



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Estimación

Fernando Berzal, berzal@acm.org

Estimación



- 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



Estimación

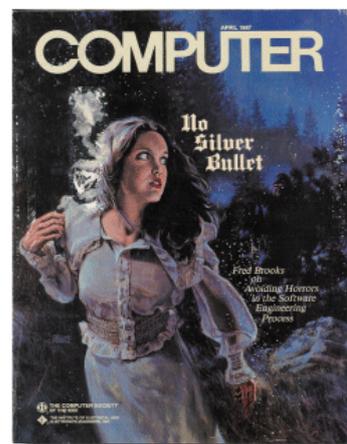


El ámbito del proyecto



“The hardest single part of building a software system is deciding precisely what to build.”

— **Frederick P. Brooks Jr.:** “No Silver Bullet: Essence and Accidents of Software Engineering”, IEEE Computer, April 1987



El ámbito del proyecto



El ámbito del proyecto debe establecerse de forma no ambigua y comprensible para los distintos "stakeholders" del proyecto (gestores, técnicos y clientes).

- **Contexto:** ¿Cómo encaja el sistema en su entorno?
- **Objetivos:** ¿Qué resultados se esperan del proyecto? Casi tan importante, ¿qué queda fuera 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

- 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?

Barry Boehm: "Anchoring the Software Process,"
IEEE Software, vol. 13, no. 4, July 1996, pp. 73–82.



El ámbito del proyecto



Descripción del ámbito del proyecto

Una vez acordado por todos los "stakeholders", se elabora un documento que recoja lo acordado en uno de los dos formatos siguientes:

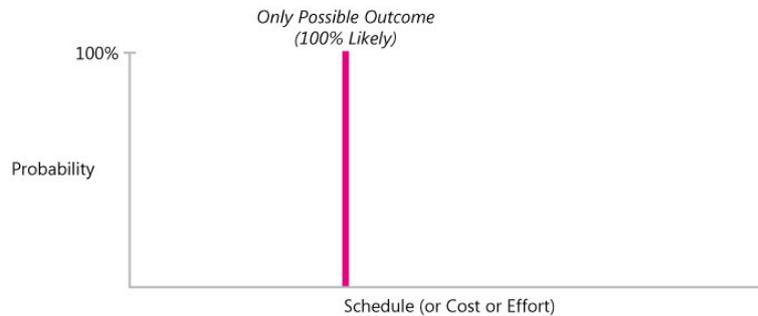
- Una narración descriptiva (en pocas páginas)
- Un conjunto de casos de uso "esenciales".



¿Qué es una estimación?



Estimación puntual



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

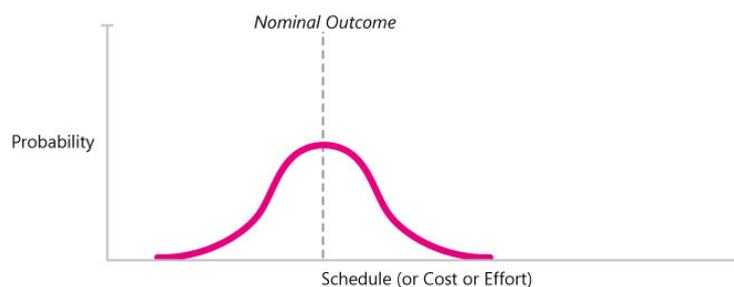
Muy poco realista...



¿Qué es una estimación?



Estimación normal / gaussiana



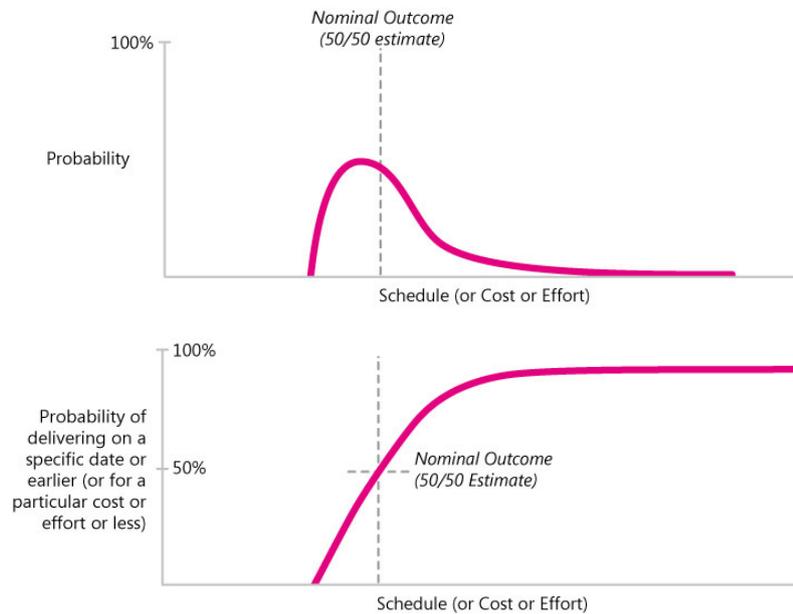
Basada en suposiciones incorrectas: existen límites en cuanto a la eficiencia con la que un equipo puede completar cualquier tarea que se le encomiende.



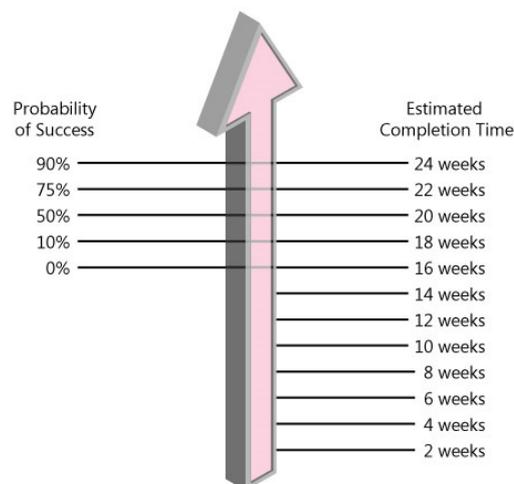
¿Qué es una estimación?



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:

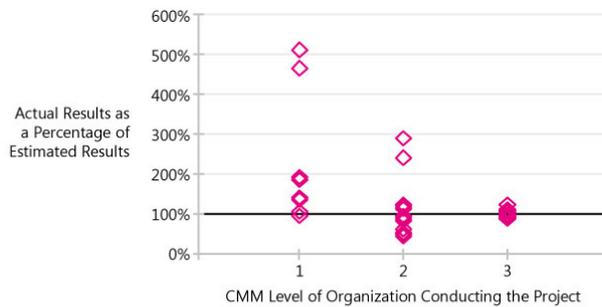
Software Engineering Metrics and Models, 1986



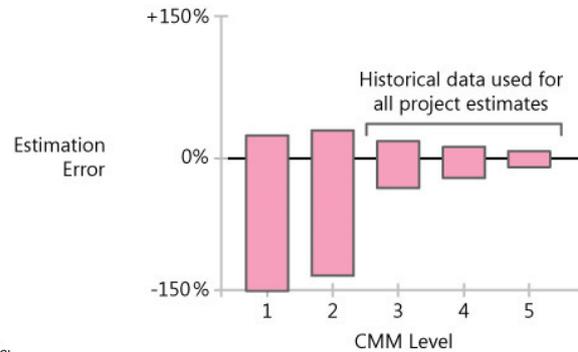
¿Qué es una estimación?



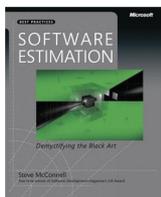
U.S. Air Force



Boeing



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



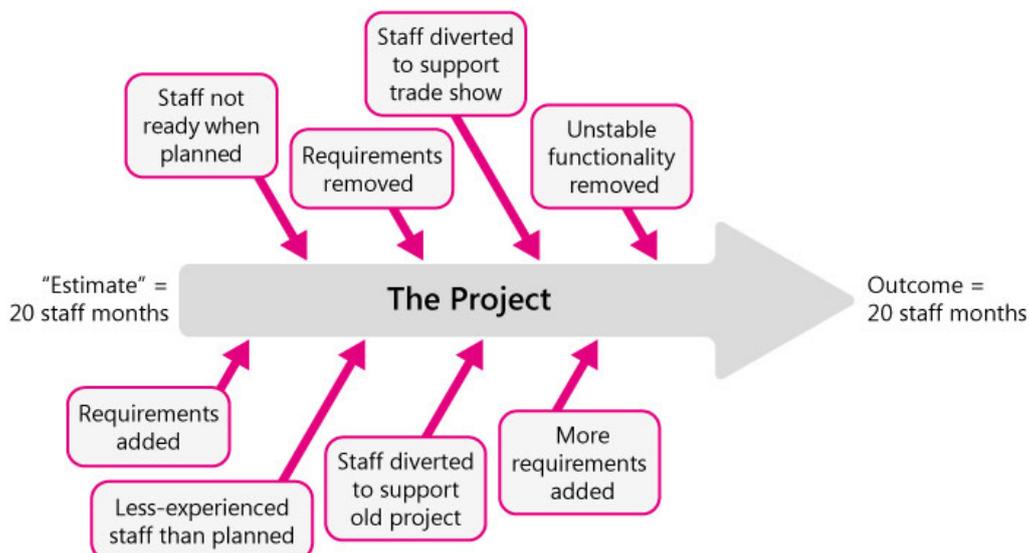
Steve McConnell:
Software Estimation: Demystifying the Black Art
 Microsoft Press, 2006. ISBN 0735605351



¿Qué es una estimación?



Factores que influyen en la precisión de una estimación



Algunos los podemos controlar, otros no...



¿Qué es una estimación?



[The common definition of estimate is]

"the most optimistic prediction that has a non-zero probability of coming true." ...

Accepting this definition leads irrevocably toward a method called what's-the-earliest-date-by-which-you-can't-prove-you-won't-be-finished estimating

—Tom DeMarco



¿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 del proyecto tomar buenas decisiones sobre cómo controlar el proyecto para lograr sus objetivos.

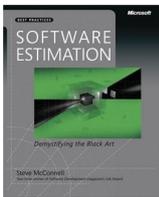
-- Steve McConnell: Software Estimation, 2006



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



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

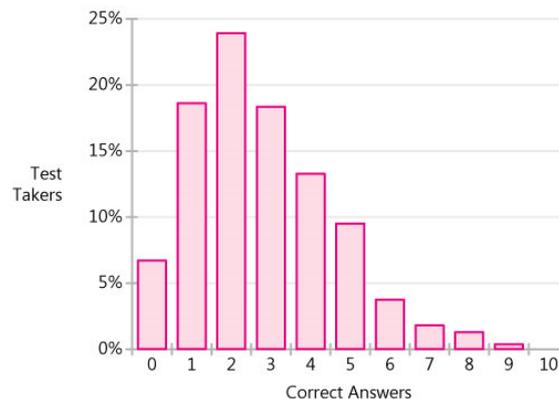


Test de autoevaluación



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 :-)



Test de autoevaluación



Consejos de Steve McConnell

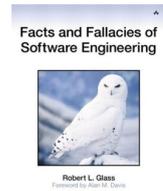
- No proporcione un porcentaje de confianza para sus estimaciones (p.ej. "90% de confianza") si no dispone de un mecanismo cuantitativo para derivarlo.
- Evite utilizar intervalos artificialmente estrechos: los intervalos deben representar su confianza en las estimaciones.
- Si siente presión para estrechar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



Estimaciones



Una de las dos causas más frecuentes del fracaso de un proyecto se debe a errores de estimación.



-- **Robert L. Glass:**

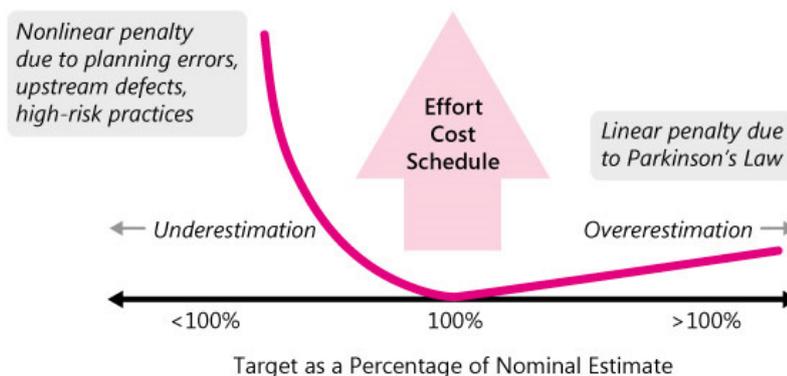
"Facts and Fallacies of Software Engineering", 2003.

- Barbara Kitchenham, Shari Lawrence Pfleeger, Beth McCall & Suzanne Eagan: "An Empirical Study of Maintenance and Development Estimation Accuracy" *Journal of Systems and Software*, September 2002.
- Andy Cole: "Runaway Projects—Causes and Effects" *Software World (UK)* 26:3, 1995.
- Michiel Van Genuchten: "Why Is Software Late?" *IEEE Transactions on Software Engineering*, June 1991.
- S.N. Mohanty: "Software Cost Estimation: Present and Future" *Software Practice and Experience* 11:103-21, 1981.
- Capers Jones: "Assessment and Control of Software Risks", Yourdon Press, 1994

NOTA: La otra es la inestabilidad de los requisitos.



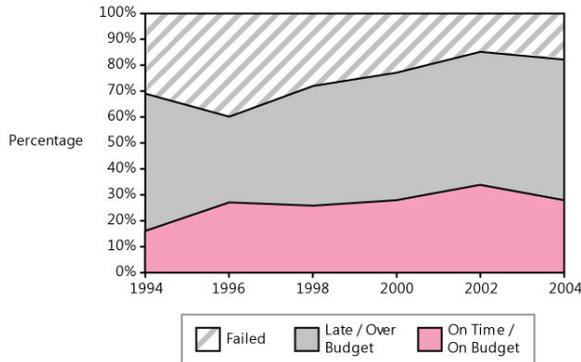
Estimaciones



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



Estimaciones



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%

Chaos Report
The Standish Group
1994-2004

Capers Jones
Estimating Software Costs
1998

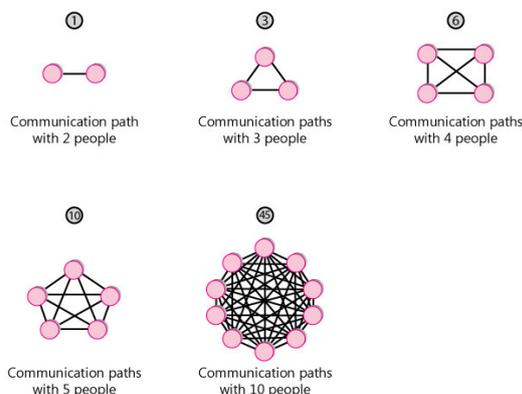


Estimaciones



Deseconomías de escala

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



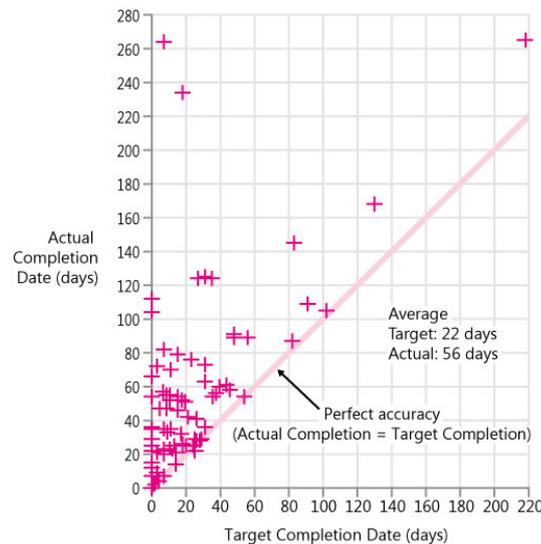
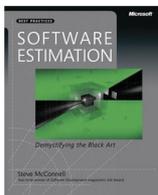
Conforme aumenta el tamaño del proyecto, el número de canales de comunicación aumenta al cuadrado.



Estimaciones



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

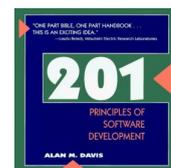


Estimaciones



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** ↑
- El equipo de un proyecto que va por delante del calendario previsto tiende a relajarse (días libres, más tiempo leyendo el correo...): **productividad** ↓



Si la subestimación es excesiva (poco realista), no obstante, la motivación y la productividad se desploman.

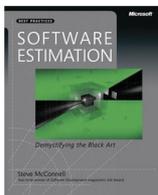
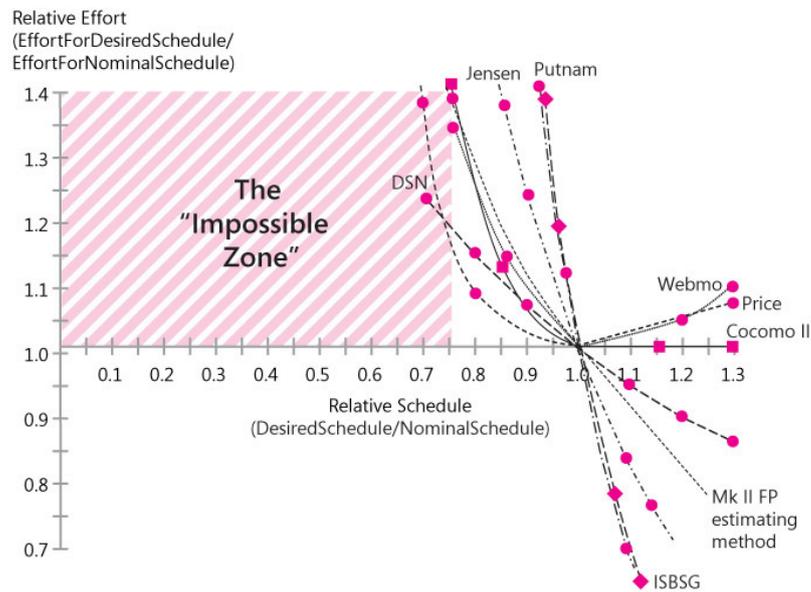
T. Abdel-Hamid & S. Madnick: "Impact of Schedule Estimation on Software Project Behavior", IEEE Software 3(4):70-75, July 1986



Estimaciones



... pero son siempre peligrosas: **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



There's no point in being exact about something if you don't even know what you're talking about.

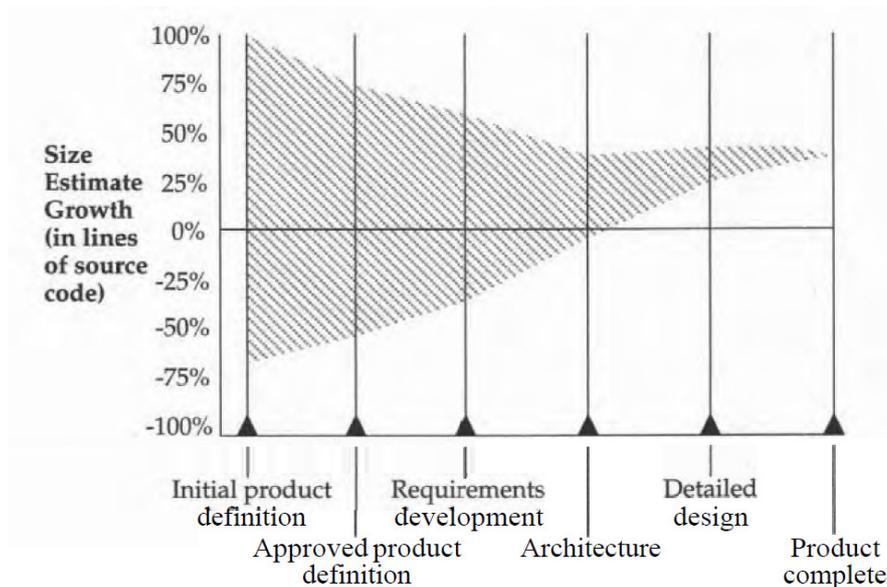
— John von Neumann

It should be clear that no manager, no architect, no designer, no matter how lofty, can precisely estimate just how many features can be done by how many programmers in how much time.

-- Ron Jeffries: "Making the Date", 2005



Cono de incertidumbre



Steve McConnell: Software Project Survival Guide, 1997



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

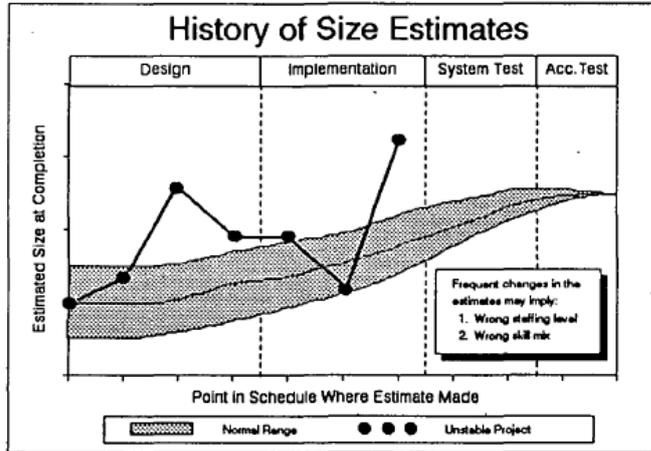
SEL's Manager's Handbook: Las estimaciones base típicamente crecen un **40%** a lo largo del proyecto.



Cono de incertidumbre



NASA Software Engineering Laboratory Desviaciones en las estimaciones



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:

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

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

If size or effort estimates are changing frequently
then the estimates are being adjusted to match the wrong staffing level or the development team has the wrong skill mix



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

En cuanto se ha delimitado el ámbito del proyecto y se han establecido los requisitos del mismo, el equipo del proyecto puede realizar estimaciones con sentido del esfuerzo, coste y duración del proyecto.

"Hofstadter's Law: It always takes longer than you expect, even when you take Hofstadter's Law into account". — Douglas Hofstadter: GEB

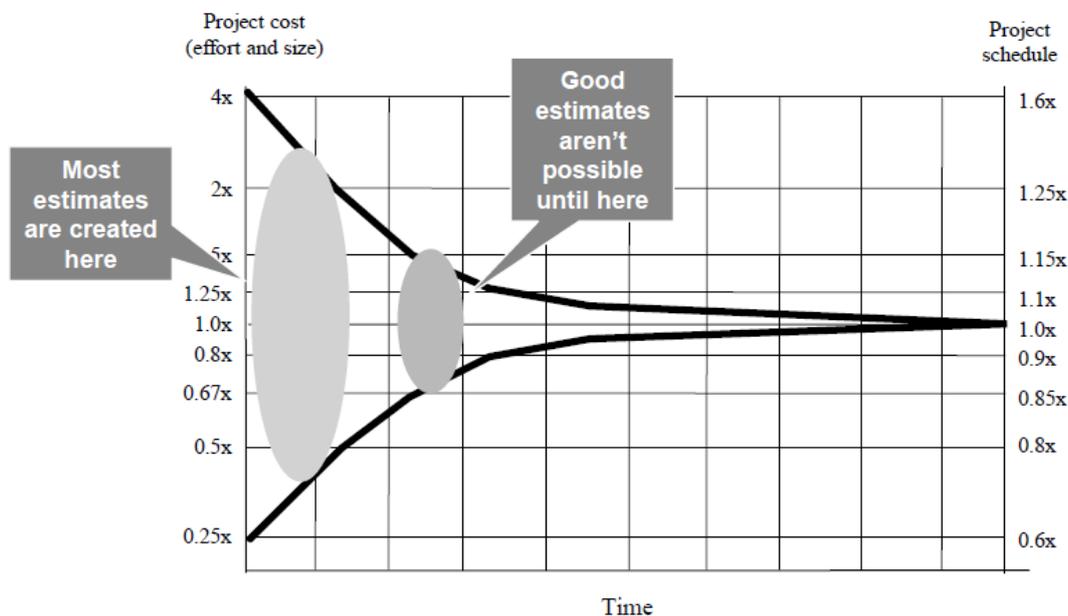


Realización de estimaciones



Estimación inicial

PROBLEMA



Realización de estimaciones



Estimación inicial

PROBLEMA

La estimación inicial se realiza antes de que se definan los requisitos y, por tanto, antes de comprender el problema que se pretende resolver.

"My problem is that delivery dates and budgets are established before we begin a project. The only question my management asks is 'Can we get this project out the door by June 1?' What's the point in doing detailed project estimates when deadlines and budgets are predefined?"

-- **Roger S. Pressman:** *"Software Project Management: Q and A."*
American Programmer (now Cutter IT Journal), December 1992

Facts and Fallacies of Software Engineering



Robert L. Glass
Foreword by Steve McConnell



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

Las estimaciones se deberían realizar siguiendo un procedimiento escrito, acordado por los distintos "stakeholders" del proyecto:

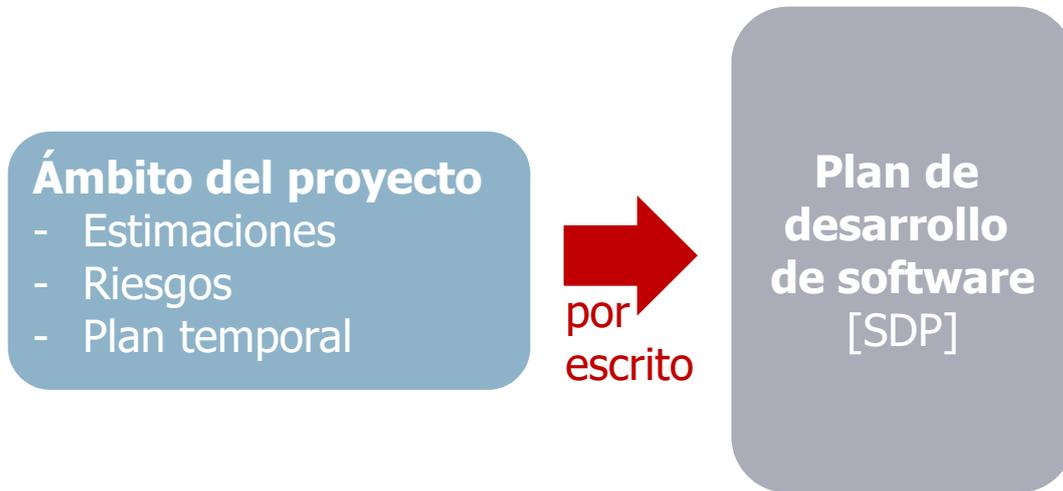
Una vez acordado el procedimiento de estimación, se puede realizar una negociación racional sobre las entradas de la estimación (características/requisitos y recursos) en vez de una discusión irracional sobre sus salidas (presupuesto y planificación temporal).



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



OBSERVACIONES

- Para minimizar el riesgo de que las estimaciones no se puedan cumplir, es mejor añadir recursos adicionales al inicio del proyecto.
- El plan previsto no debe asumir que el equipo trabajará horas extra.

“Projects should be run like marathons. You have to set a healthy pace that can win the race and expect to sprint for the finish line.”

— **Ed Catmull**, CTO, Pixar Animation Studios



Realización de estimaciones



Al realizar estimaciones basadas en datos de proyectos anteriores, es fundamental utilizar los datos reales de ejecución de los proyectos anteriores (no las estimaciones realizadas durante la planificación de esos proyectos)

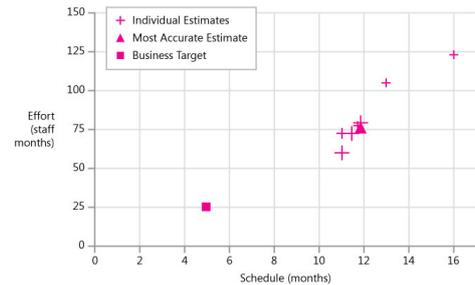


Realización de estimaciones



Para obtener buenas estimaciones:

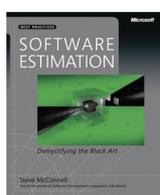
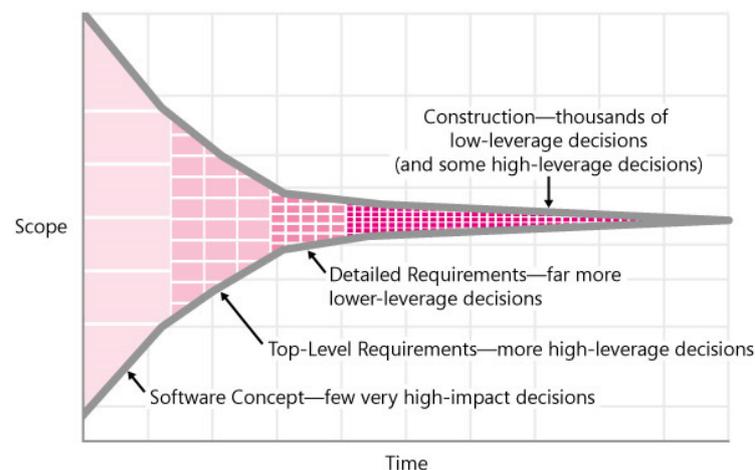
- Descomponemos el proyecto.
- Realizamos **dos o más** estimaciones utilizando técnicas diferentes (usando tamaño en líneas de código, puntos funcionales, casos de uso o tareas).
- Reconciliamos las diferentes estimaciones.



Realización de estimaciones



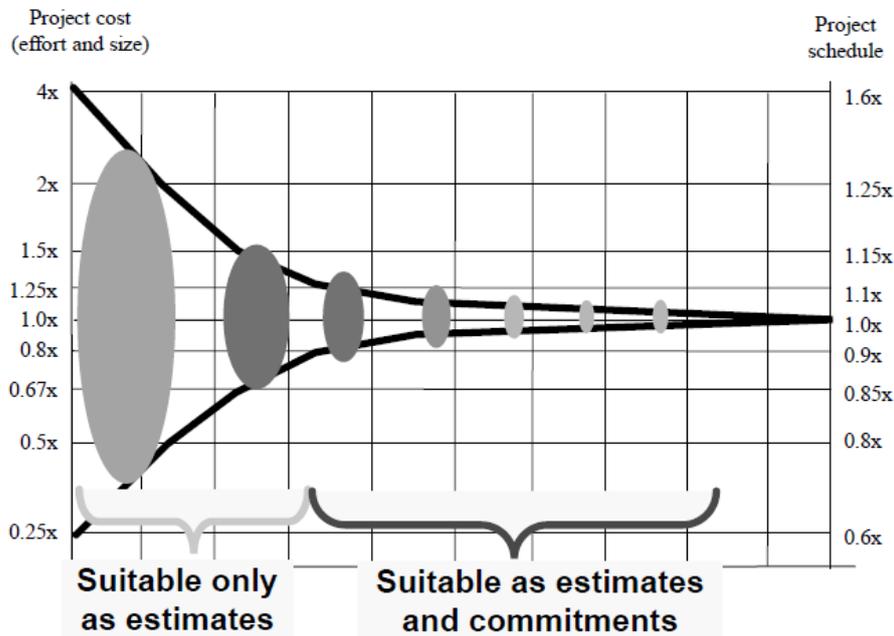
Conforme avanza el proyecto, las estimaciones se hacen con mayor grado de detalle/precisión:



Realización de estimaciones



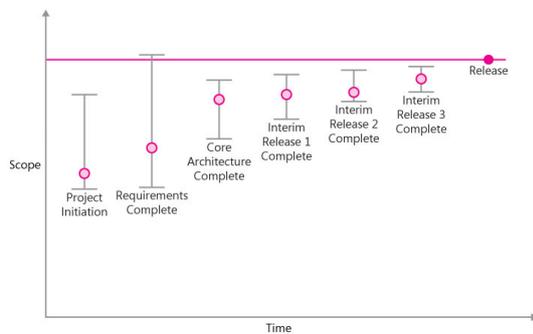
Revisiones a lo largo del proyecto



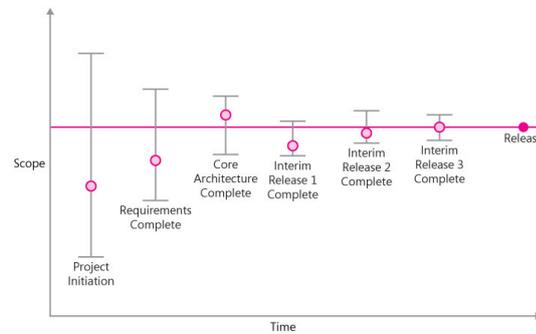
Realización de estimaciones



MAL



BIEN



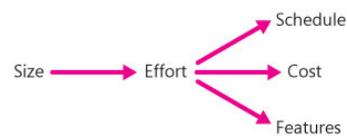
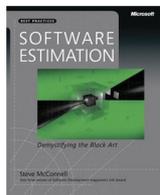
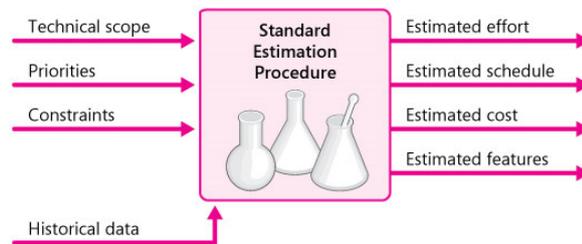
Técnicas de estimación



MAL



BIEN



Técnicas de estimación



¿Cuál de las siguientes estimaciones es más exacta?

Nuestra inclinación natural

$$esfuerzo = 2.94 * KSLOC^{0.91+0.01 \sum_{j=1}^5 SF_j} * \prod_{i=1}^{17} EM_i$$

En la práctica...

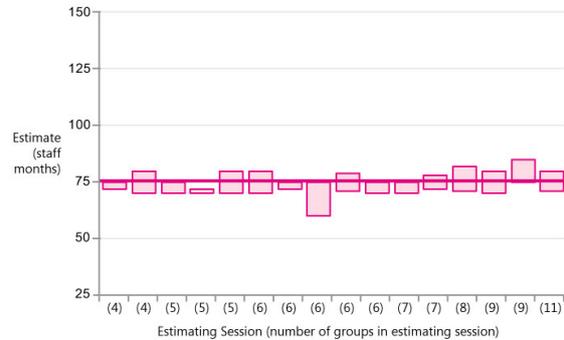
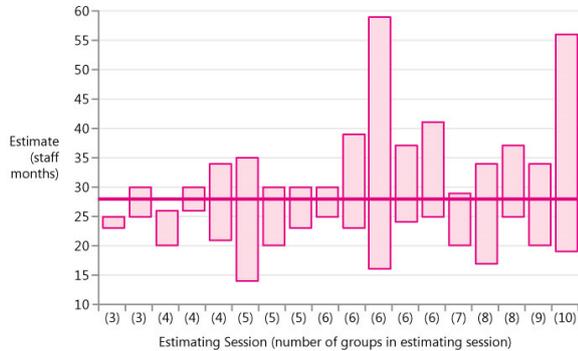
$$esfuerzo = \#requisitos * esfuerzoMedioPorRequisito$$



Técnicas de estimación



Variación en las estimaciones



Numerosos
parámetros
ajustables
Cocomo II (17 effort multipliers)

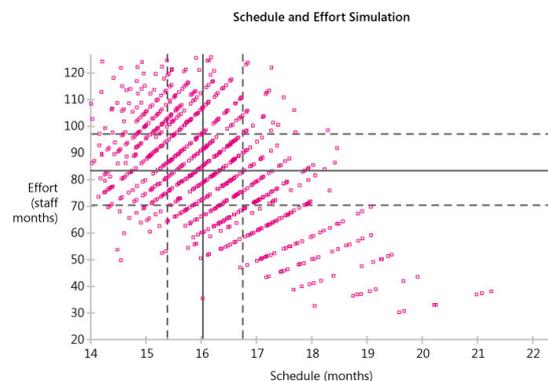
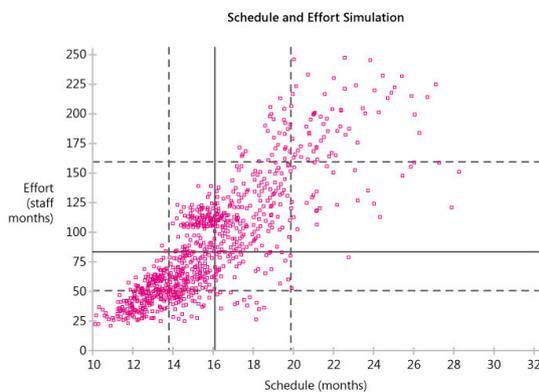
Un único
parámetro
ajustable



Técnicas de estimación



Calibración de los modelos de estimación



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

Datos de la industria
(variación 10x)

Datos históricos
(variación 4x)



Técnicas de estimación



Consejos de Steve McConnell

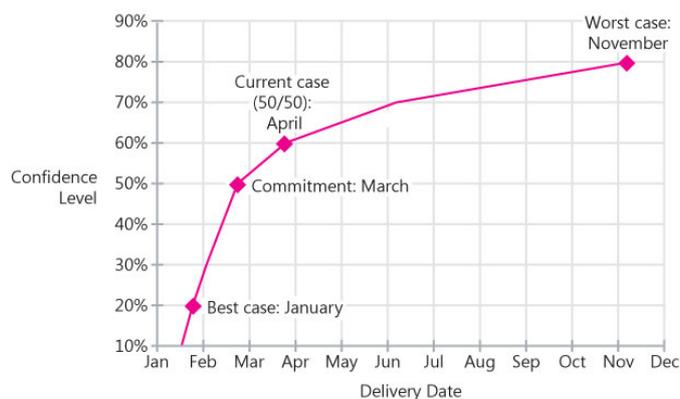
- Evite técnicas con demasiados parámetros que puedan ajustarse subjetivamente (dan una sensación de falsa precisión aunque degradan la precisión real).
- No realice estimaciones a bote pronto. Incluso una estimación realizada en unos minutos será más precisa.
- Ajuste el número de dígitos significativos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



Técnicas de estimación



Presentación de las estimaciones



NOTA:

También se debería incluir el posible impacto de los riesgos asociados al proyecto (cuantificado): $\pm X t$, $\pm X €$



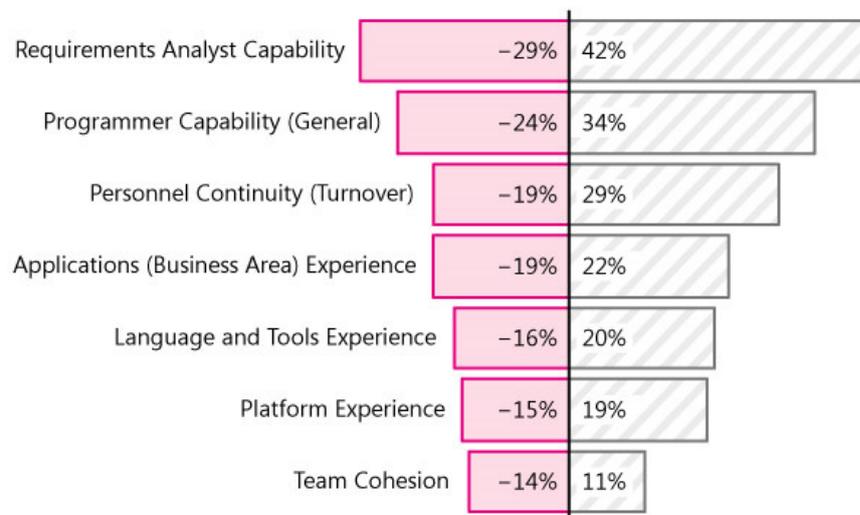
Técnicas de estimación



- El tamaño del proyecto es el factor más importante que determina el esfuerzo y tiempo necesarios: El esfuerzo necesario no aumenta linealmente, sino exponencialmente (deseconomías de escala).
- El tipo de software que se desarrolla en el segundo factor más significativo que influye en el esfuerzo necesario (p.ej. factores de ajuste).
- El equipo del proyecto también ejerce una influencia significativa sobre el esfuerzo requerido para completar el proyecto...



Técnicas de estimación



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



Técnicas de estimación



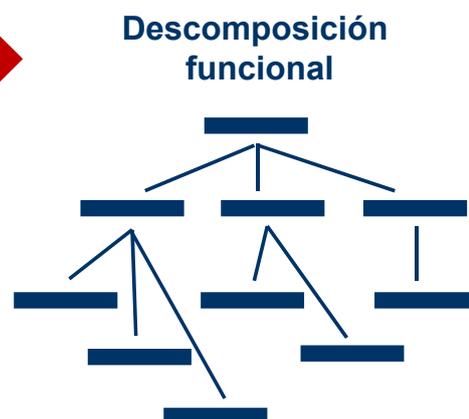
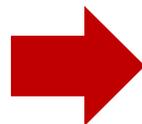
- Estimación por descomposición.
- Estimación de tamaño (líneas de código [KLOC], puntos funcionales [FP]...).
- Estimación por analogía: Experiencia con otros proyectos similares.
- Modelos empíricos
- Herramientas software de estimación



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.

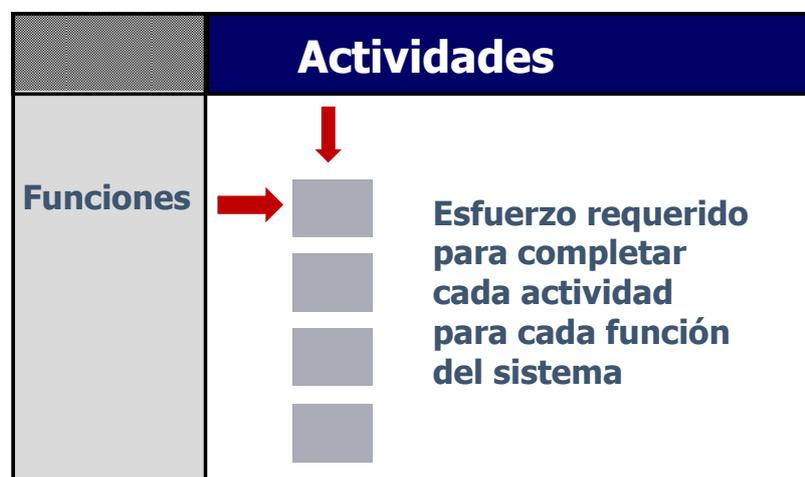
- Costes laborales: 4000€/pm
- Estimación: 36000€



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 (en líneas de código [KLOC], puntos funcionales [FP], casos de uso...).
- Estimar coste y esfuerzo a partir del tamaño estimado (usando datos históricos).



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: 42800€ y 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	5	88
Ficheros	4	4	5	4	10	42
Interfaces externos	2	2	3	2	7	15
Total						320

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



Técnicas de estimación



Estimación del tamaño del proyecto: Casos de uso

Subsistema	Casos de uso	Escenarios	Páginas	KLOC
UI	6	10	6	7.36
Algoritmos	10	20	8	5.15
Infraestructura	5	6	5	6.28
Total				18.79

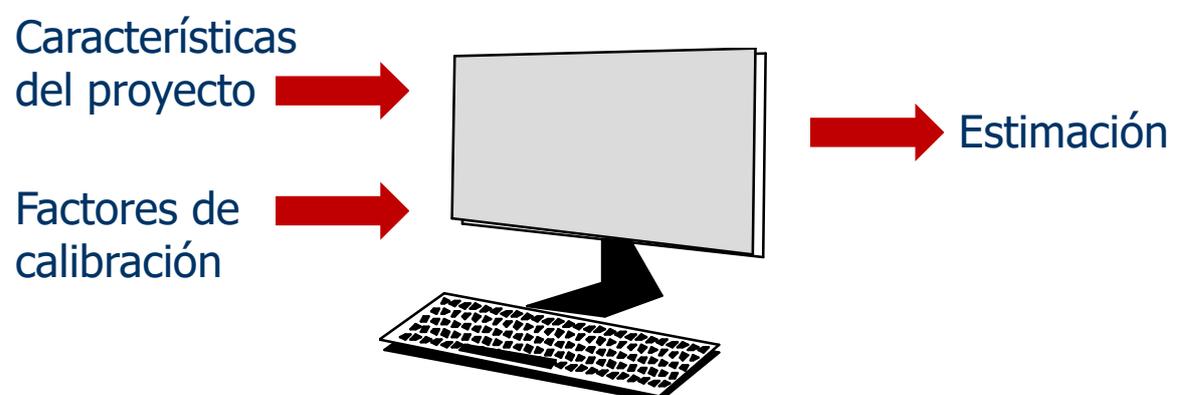
- **KLOC** = $N * KLOC_{avg} + ((S/S_{avg} - 1) + (P/P_{avg} - 1)) * KLOC_{adjust}$
- Productividad media: 2 KLOC / pm [person-month]
- Costes laborales: 4000€/pm (2€/LOC)
- Estimación: 37600€ y 9.5 pm



Técnicas de estimación



Estimación con herramientas software

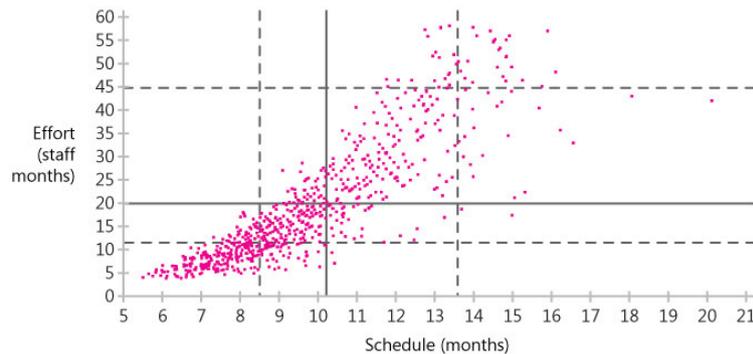


Técnicas de estimación



Estimación con herramientas software

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



Construx Estimate

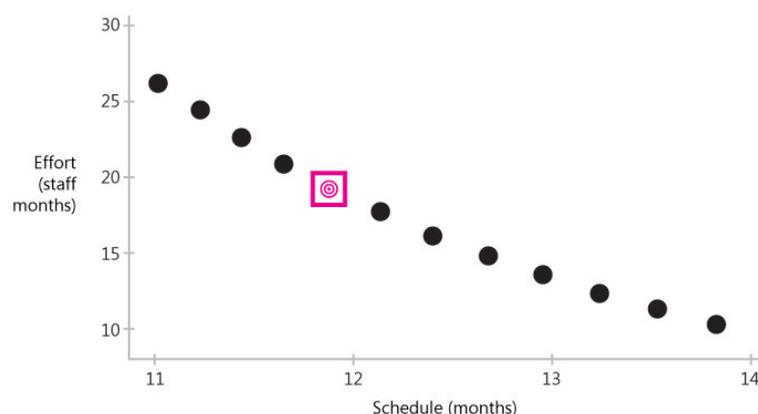


Técnicas de estimación



Estimación con herramientas software

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



Construx Estimate



Técnicas de estimación



Modelos empíricos

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

EJEMPLOS

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

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

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

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

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

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

...

Walston-Felix

Boehm

Doty (KLOC > 9)

Bailey-Basili

Albretch & Gaffney

Kemerer



Técnicas de estimación



Modelos empíricos

IMPORTANTE

Todos estos modelos sólo deben utilizarse una vez calibrados para el entorno concreto en el que se desarrollará un proyecto.



Técnicas de estimación



COCOMO II [CONstructive COst MODEL]

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

<http://csse.usc.edu/tools/COCOMOII.php>

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.

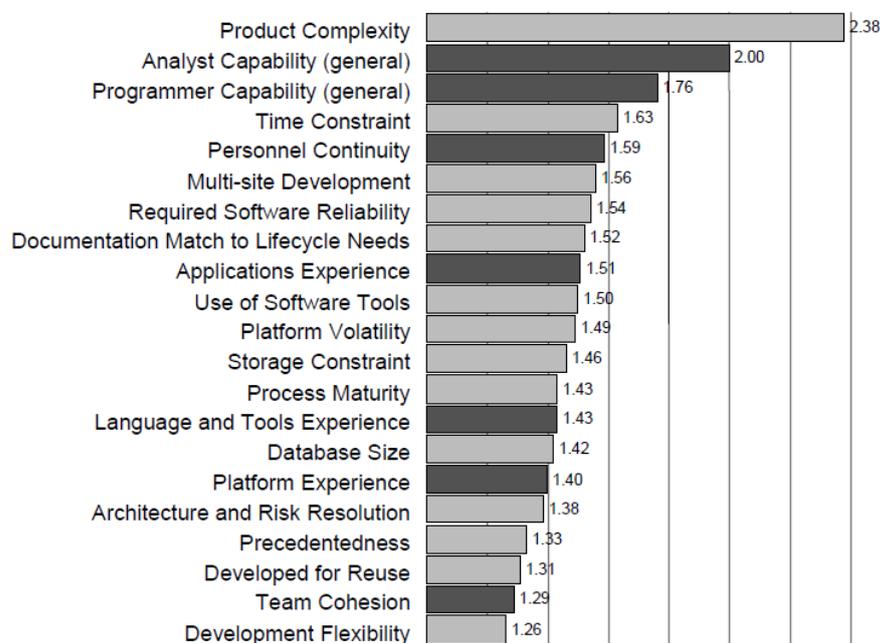


Técnicas de estimación



COCOMO II [CONstructive COst MODEL]

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



Técnicas de estimación



Modelo de Putnam: "La ecuación del software"

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

https://en.wikipedia.org/wiki/Putnam_model

Modelo dinámico multivariable derivado de datos reales recopilados de más de 4000 proyectos:

$$E = B \times \left(\frac{LOC}{P} \right)^3 \times \frac{1}{t^4}$$

- E Esfuerzo (personas-año)
- t Duración del proyecto (años)
- B Factor de escala: "skills factor"
- P Parámetro de productividad



Técnicas de estimación



Modelo de Putnam: "La ecuación del software"

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

$$E = B \times \left(\frac{LOC}{P} \right)^3 \times \frac{1}{t^4}$$

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 al menos 6 meses de duración)

$$t_{min} = 8.14 \frac{LOC}{p^{0.43}}$$

Esfuerzo necesario

(para proyectos de al menos 20 personas·mes)

$$E = 180Bt^3$$



Comentarios finales



¿Una falacia?

To estimate cost and schedule,
first estimate lines of code.

So why is this method, acknowledged to be the most popular in the field, fallacious? Because there is no particular reason why the estimation of LOC is any easier or more reliable than the estimation of cost and schedule...

... this idea would be laughable—in the sense that it is probably harder to know how many LOC a system will contain than what its schedule and cost will be—if it were not for the fact that so many otherwise bright computer scientists advocate it.

-- **Robert L. Glass:**

“Facts and Fallacies of Software Engineering”, 2003

Facts and Fallacies of
Software Engineering



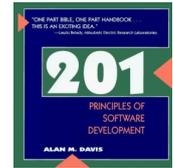
Robert L. Glass
Foreword by Alan W. Watts



Comentarios 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).



Tom Gilb: "Principles of Software Engineering Management", 1988

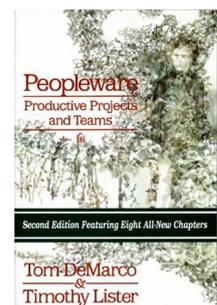
Comentarios finales



¿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 alguien que goce de reputación desde el punto de vista técnico), pero no el gestor...



Comentarios finales

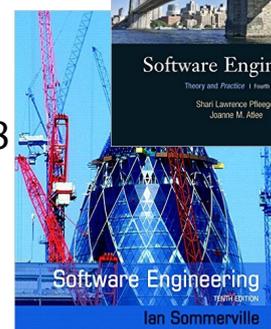
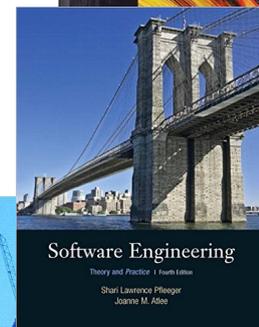
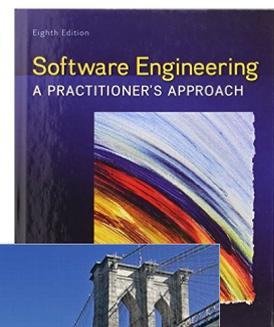


Bibliografía



Libros de texto

- 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

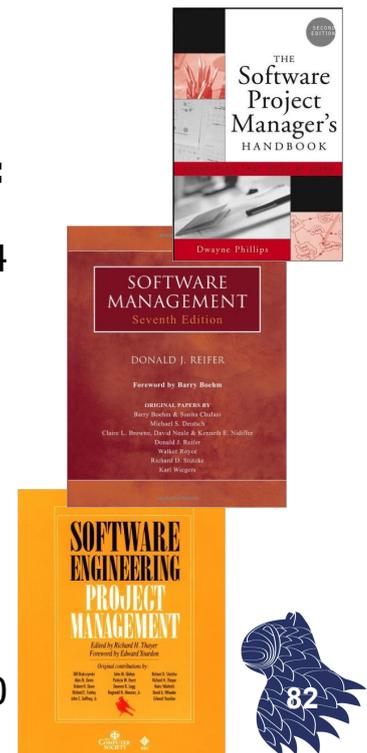


Bibliografía



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.

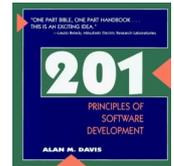
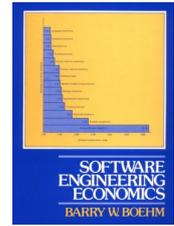
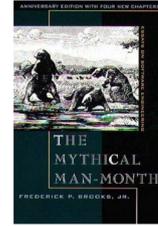


Bibliografía complementaria



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.

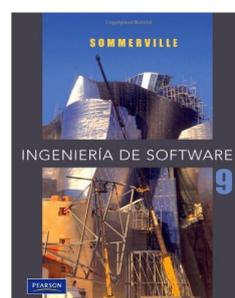


Bibliografía



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



Analice los distintos parámetros que permiten calibrar el modelo de estimación COCOMO II:

$$esfuerzo = 2.94 * KSLOC^{0.91+0.01 \sum_{j=1}^5 SF_j} * \prod_{i=1}^{17} EM_i$$

- 5 factores de escala [SF: scale factors]
- 17 multiplicadores de esfuerzo [EM: effort multipliers]

Sección 3 del manual de COCOMO II

http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

COCOMO® II Cost Driver and Scale Driver Help

http://sunset.usc.edu/research/COCOMOII/expert_cocomo/drivers.html



Ejercicios



Para proyectos del mismo tamaño (en KSLOC), compruebe el efecto que tiene el ajuste de dichos parámetros en diferentes escenarios. En particular, realice estimaciones para los siguientes proyectos:

- Aplicación web desarrollada por un equipo experimentado en el uso de las herramientas necesarias para el proyecto.
- Middleware de alto rendimiento para las construcción de sistemas distribuidos heterogéneos.
- Sistema empotrado para una plataforma hardware actualmente en desarrollo (totalmente novedosa).



Ejercicios



Elabore una tabla en la que se recojan los valores numéricos adecuados de los distintos parámetros para cada uno de los escenarios planteados:

Parámetro	Web App	Middleware	Hardware
SF ₁ PREC	X.XX	X.XX	X.XX
SF ₂ FLEX	X.XX	X.XX	X.XX
SF ₃ RESL	X.XX	X.XX	X.XX
SF ₄ TEAM	X.XX	X.XX	X.XX
SF ₅ PMAT	X.XX	X.XX	X.XX
ΣSF_j	X.XX	X.XX	X.XX
EM ₁
...			
EM ₁₇
ΠEM_i	X.XX	X.XX	X.XX



Ejercicios



Utilizando la expresión general del modelo COCOMO II

$$esfuerzo = 2.94 * KSLOC^{0.91+0.01 \sum_{j=1}^5 SF_j} * \prod_{i=1}^{17} EM_i$$

estime el esfuerzo necesario para desarrollar un proyecto de 10 KSLOC, 100 KSLOC y 1000 KSLOC en cada uno de los escenarios anteriores y rellene la siguiente tabla:

Tamaño	Web App	Middleware	Hardware
10 KSLOC	XX p.m.	XX p.m.	XX p.m.
100 KSLOC	XX p.m.	XX p.m.	XX p.m.
1000 KSLOC	XX p.m.	XX p.m.	XX p.m.





Parámetros del modelo COCOMO II

Table 62. COCOMO II 2000 Calibrated Post-Architecture Model Values

Driver	Symbol	VL	L	N	H	VH	XH
PREC	SF ₁	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	SF ₂	5.07	4.05	3.04	2.03	1.01	0.00
RESL	SF ₃	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	SF ₄	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	SF ₅	7.80	6.24	4.68	3.12	1.56	0.00
RELY	EM ₁	0.82	0.92	1.00	1.10	1.26	
DATA	EM ₂		0.90	1.00	1.14	1.28	
CPLX	EM ₃	0.73	0.87	1.00	1.17	1.34	1.74
RUSE	EM ₄		0.95	1.00	1.07	1.15	1.24
DOCU	EM ₅	0.81	0.91	1.00	1.11	1.23	
TIME	EM ₆			1.00	1.11	1.29	1.63
STOR	EM ₇			1.00	1.05	1.17	1.46
PVOL	EM ₈		0.87	1.00	1.15	1.30	
ACAP	EM ₉	1.42	1.19	1.00	0.85	0.71	
PCAP	EM ₁₀	1.34	1.15	1.00	0.88	0.76	
PCON	EM ₁₁	1.29	1.12	1.00	0.90	0.81	
APEX	EM ₁₂	1.22	1.10	1.00	0.88	0.81	
PLEX	EM ₁₃	1.19	1.09	1.00	0.91	0.85	
LTEX	EM ₁₄	1.20	1.09	1.00	0.91	0.84	
TOOL	EM ₁₅	1.17	1.09	1.00	0.90	0.78	
SITE	EM ₁₆	1.22	1.09	1.00	0.93	0.86	0.80
SCED	EM ₁₇	1.43	1.14	1.00	1.00	1.00	



COCOMO II Scale Factors

Table 10. Scale Factor Values, SF_j, for COCOMO II Models

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
SF _j	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
SF _j	5.07	4.05	3.04	2.03	1.01	0.00
RESL	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
SF _j	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
SF _j	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	The estimated Equivalent Process Maturity Level (EPML) or					
SF _j	SW-CMM Level 1 Lower	SW-CMM Level 1 Upper	SW-CMM Level 2	SW-CMM Level 3	SW-CMM Level 4	SW-CMM Level 5
	7.80	6.24	4.68	3.12	1.56	0.00





COCOMO II Effort Multipliers

Cost Drivers	Very Low	Low	Nominal	High	Very High	Extra High
RELY	slight inconvenience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	
DATA		Testing DB bytes / Pgm SLOC < 10	$10 \leq D/P < 100$	$100 \leq D/P < 1000$	$D/P > 1000$	
CPLX	see Table 19					
RUSE		none	across project	across program	across product line	across multiple product lines
DOCU	Many life-cycle needs uncovered	Some life-cycle needs uncovered.	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	
TIME			$\leq 50\%$ use of available execution time	70%	85%	95%
STOR			$\leq 50\%$ use of available storage	70%	85%	95%
PVOL		major change every 12 mo.; minor change every 1 mo.	major: 6 mo.; minor: 2 wk.	major: 2 mo.; minor: 1 wk.	major: 2 wk.; minor: 2 days	
ACAP	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
PCAP	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
PCON	48% / year	24% / year	12% / year	6% / year	3% / year	



COCOMO II Effort Multipliers

Cost Drivers	Very Low	Low	Nominal	High	Very High	Extra High
APEX	≤ 2 months	6 months	1 year	3 years	6 years	
PLEX	≤ 2 months	6 months	1 year	3 years	6 year	
LTEX	≤ 2 months	6 months	1 year	3 years	6 year	
TOOL	edit, code, debug	simple, frontend, backend CASE, little integration	basic lifecycle tools, moderately integrated	strong, mature lifecycle tools, moderately integrated	strong, mature, proactive lifecycle tools, well integrated with processes, methods, reuse	
SITE: Collocation	International	Multi-city and multi-company	Multi-city or multi-company	Same city or metro area	Same building or complex	Fully collocated
SITE: Communication	Some phone, mail	Individual phone, FAX	Narrow-band email	Wide-band electronic communication.	Wide-band elect. comm, occasional video conf.	Interactive multimedia
SCED	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	





COCOMO II Component Complexity Rating

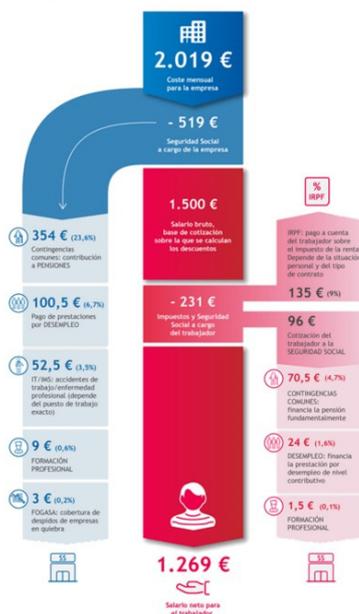
Table 19. Component Complexity Ratings Levels

	Control Operations	Computational Operations	Device-dependent Operations	Data Management Operations	User Interface Management Operations
High	Highly nested structured programming operators with many compound predicates. Queue and stack control. Homogeneous, distributed processing. Single processor soft real-time control.	Basic numerical analysis: multivariate interpolation, ordinary differential equations. Basic truncation, round-off concerns.	Operations at physical I/O level (physical storage address translations; seeks, reads, etc.). Optimized I/O overlap.	Simple triggers activated by data stream contents. Complex data restructuring.	Widget set development and extension. Simple voice I/O, multimedia.
Very High	Reentrant and recursive coding. Fixed-priority interrupt handling. Task synchronization, complex callbacks, heterogeneous distributed processing. Single-processor hard real-time control.	Difficult but structured numerical analysis: near-singular matrix equations, partial differential equations. Simple parallelization.	Routines for interrupt diagnosis, servicing, masking. Communication line handling. Performance-intensive embedded systems.	Distributed database coordination. Complex triggers. Search optimization.	Moderately complex 2D/3D, dynamic graphics, multimedia.
Extra High	Multiple resource scheduling with dynamically changing priorities. Microcode-level control. Distributed hard real-time control.	Difficult and unstructured numerical analysis: highly accurate analysis of noisy, stochastic data. Complex parallelization.	Device timing-dependent coding, micro-programmed operations. Performance-critical embedded systems.	Highly coupled, dynamic relational and object structures. Natural language data management.	Complex multimedia, virtual reality, natural language interface.



Costes asociados a un contrato laboral en España

Nómina de un trabajador temporal



Nómina de un trabajador indefinido

