de una fuente externa y no de usted mismo.

Test de autoevaluación

Causas más frecuentes del fracaso de un proyecto:

- Errores de estimación.
- Inestabilidad en los requisitos.

Estimaciones



Steve McConnell:
Software Estimation: Demystifying the Black Art
 Microsoft Press, 2006. ISBN 0735605351

Test de autoevaluación

Problemas de la sobreestimación:

- **Parkinson Law:** Si le das a un desarrollador 5 días para entregar una tarea que podría ser completada en 4 días, el desarrollador encontrará algo que hacer con el día adicional
- **Goldratt's "Student Syndrome" Law:** Si a los desarrolladores se les da demasiado tiempo, lo aplazarán hasta el final del proyecto, en cuyo momento se apresurarán a completar su trabajo, y probablemente no terminarán el proyecto a tiempo.

Test de autoevaluación

Problemas de la subestimación:

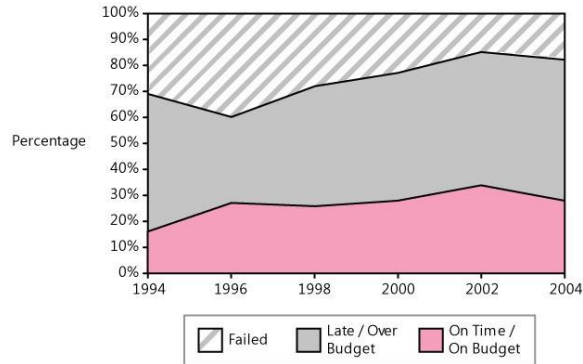
- Se reducen las probabilidades de acabar a tiempo.
- Se reduce la eficacia de los planes de proyectos.
- Se acaba dedicando menos tiempo al diseño y obtención de requisitos lo que luego genera problemas.
- El proyecto pasa a estar retrasado lo que implica que el equipo entre en dinámicas en las que no entraría de ir a tiempo.

Test de autoevaluación

Dinámicas destructivas típicas de un proyecto retrasado

- Más reuniones con la dirección para discutir cómo volver a poner el proyecto en marcha.
- Frecuentes reestimaciones, en las últimas fases del proyecto, para determinar cuándo se completará.
- Disculpas a los clientes clave por no cumplir con las fechas de entrega (incluyendo la asistencia a reuniones con esos clientes).
- Preparar comunicados provisionales para apoyar demostraciones de clientes, ferias comerciales, etc.
- Tratar con código de implementación deficiente hecho con prisas...

Estimaciones



Chaos Report
The Standish Group
1994-2004

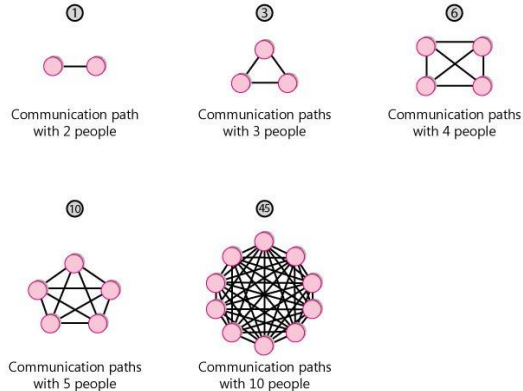
Size (KLOC)	Early	On Time	Late	Failed
1	11%	81%	6%	2%
10	6%	75%	12%	7%
100	1%	61%	18%	20%
1000	<1%	28%	24%	48%
10000	0%	14%	21%	65%

Capers Jones
Estimating Software Costs
1998

Estimaciones

Deseconomías de escala

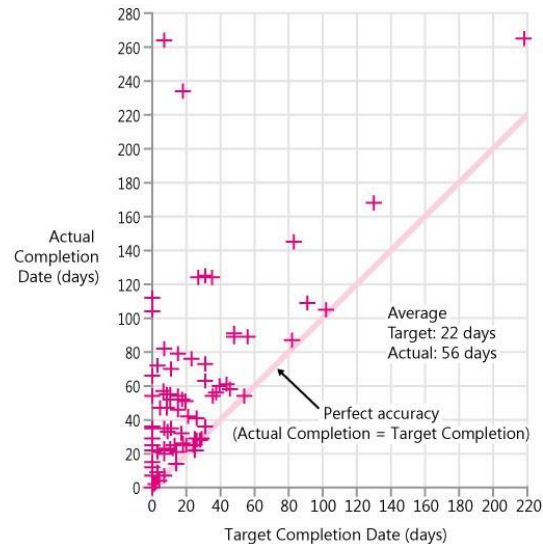
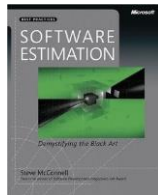
Los proyectos más grandes involucran a grupos más grandes de personas, por lo que requieren más comunicación.



Conforme aumenta el tamaño del proyecto, el número de canales de comunicación aumenta al cuadrado.

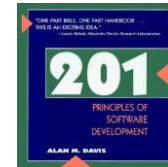
Estimaciones

Aunque sea preferible error por arriba,
las subestimaciones son más frecuentes (y peligrosas)



Estimaciones

- Las subestimaciones no siempre son malas...

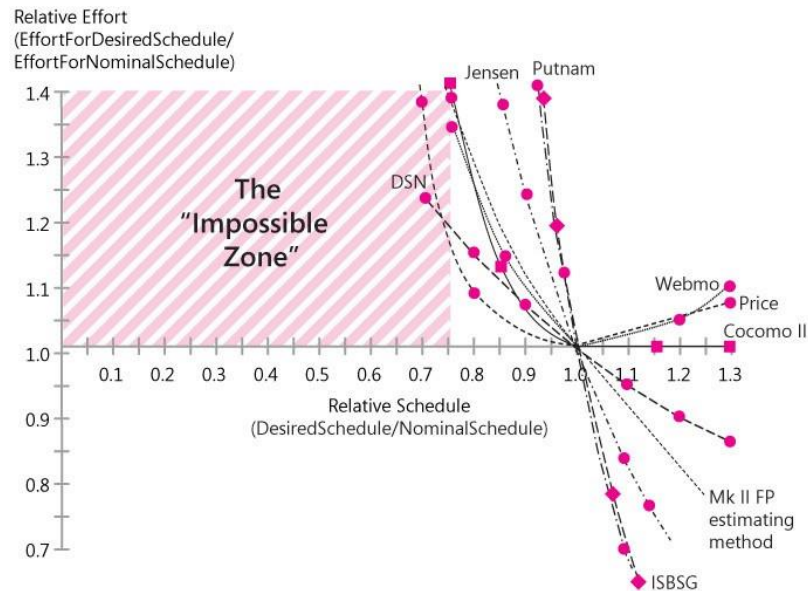


- El equipo de un proyecto que va ligeramente por detrás del calendario previsto tiende a trabajar más para ponerse al día: **productividad** ↑
- Si la subestimación es excesiva (poco realista), no obstante, la motivación y la productividad se desploman.

T. Abdel-Hamid & S. Madnick: "Impact of Schedule Estimation on Software Project Behavior", IEEE Software 3(4):70-75, July 1986

Estimaciones

Hay que evitar caer en la **zona imposible** ...

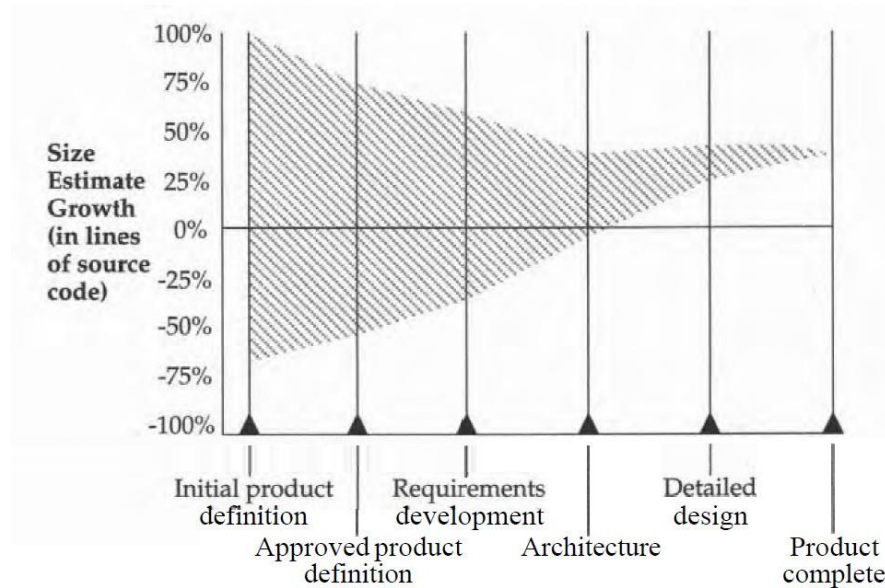


Source: Adapted and extended from Software Sizing and Estimating: Mk II (Symons 1991), Software Cost Estimation with Cocomo II (Boehm et al 2000), "Estimating Web Development Costs: There Are Differences" (Reifer 2002), and Practical Project Estimation, 2nd Edition (ISBSG 2005).

Cono de incertidumbre

- No tiene sentido dar estimaciones exactas cuando no se tiene información al respecto.
- Ningún gerente, arquitecto o diseñador, no importa cuánto sepa ni su experiencia, puede estimar con precisión cuántas características pueden ser hechas por cuántos programadores en cuánto tiempo.
- Cuanta más temprana es la etapa del proyecto, más difícil es estimar → Cono de incertidumbre.

Cono de incertidumbre



Cono de incertidumbre

NASA Software Engineering Laboratory

Estimación inicial al final del análisis de requisitos seguida de cinco revisiones a lo largo del proyecto.

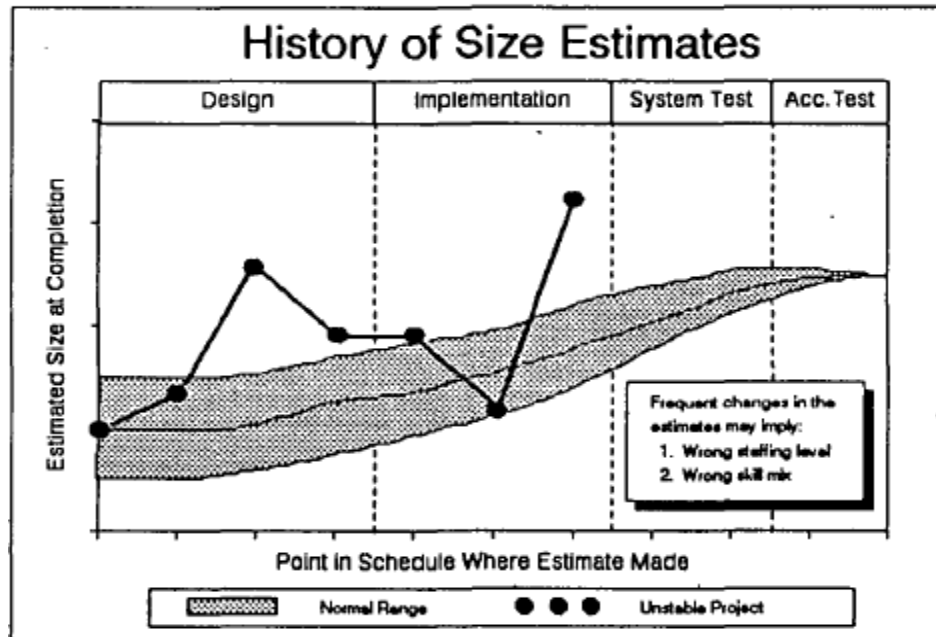
TABLE 7-1
EFFORT ESTIMATE REFINEMENT
THROUGHOUT A PROJECT

Estimation Point	Upper Limit	Lower Limit
End of requirements definition and specification	x 2.0	x 0.50
End of requirements analysis	x 1.75	x 0.57
End of preliminary design	x 1.4	x 0.71
End of detailed design	x 1.25	x 0.80
End of implementation	x 1.10	x 0.91
End of system testing	x 1.05	x 0.95

SEL's Manager's Handbook: Las estimaciones base típicamente crecen un **40%** a lo largo del proyecto.

Cono de incertidumbre

NASA Software Engineering Laboratory Desviaciones en las estimaciones



Cono de incertidumbre

NASA Software Engineering Laboratory Desviaciones en las estimaciones

End of Phase	Estimate Uncertainty	Lower Bound	Upper Bound
Requirements Definition	1.00	0.50	2.00
Requirements Analysis	0.75	0.57	1.75
Preliminary Design	0.40	0.71	1.40
Detailed Design	0.25	0.80	1.25
Implementation	0.10	0.91	1.10
System Testing	0.05	0.95	1.05

where:

The uncertainty may be applied to effort or size estimates made at the end of each named life-cycle phase as follows:

$$\text{Estimate}_{\text{upper bound}} = \text{Estimate} * (1.0 + \text{uncertainty})$$

$$\text{Estimate}_{\text{lower bound}} = \text{Estimate} / (1.0 + \text{uncertainty})$$

Cono de incertidumbre

Fase	Error por abajo	Error por arriba	Rango de las estimaciones
Concepto inicial	0.25x (-75%)	4.0x (+300%)	16x
Definición del producto aprobada	0.50x (-50%)	2.0x (+100%)	4x
Análisis de requisitos completado	0.67x (-33%)	1.5x (+50%)	2.25x
Diseño UI completado	0.80x (-20%)	1.25x (+25%)	1.6x
Diseño detallado completo	0.90x (-10%)	1.10x (+10%)	1.2x

Barry Boehm et al.:
Software Cost Estimation with Cocomo II, 2000

Cono de incertidumbre

Otras fuentes de error en las estimaciones

- Proyecto mal gestionado (fuera de control).
- Requisitos inestables (hay que controlarlos): +40%.
NASA Software Engineering Laboratory: "Manager's Handbook for Software Development," SEL-84-101, revision 1, 1990.
- Actividades omitidas en las estimaciones.
- Optimismo en las estimaciones, p.ej. 20%-30%
Michiel van Genuchten: "Why Is Software Late? An Empirical Study of Reasons for Delay in Software Development" IEEE Transactions on Software Engineering SE-17, no. 6, pp. 582-590, June 1991

Realización de estimaciones

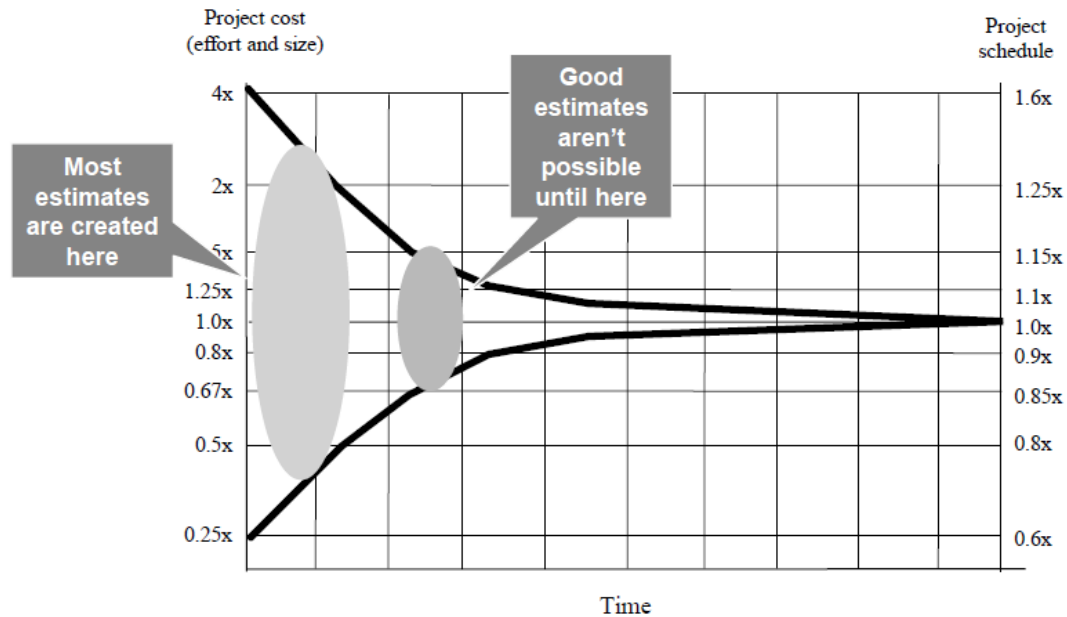
Estimación inicial

- Se realiza después de delimitar el ámbito del proyecto y establecer los requisitos.
- Se pueden realizar estimaciones del esfuerzo, coste y duración del proyecto.

Siempre se tarda más de lo que se espera, incluso cuando se tiene en cuenta esta ley". -- Douglas Hofstadter: GEB

Realización de estimaciones

Problemas de las estimaciones iniciales



Realización de estimaciones

- En muchos casos las estimaciones iniciales se hacen antes de los requisitos.
- **PROBLEMA:** No se comprende el problema a resolver.
- Se pueden realizar estimaciones del esfuerzo, coste y duración del proyecto.
- ¿"Es posible hacer este proyecto para el 1 de Junio"?
- ¿Cómo planificar bien si el dinero y las fechas de entrega ya están predefinidas?

Realización de estimaciones

Puntos clave

- Se pueden realizar estimaciones precisas, aunque las estimaciones precisas requieren tiempo.
- Las estimaciones precisas requieren el uso de técnicas cuantitativas, preferiblemente apoyadas por alguna herramienta software.
- Las estimaciones más precisas se basan en datos de proyectos ya completados por la organización encargada del proyecto actual.
- Las estimaciones requieren revisiones conforme el proyecto avanza.

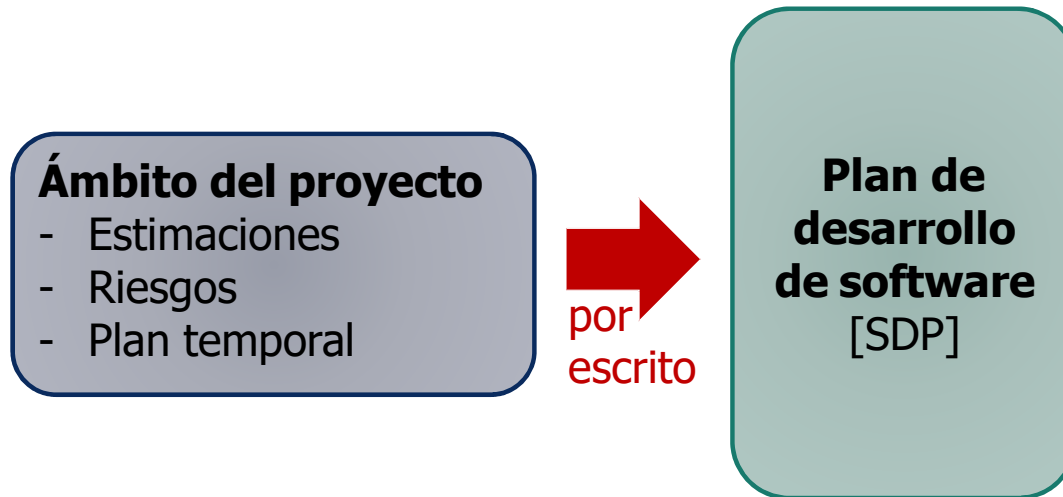
Realización de estimaciones

Procedimiento

- Se acuerda un procedimiento escrito entre los distintos stakeholders del proyecto.
- Se realiza una negociación racional de las entradas de la estimación (requisitos y recursos) evitando discutir las salidas (presupuesto y planificación temporal).

Realización de estimaciones

Procedimiento



Realización de estimaciones

Actividades que se han de incluir

Las estimaciones deberían incluir tiempo para todas las actividades normales del proyecto, incluyendo las que pueden no resultar obvias.

- Tareas de planificación.
- Diseño arquitectónico y diseño detallado.
- Implementación y pruebas.
- Creación de la documentación de usuario.
- Creación de programas de instalación.
- Creación de programas de migración del sistema antiguo al nuevo (p.ej. volcado de datos).

Realización de estimaciones

Actividades que se han de incluir

- Interacción con clientes y usuarios finales.
- Demostraciones de prototipos.
- Revisiones (planes, estimaciones, diseños, código, casos de prueba...).
- Corrección de problemas detectados en las revisiones.
- Mantenimiento del sistema de control de revisiones.
- Mantenimiento del entorno de desarrollo (p.ej. build scripts, copias de seguridad...).

Realización de estimaciones

Actividades que se han de incluir

- Consultas, QA, documentación...
- Evaluación del impacto de los cambios propuestos.
- Soporte a proyectos antiguos.
- Formación del personal que dará soporte al sistema.
- Formación técnica (del personal del equipo).
- Vacaciones y días libres.
- Bajas por enfermedad.
- ...

Realización de estimaciones

DETALLES A TENER EN CUENTA

- Añadir recursos adicionales al principio de proyecto minimiza el riesgo de que las estimaciones no se puedan cumplir.
- No se debe asumir que se trabajarán horas extra.

"Los proyectos deben ser como maratones. Tienes que establecer un ritmo saludable que pueda ganar la carrera y poder esprintar para llegar a la línea de meta."

- Ed Catmull, CTO, Pixar Animation Studios

Realización de estimaciones

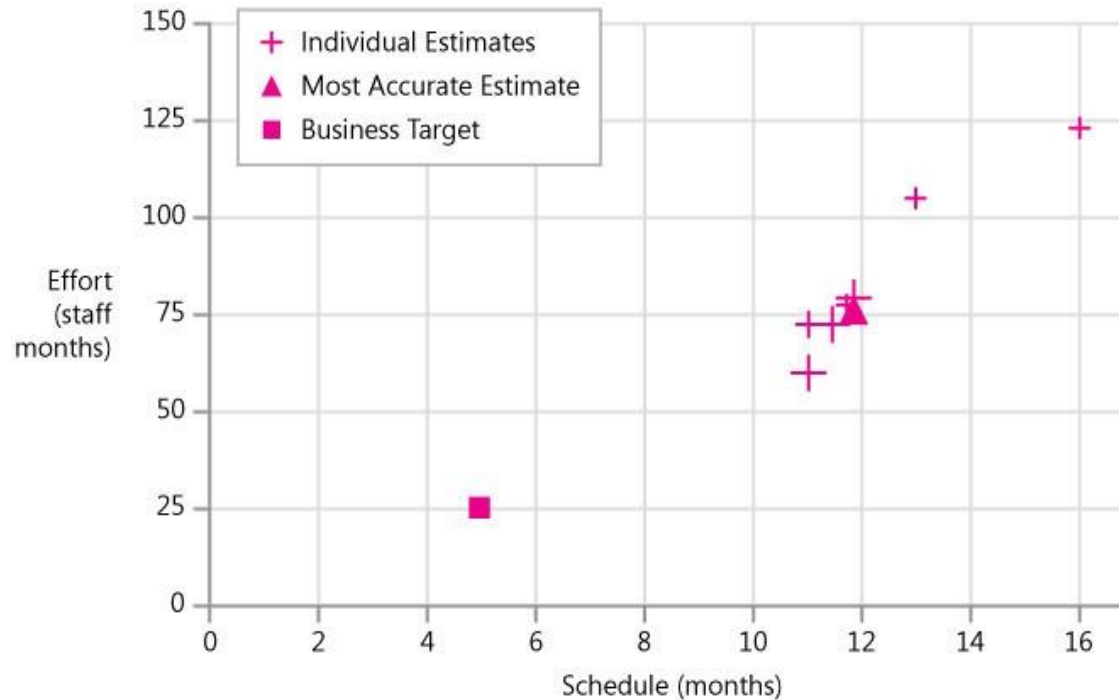
- Para la realización de estimaciones basadas en datos de proyectos anteriores:
 - Usar datos reales de ese proyecto.
 - **NO USAR** los datos de las estimaciones que se hicieron durante su planificación.

Realización de estimaciones

Para obtener buenas estimaciones:

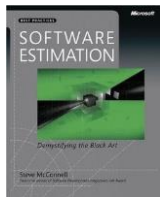
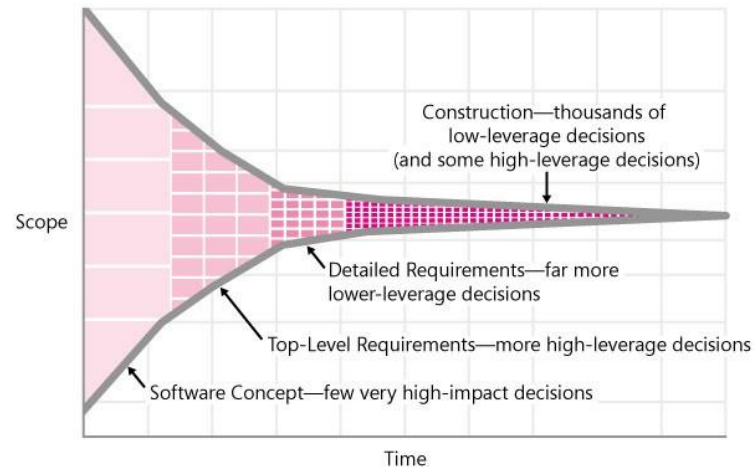
- Descomponemos el proyecto.
- Realizamos **dos o más** estimaciones utilizando técnicas diferentes:
 - Tamaño en líneas de código.
 - Puntos funcionales.
 - Casos de uso.
 - Tareas.
- Reconciliamos las diferentes estimaciones.

Realización de estimaciones



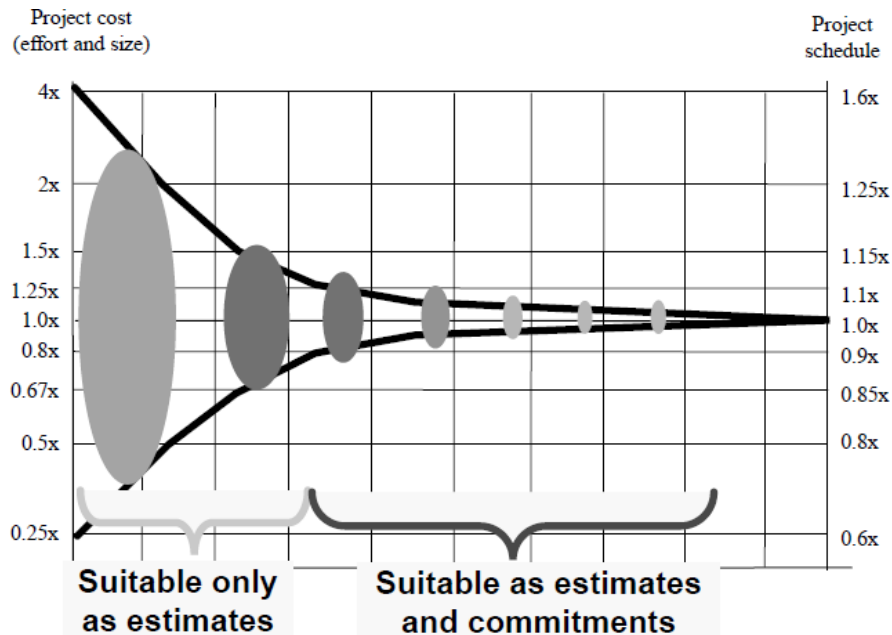
Realización de estimaciones

Conforme avanza el proyecto, las estimaciones se hacen con mayor grado de detalle/precisión:



Realización de estimaciones

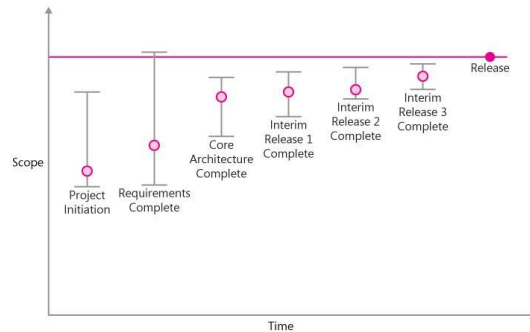
Revisiones a lo largo del proyecto



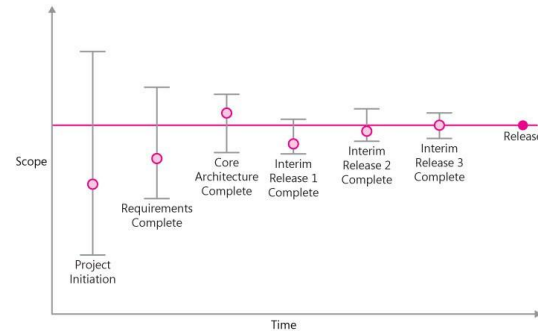
Realización de estimaciones

- Es deseable que las estimaciones que hagamos se cumplan, aunque eso implique rangos más amplios.

MAL

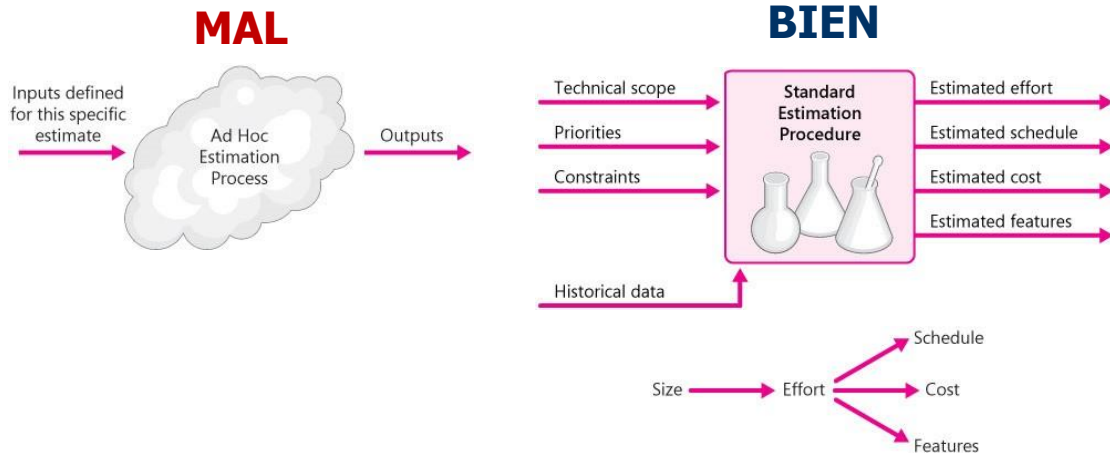


BIEN



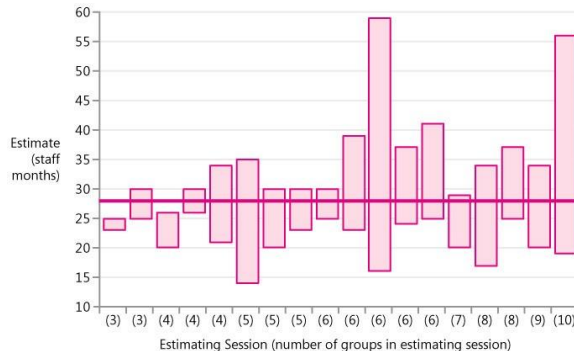
Realización de estimaciones

- Es deseable que tanto las entradas como las salidas de nuestras estimaciones estén bien especificadas y que se usen procedimientos estándar.

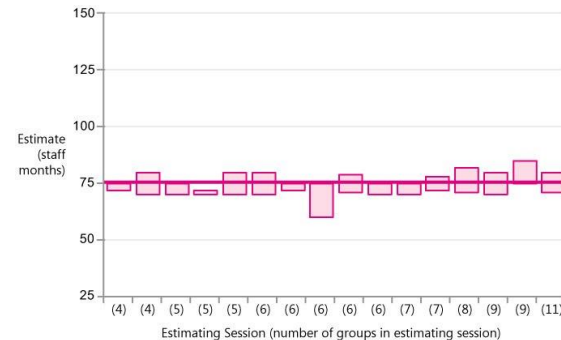


Técnicas de estimación

Variación en las estimaciones



Numerosos parámetros
subjetivos ajustables
Cocomo II (17 effort multipliers)

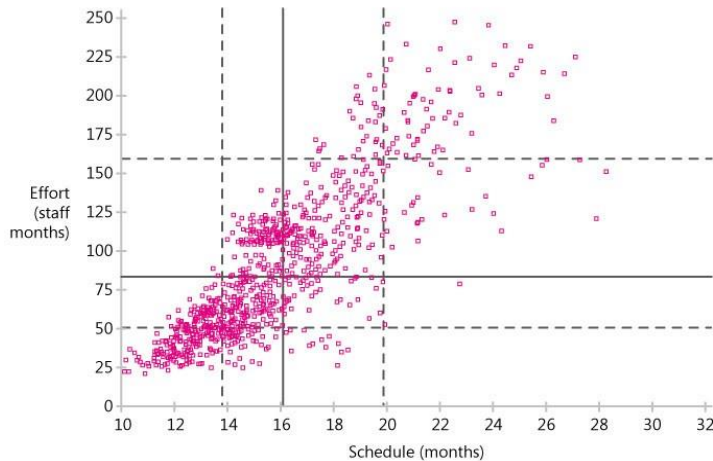


Subjetividad en un único
parámetro ajustable (los
demás son objetivos)

Técnicas de estimación

Calibración de los modelos de estimación

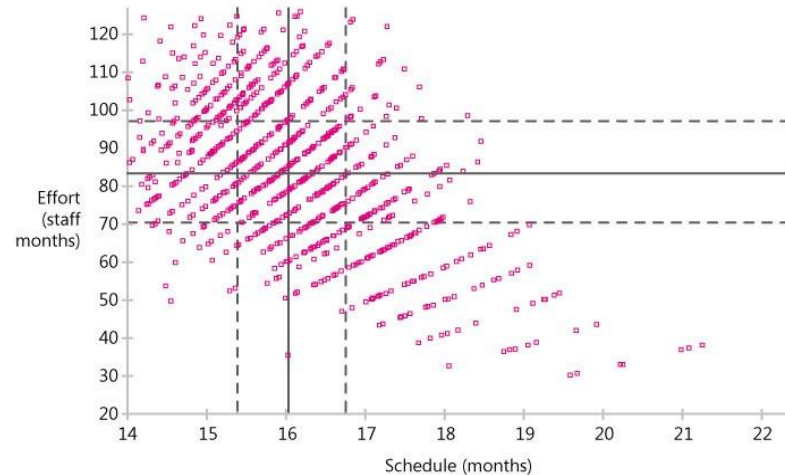
Schedule and Effort Simulation



Source: Estimated prepared using Construx Estimate, available at www.construx.com/estimate.

Datos medios de las
industria(variación 10x)

Schedule and Effort Simulation



Datos históricos de la propia
empresa (variación 4x)

Técnicas de estimación

Consejos de Steve McConnell

- Evite técnicas con demasiados parámetros que puedan ajustarse subjetivamente (dan una sensación de falsa precisión aunque degradan la precisión real).
- No realice estimaciones a bote pronto. Incluso una estimación realizada en unos minutos será más precisa.
- Ajuste el número de dígitos significativos de su estimación (su precisión) a la exactitud proporcionada por la técnica de estimación utilizada.

Técnicas de estimación

Presentación de las estimaciones



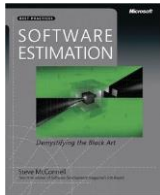
- Las estimaciones deben incluir el impacto de los riesgos asociados al proyecto.
- Esto se puede hacer de forma cuantificada: $\pm X t$, $\pm X €$

Técnicas de estimación

- Factor más significativo: El tamaño del proyecto es el **factor más significativo** para determinar el esfuerzo y tiempo necesarios.
 - El esfuerzo aumenta exponencialmente en función del tamaño.
- El tipo de software que se desarrolla en el **segundo factor más significativo** que influye en el esfuerzo necesario.
- El equipo del proyecto también ejerce una influencia significativa sobre el esfuerzo requerido para completar el proyecto...

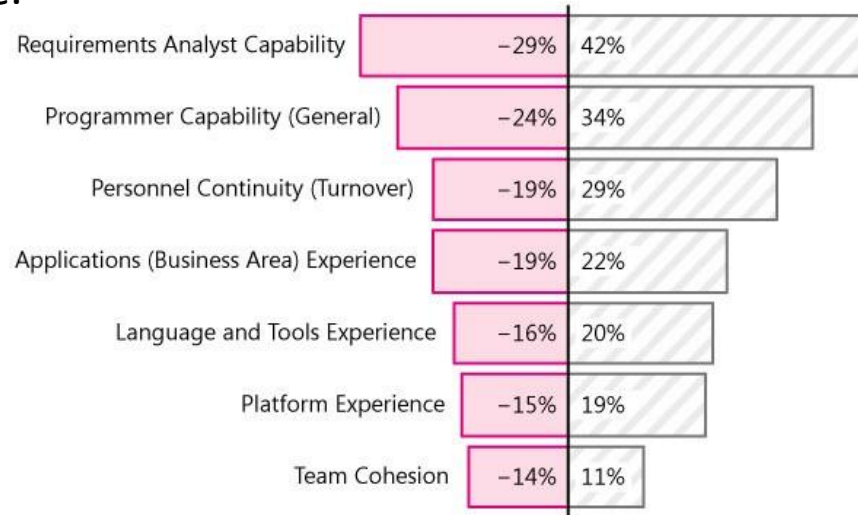
Técnicas de estimación

Efecto de los factores de personal en el esfuerzo del proyecto. Según la fuerza o debilidad de cada factor, los resultados del proyecto pueden variar en la cantidad que se indique.



Steve McConnell:
**Software Estimation:
Demystifying the Black
Art**

Microsoft Press, 2006.
ISBN 0735605351

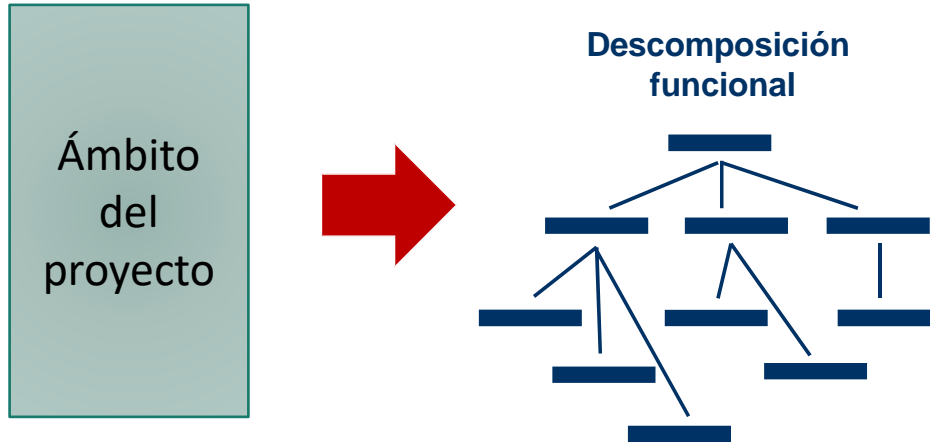


Técnicas de estimación

- Estimación por descomposición.
- Estimación de tamaño.
 - Líneas de código [KLOC].
 - Puntos funcionales [FP].
- Estimación por analogía:
Experiencia con otros proyectos similares.
- Modelos empíricos.
- Herramientas software de estimación.

Técnicas de estimación

Estimación por descomposición funcional



Técnicas de estimación

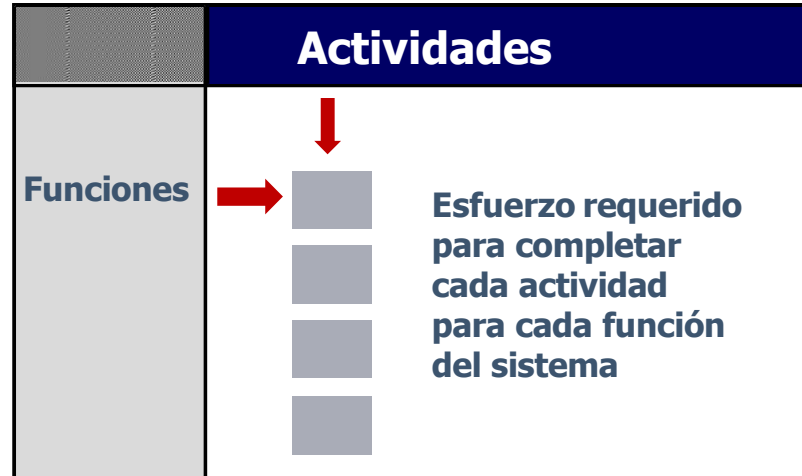
Estimación por descomposición funcional

Módulo	Esfuerzo estimado
Base de datos	1 p.m.
Modelo de clases	1 p.m.
Algoritmos X	2 p.m.
Interfaz de usuario	4 p.m.
Comunicaciones	1 p.m.
TOTAL	9 p.m.

- Costes laborales: 4000€/pm
- Estimación: $4000 * 9 = 36000€$

Técnicas de estimación

Estimación por descomposición de actividades



Técnicas de estimación

Estimación por descomposición de actividades

Módulo	Plan	Análisis	Diseño	Código	Test	TOTAL
BD		0.50	0.25	0.10	0.10	0.95
Modelo		0.50	0.25	0.10	0.15	1.00
Algoritmos		0.50	0.50	0.75	1.50	3.25
UI		1.00	1.00	1.00	1.00	4.00
Redes		0.25	0.25	0.25	0.25	1.00
TOTAL	0.25	2.75	2.25	2.20	3.00	10.45
%	2%	26%	22%	21%	29%	

- Costes laborales: 4000€/pm
- Estimación: 41800€ (10.45 pm)

Técnicas de estimación

Estimación del tamaño del proyecto

- Estimar el tamaño del proyecto:
 - Líneas de código (KLOC).
 - Puntos funcionales (FP).
 - Casos de uso.
- Estimar coste y esfuerzo a partir del tamaño estimado (usando datos históricos).
- **Principal dificultad:** Es difícil estimar a priori el tamaño de un proyecto.

Técnicas de estimación

Estimación del tamaño del proyecto: KLOC

Módulo	Tamaño estimado
Base de datos	3.3 KLOC
Modelo de clases	2.3 KLOC
Algoritmos X	5.3 KLOC
Interfaz de usuario	8.4 KLOC
Comunicaciones	2.1 KLOC
TOTAL	21.4 KLOC

- Productividad media en sistemas de este tipo:
2 KLOC / pm [person-month]
- Costes laborales: 4000€/pm (2€/LOC)
- Estimación: $2 * 21400 = 42800€$ y $21.4/2 \cong 11$ pm

Técnicas de estimación

Estimación del tamaño del proyecto: FP

	o	m	p	est.	peso	FP
Entradas	20	24	30	24	4	97
Salidas	12	15	22	16	5	78
Consultas	16	22	28	22	4	88
Ficheros	4	4	5	4	10	42
Interfaces externos	2	2	3	2	7	15
Total						320

- $FP_{\text{estimado}} = FP_{\text{total}} * (0.65 + 0.01 \Sigma ACT) = 1.17 * FP = 375$
- Productividad: 39 FP / pm [person-month]
- Costes laborales: 4000€/pm (~100€/FP)
- Estimación: 38500€ y 10 pm

Nº Factor	Descripción	Valor (0 a 5)
1	Comunicación de datos.	
2	Proceso distribuido.	
3	Objetivos de rendimiento.	
4	Configuración de explotación compartida.	
5	Tasa de transacciones.	
6	Entrada de datos en línea.	
7	Eficiencia con el usuario final.	
8	Actualizaciones en línea.	
9	Lógica de proceso interno compleja.	
10	Reusabilidad del código.	
11	Conversión e instalación contempladas.	
12	Facilidad de operación.	
13	Instalaciones múltiples.	
14	Facilidad de cambios.	

Cálculo del Ajuste de complejidad técnica (ACT) = Suma de los valores

Técnicas de estimación

Estimación del tamaño del proyecto: Casos de uso

- Cálculo de UCP (Use case points).
 - El resultado son horas.
- Se utiliza:
 - Technical Complexity Factor (TCF).
 - Environment Complexity Factor (ECF).
 - Unadjusted Use Case Points (UUCP).
 - Productivity Factor (PF).

$$\text{UCP} = \text{TCF} \cdot \text{ECF} \cdot \text{UUCP} \cdot \text{PF}$$

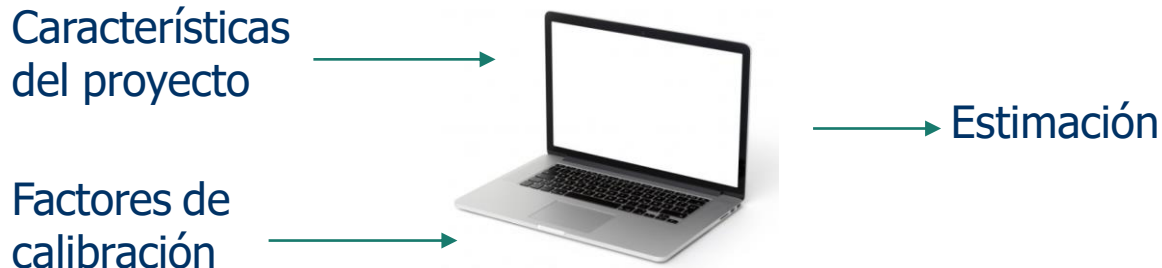
Técnicas de estimación

Estimación del tamaño del proyecto: Casos de uso

- $TCF = 0.6 + (.01 * \text{Total Complexity Factor})$.
- $ECF = 1.4 + (-0.03 * \text{Total Environmental Factor})$
- $UUCP = UUCW + UAW$
- Ejemplo de aplicación en:
<https://www.codeproject.com/Articles/9913/Project-Estimation-with-Use-Case-Points>

Técnicas de estimación

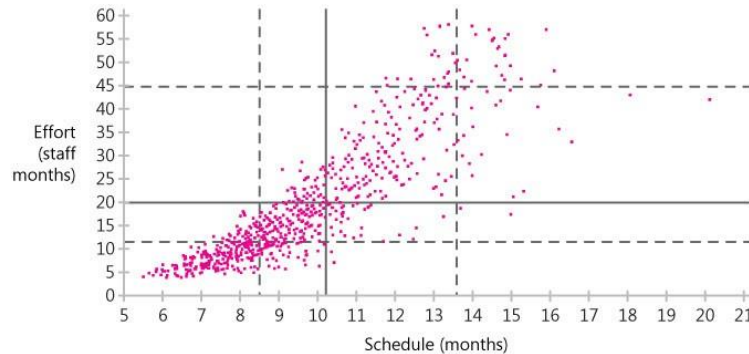
Estimación con herramientas software



Técnicas de estimación

Estimación con herramientas software

Las herramientas nos permiten realizar simulaciones estadísticas que nos permiten incorporar distintas fuentes de variabilidad en nuestras estimaciones:

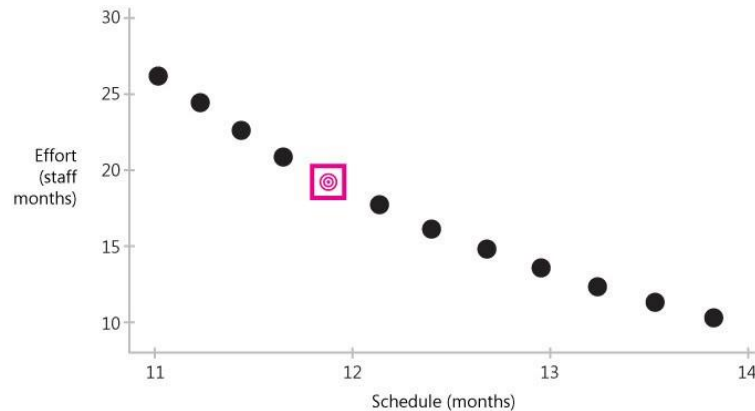


Construx Estimate

Técnicas de estimación

Estimación con herramientas software

También nos permiten plantear escenarios (p.ej. trade-off esfuerzo-tiempo).



Construx Estimate

Técnicas de estimación

Modelos empíricos

Cálculo del esfuerzo

$$\text{esfuerzo} = A + B \times \text{tamaño}^c$$

EJEMPLOS

$$E = 5.2 \times \text{KLOC}^{0.91}$$

Walston-Felix

$$E = 3.2 \times \text{KLOC}^{1.05}$$

Boehm

$$E = 5.288 \times \text{KLOC}^{1.047}$$

Doty (KLOC > 9)

$$E = 5.5 + 0.73 \times \text{KLOC}^{1.16}$$

Bailey-Basili

$$E = -0.91 + 0.355 \text{ FP}$$

Albretch & Gaffney

$$E = -37 + 0.96 \text{ FP}$$

Kemerer

Técnicas de estimación

COCOMO II [COnstructive COst MOdel]

Barry Boehm et al.: **Software Cost Estimation with Cocomo II**, 2000.

<http://softwarecost.org/tools/COCOMO/>

Jerarquía de modelos de estimación:

- Estimación del tamaño: líneas de código [LOC], puntos funcionales [FP] o puntos "de objetos" [OP].
- Ajuste por reutilización: $NOP = OP * (100 - \%reuse) / 100$
- Productividad: $PROD = NOP / \text{person-month}$
- Esfuerzo estimado: $EFF = NOP / PROD$

COCOMO II admite una amplia gama de factores de escala y procedimientos de ajuste.

Técnicas de estimación

COCOMO II [COConstructive COst MOdel]

Dependiendo del tipo de software, se aplican distintas métricas:

- Aplicaciones de usuarios finales.
- Generadores de aplicaciones.
- Aplicaciones con componentes.
- Sistemas integrados.
- Infraestructura.

Técnicas de estimación

COCOMO II [COConstructive COst MOdel]

Dependiendo de la fase del proyecto:

- Modelo de composición de aplicación.
- Modelo de diseño temprano.
- Modelo post-arquitectura.

Técnicas de estimación

Modelo de Putnam: “La ecuación del software”

Lawrence H. Putnam & Ware Myers: **Measures for Excellence**, 1992.

$$E = [LOC * B^{0.333} / P]^3 * (1/t^4)$$

B crece conforme aumenta el tamaño del proyecto:

B=0.16	5<=KLOC<=15
B=0.39	70<=KLOC

P depende del tipo de proyecto:

P = 2000	Sistemas empotrados
P = 10000	Software de sistemas y comunicaciones
P = 12000	Software científico
P = 28000	Aplicaciones de gestión

Técnicas de estimación

Modelo de Putnam: “La ecuación del software”

Lawrence H. Putnam & Ware Myers: **Measures for Excellence**, 1992.

Tiempo de desarrollo mínimo

(para proyectos de más de 6 meses de duración)

$$t_{min} = 8.14 * \left(\frac{LOC}{P} \right)^{0.43}$$

Esfuerzo necesario

(para proyectos de al menos 20 personas·mes)

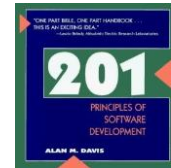
$$E = 180 * B * t^3$$

Reflexiones finales

- Para estimar el coste y el calendario → estimamos líneas de código
- ¿Es realmente la estimación de líneas de código más fiable que la del coste y el calendario?
- VENTAJA: Podemos usar datos históricos.

Reflexiones finales

Aun utilizando datos históricos cuidadosamente para calibrar un modelo de estimación que nos permita estimar los costes de un proyecto, hay tres razones por las que la estimación puede no ser precisa:



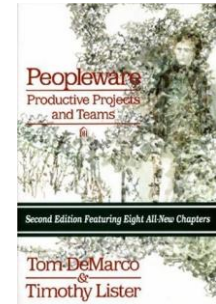
- Tú, como **gestor** de proyecto (capaz de destruir la motivación del equipo en sólo unos segundos).
- Las **suposiciones** utilizadas al realizar la estimación pueden no ser las adecuadas.
- La **probabilidad** (la estimación es sólo un pico en una distribución de probabilidad).

Reflexiones finales

¿Quién debe realizar las estimaciones?

Productividad vs. Responsable de la estimación

Estimador	Productividad	Proyectos
Programador	8.0	19
Supervisor	6.6	23
Programador & supervisor	7.8	16
Analista de sistemas	9.5	21
Sin estimación	12.0	24



¿Mayor productividad en proyectos sin calendario?

Mejor que sea el propio desarrollador el responsable de las estimaciones o un analista de sistemas externo...

Referencias

- Fernando Berzal:
Planificación y gestión de proyectos informáticos.
<http://elvex.ugr.es/decsai/project-management/>



DECSAI

Departamento de Ciencias de la Computación e I.A.
Universidad de Granada



Estimación

Fernando Berzal, berzal@acm.org

Bibliografía

- Roger S. Pressman:
Software Engineering: A Practitioner's Approach
McGraw-Hill, 8th edition, 2014. ISBN 0078022126
- Shari Lawrence Pfleeger & Hoanne M. Atlee: **Software Engineering: Theory and Practice** Prentice Hall, 4th edition, 2009. ISBN 0136061699
- Ian Sommerville:
Software Engineering
Pearson, 10th edition, 2015. ISBN 0133943038

Bibliografía

Lecturas recomendadas

- Dwayne Phillips:
**The Software Project Manager's Handbook:
Principles That Work at Work**
Wiley / IEEE Computer Society, 2nd edition, 2004
ISBN 0471674206

- Donald J. Reifer (editor):
Software Management
Wiley / IEEE Computer Society, 7th edition, 2006
ISBN 0471775622

- Richard H. Thayer (editor):
Software Engineering Project Management
Wiley / IEEE Computer Society, 2nd edition, 2000
ISBN 0818680008

Bibliografía complementaria

Estimación

- Steve McConnell:
Software Estimation: Demystifying the Black Art
Microsoft Press, 2006. ISBN 0735605351
- Steve Tockey: **Return on Software:
Maximizing the return on your software investment**
Addison-Wesley Professional, 2004. ISBN 0321228758
- Richard D. Stutzke:
**Estimating Software-Intensive Systems:
Projects, Products, and Processes**
Addison-Wesley Professional, 2005. ISBN 0201703122
- Mike Cohn:
Agile Estimating and Planning
Prentice Hall, 2005. ISBN 0131479415
- **Cost and Schedule Estimation Study Report**
NASA Software Engineering Laboratory, SEL-93-002, 1993.

Bibliografía complementaria

Clásicos

- Frederick P. Brooks, Jr.:
**The Mythical Man-Month:
Essays on Software Engineering**
Addison-Wesley, 1995. ISBN 0201835959
- Alan M. Davis:
201 Principles of Software Development
McGraw-Hill, 1995. ISBN 0070158401
- Barry W. Boehm:
Software Engineering Economics
Prentice-Hall PTR, 1991. ISBN 0138221227
- **Manager's Handbook for Software Development**
NASA Software Engineering Laboratory, SEL-84-101, rev.1, 1990.
- **Software Engineering Laboratory (SEL)
Relationships, Models, and Management Rules**
NASA Software Engineering Laboratory, SEL-91-001, 1991.

Bibliografía

Bibliografía en castellano



- Roger S. Pressman:
Ingeniería de Software: Un enfoque práctico
McGraw-Hill, 7ª edición, 2010. ISBN 6071503140
- Ian Sommerville:
Ingeniería de Software
Pearson, 9ª edición, 2012. ISBN 6073206038



Ejercicios

Usando alguno de los modelos de estimación vistos en este tema realice las siguientes tareas:

- Escoja 3 tipos diferentes de software (por ejemplo: aplicación Web, sistema empujado y middleware de alto rendimiento) y realice una comparativa entre las estimaciones realizadas para los 3 casos suponiendo que el tamaño del equipo es el mismo y la experiencia es similar en los 3 campos.
- Analice los resultados de las estimaciones e indique qué factores son los que podrían haber hecho que los resultados hayan sido diferentes (o similares).

Ejercicios

Cocomo II: <http://csse.usc.edu/tools>

Casos de uso: <http://tynerblain.com/downloads/UseCasePoints.xls>