

# Planificación y Gestión de Proyectos Informáticos

Técnicas de estimación

Jose M. Moyano

`jmoiano@ugr.es`

**Departamento de Ciencias de la Computación e Inteligencia Artificial**

**Universidad de Granada**

Curso 2022-23



**DaSCI**

Instituto Andaluz de Investigación en  
Data Science and Computational Intelligence



**UNIVERSIDAD  
DE GRANADA**

# Índice

El ámbito del proyecto

¿Qué es una estimación?

Estimaciones

Cono de incertidumbre

Realización de estimaciones

Técnicas de estimación

- Estimación por descomposición

- Estimación por tamaño

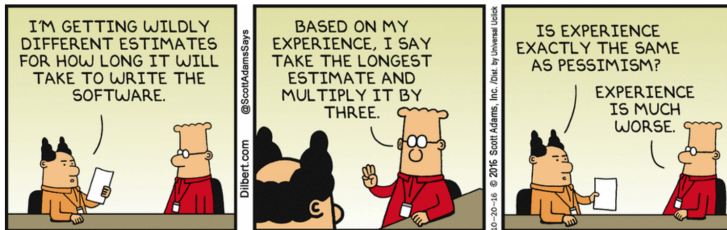
- Estimación con herramientas software

- Modelos empíricos

  - COCOMO II

Comentarios Finales

# Estimación



# 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 (¿qué debe hacer?) y casi tan importante: ¿qué NO debe hacer? (que es lo que queda fuera).
- ▶ **Restricciones:** ¿Qué requisitos debe satisfacer el sistema?

# El ámbito del proyecto

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

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

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

# El principio W<sup>5</sup>HH

- ▶ 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?

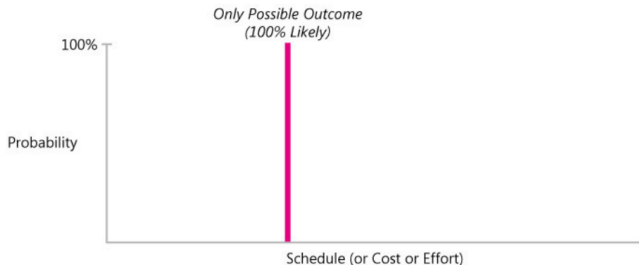
# Descripción del ámbito del proyecto

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”.

# Estimación puntual

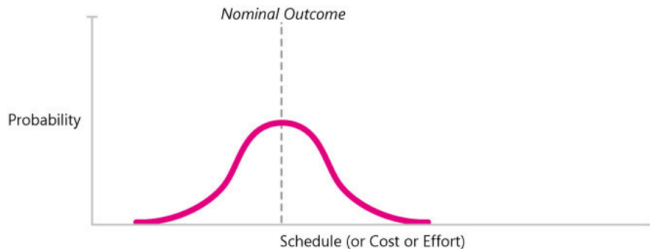
Asume un 100 % de probabilidad de que el resultado real sea exactamente igual que el esperado.



Muy poco realista...

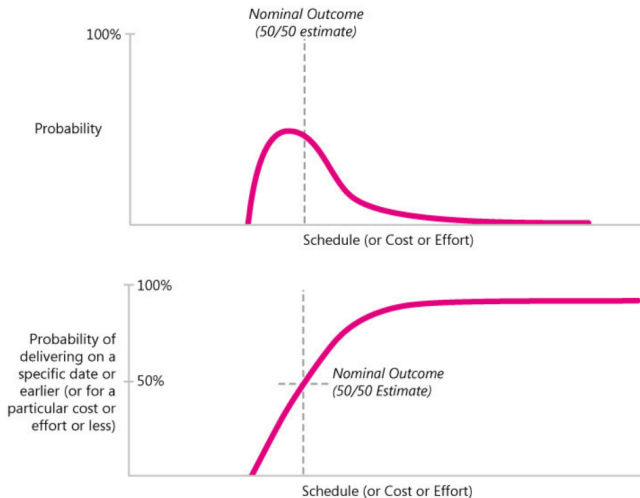


# Estimación normal/gaussiana

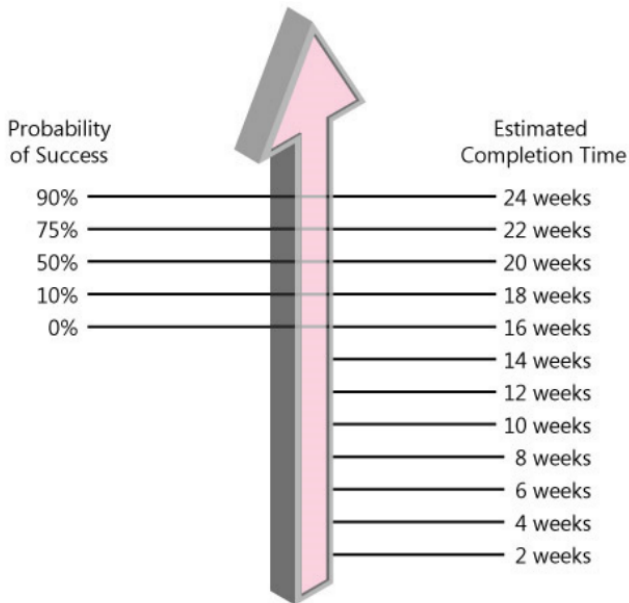


Basada en suposiciones incorrectas: existen límites en cuanto a la eficiencia con la que un equipo puede completar cualquier tarea que se le encomiende.

# Estimación realista



# ¿Qué es una estimación?

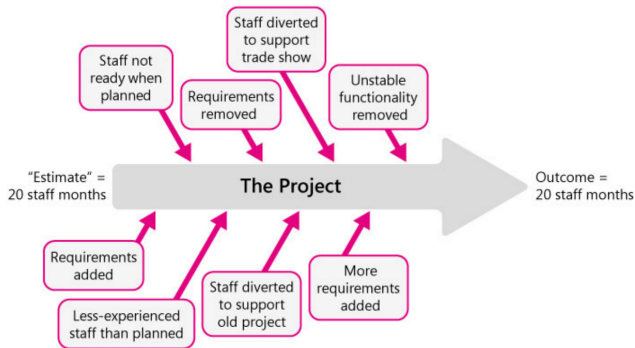


# Consejos de Steve McConnell

- ▶ Distinga entre estimaciones, objetivos y compromisos.
- ▶ Cuando le pidan una estimación, intente tener claros los objetivos.
  - ▶ ¿Para qué necesita esta estimación? ¿Qué está estimando? ¿Está estimando solo algunas actividades de implementación o el proyecto al completo? ¿Está estimando solo el esfuerzo requerido, o el esfuerzo más vacaciones, adaptación y aprendizaje de nuevo personal, y otras actividades no tan relacionadas con el proyecto?
- ▶ 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?

Factores que influyen en la precisión de una estimación



Algunos los podemos controlar, otros no...

# ¿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

# ¿Cómo de bueno eres estimando?

Da un intervalo estimado de:

1. Temperatura de la superficie del Sol.
2. Latitud de Shanghai.
3. Superficie de Asia.
4. Volumen de los Grandes Lagos
5. Longitud de la línea costera del Pacífico.
6. Libros publicados al año en EEUU desde 1776.
7. Taquilla global de la película Titanic.
8. Dólares en circulación en 2004.
9. Año de nacimiento de Alejandro Magno.
10. Peso de la mayor ballena azul jamás registrada.

# ¿Cómo de bueno eres estimando?

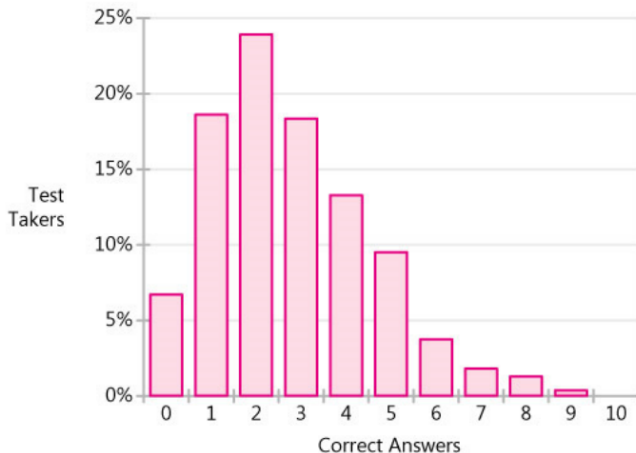
Da un intervalo estimado de:

1. Temperatura de la superficie del Sol. **6000°C**
2. Latitud de Shanghai. **31º Norte**
3. Superficie de Asia. **44.339.000 km<sup>2</sup>**
4. Volumen de los Grandes Lagos. **23000 km<sup>3</sup>**
5. Longitud de la línea costera del Pacífico. **135663 km**
6. Libros publicados al año en EEUU desde 1776. **22 millones.**
7. Taquilla global de la película Titanic. **\$ 1835 millones**
8. Dólares en circulación en 2004. **\$ 719.900 millones**
9. Año de nacimiento de Alejandro Magno. **356 A.C.**
10. Peso de la mayo ballena azul jamás registrada. **170.000 kg**



# Resultados del test

Si realmente se hubieran realizado estimaciones con un 90 % de confianza, debería haber acertado 9 respuestas....



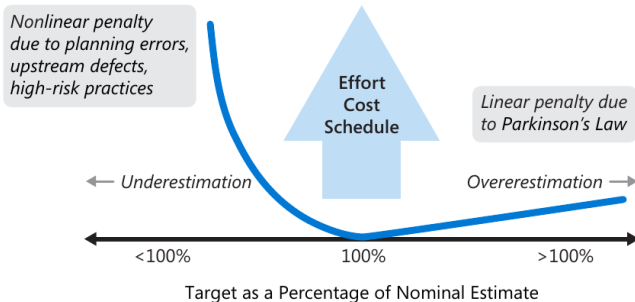
Nº de medio de respuestas correctas: 2.8

# Consejos de McConnell

- ▶ No proporcione un porcentaje de confianza para sus estimaciones 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 estos intervalos, asegúrese que la presión proviene de una fuente externa y no de usted mismo.

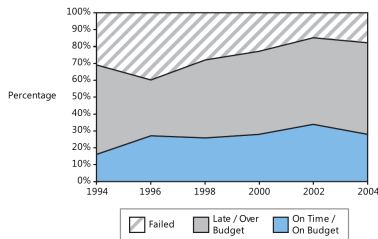
# Estimaciones

¿Es mejor subestimar o sobrestimar?



No subestimes de manera deliberada...

# 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

Conclusión: Proyectos más grandes tienen más probabilidad de fallar (o retrasarse)

# Deseconomías de escala

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

①



Communication path  
with 2 people

③



Communication paths  
with 3 people

⑥



Communication paths  
with 4 people

⑩



Communication paths  
with 5 people

④⑤

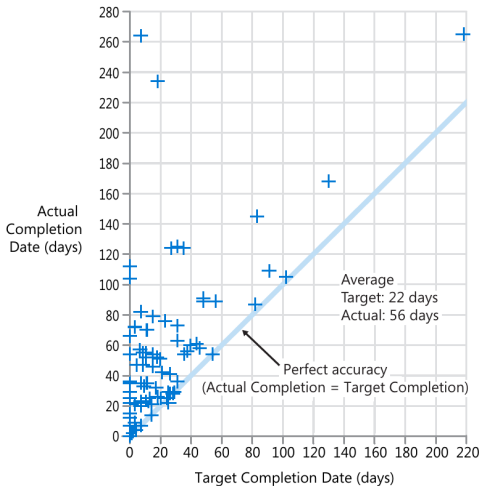


Communication paths  
with 10 people

Conforme aumenta le tamaño del proyecto, el número de canales de comunicación aumenta exponencialmente.

# Estimaciones

Tal y como hemos visto anteriormente, es preferible sobrestimar. Sin embargo, las subestimaciones son más frecuentes (y peligrosas):



# Estimaciones

Las subestimaciones no siempre son malas...

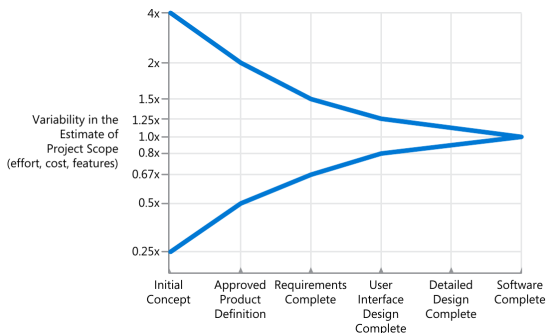
- ▶ El equipo de un proyecto que va **ligeramente** por detrás del calendario previsto tiende a trabajar más para ponerse al día.
- ▶ El equipo que va por delante del calendario previsto tiende a relajarse: Ley de Parkinson.

Si la subestimación es excesiva, la motivación y la productividad se desploman.

... pero son peligrosas: **la zona imposible**

# Cono de incertidumbre

Inicialmente, las estimaciones realizadas presentan mucha incertidumbre: cuanto más avanzado esté el proyecto, menos incertidumbre tendremos, por lo que las estimaciones tienden a ser más precisas.



Estos errores no se pueden reducir hasta que no se avanza en el proyecto (y se eliminan los problemas...)



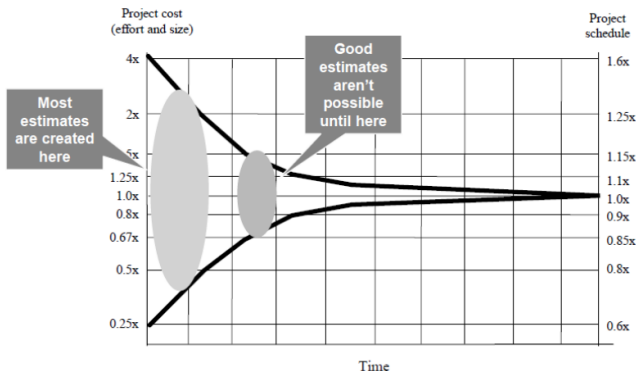
# Cono de incertidumbre

El cono de incertidumbre representa el mejor caso; si no se gestiona bien el proyecto, puede haber otras fuentes de error, como:

- ▶ Proyecto mal gestionado (fuera de control).
- ▶ Requisitos inestables.
- ▶ Actividades omitidas en las estimaciones.
- ▶ Optimismo en las estimaciones

# Realización de estimaciones

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



o peor aún: la estimación inicial se realiza antes de que se definan los requisitos...

# Realización de estimaciones

## Puntos clave:

- ▶ Se pueden realizar estimaciones precisas, aunque las estimaciones precisas requieren tiempo.
- ▶ Las estimaciones precisas requiere el uso de técnicas cuantitativas, preferiblemente apoyadas por alguna herramienta software.
- ▶ Las estimaciones más precisas se basan en datos **reales** de proyectos ya completados por la organización encargada del proyecto actual (**y no en sus estimaciones**).
- ▶ 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

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.

# 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.
- ▶ etc.

# Realización de estimaciones

## Algunas 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

Para obtener buenas estimaciones:

- ▶ Descomponemos el proyecto.
- ▶ Realizamos **dos o más** estimaciones utilizando técnicas diferentes.
- ▶ Reconciliamos las diferentes estimaciones.

Recordad que, conforme avanza el proyecto las estimaciones se hacen con mayor grado de detalle/precisión (al reducirse la incertidumbre).



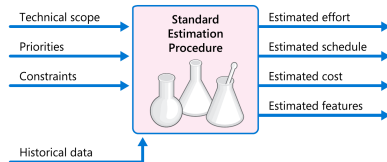
# Técnicas de estimación

## Proyectos mal estimados:



Tanto entradas, como procesos de estimación y salidas no están bien definidos y están abiertos a debate → Tiende a subestimar.

## Proyectos bien estimados:



Si se definen bien entradas y procesos, modificar la salida para “estar a gusto” es un sinsentido.

# Técnicas de estimación

Subjetividad vs exactitud:

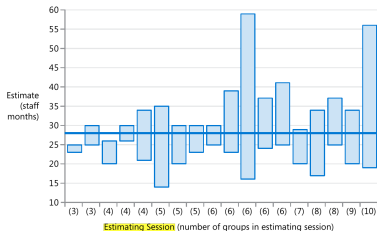
¿Cuál de estas dos estimaciones es más exacta?

$$E = 2,94 * KSLOC^{0,91+0,01 \sum_{j=1}^5 SF_j} * \prod_{i=1}^{17} EM_i \quad (1)$$

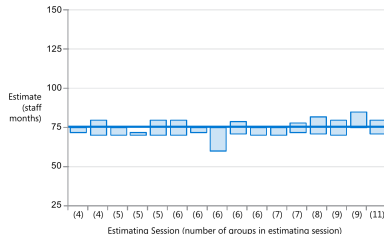
$$E = \#Requisitos * esfuerzoMedioPorRequisito \quad (2)$$

# Técnicas de estimación

Cuanto más parámetros tenga nuestro modelo más posibilidad hay de introducir elementos subjetivos y que las estimaciones sean muy variables:



Numerosos parámetros  
(Cocomo II - 17 params.)



Un único parámetro

# Calibración de los modelos de estimación

La calibración se emplea para convertir valores a estimaciones, p.ej. líneas de código a esfuerzo, historias de usuario a tiempo de calendario, etc. Para calibrar, se pueden emplear tres fuentes de datos diferentes (De menor a mayor precisión):

- ▶ Datos de la industria. Datos sobre proyectos similares en otras empresas del sector
- ▶ Datos históricos de la empresa. Datos de rendimientos pasados de la empresa
- ▶ Datos del proyecto. Datos de rendimiento en etapas anteriores del proyecto.

# Técnicas de estimación

Factores clave que determinan el esfuerzo y tiempo:

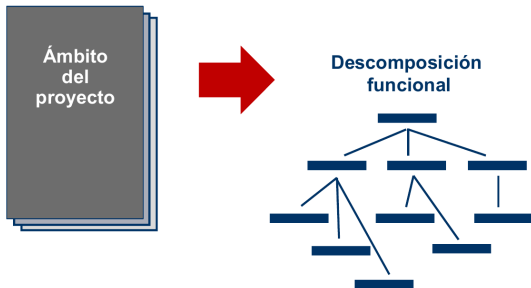
- ▶ Tamaño del proyecto: el esfuerzo y tiempo necesario no aumentan linealmente, sino exponencialmente (deseconomías de escala).
- ▶ El tipo de software: el esfuerzo requerido para software de sistemas críticos es mucho mayor que para otro tipo de software mas convencional. Es necesario aplicar factores de ajuste.
- ▶ El factor humano. El equipo de trabajo influye significativamente en el sobre el esfuerzo del proyecto.

# Técnicas de estimación

Existen una gran cantidad de técnicas de estimación:

- ▶ Estimación por descomposición.
- ▶ Estimación de tamaño: líneas de código (KLOC), puntos funcionales (FP)...
- ▶ Estimación por analogía.
- ▶ Modelos empíricos.
- ▶ Herramientas software de estimación.

# Estimación por descomposición funcional



# Estimación por descomposición funcional

<b>Módulo</b>	<b>Esfuerzo estimado</b>
Base de datos	1 pm
Modelo de clases	1 pm
Algoritmos X	2 pm
Interfaz de usuario	4 pm
Comunicaciones	1 pm
<b>TOTAL</b>	<b>9 pm</b>

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



# 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
<b>TOTAL</b>	0.25	2.75	2.25	2.20	3.00	10.45
%	2 %	26 %	22 %	21 %	29 %	

► Costes laborales: 4000€/pm

► Estimación: 41800€

# Estimación en 3 puntos

- ▶ Este proceso es aplicable a la mayoría de estimaciones más básicas.
- ▶ En lugar de dar una única estimación, estimamos tres valores distintos: valor optimista, valor más probable, y valor pesimista.
  - ▶ La estimación será probablemente más precisa.
  - ▶ Calculamos la duración esperada como:  
$$(pesimista + 4 * masProbable + optimista) / 6$$
  - ▶ Calculamos la desviación estándar de la actividad como:  
$$(pesimista - optimista) / 6$$

# Estimación del tamaño del proyecto

- ▶ Estimar el tamaño del proyecto (en líneas de código, puntos funcionales, casos de uso, etc.)
- ▶ Estimar coste y esfuerzo a partir del tamaño estimado (usando datos históricos).

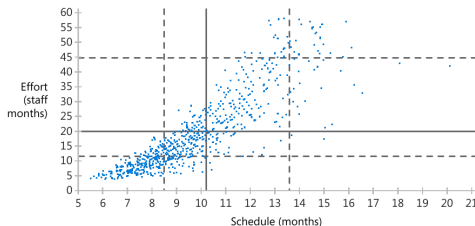
# Estimación del tamaño del proyecto: KLOC

Módulo	Esfuerzo 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
<b>TOTAL</b>	<b>21.4 KLOC</b>

- ▶ Productividad media para sistemas de este tipo: 2 KLOC/pm
- ▶ Costes laborales: 4000€/pm (2€/LOC)
- ▶ Estimación: 42800€ y 11 pm

# Estimación con herramientas software

Las herramientas software permiten realizar simulaciones estadísticas que nos permiten incorporar fuentes de variabilidad a nuestras estimaciones:



**IMPORTANTE:** hay que usar los inputs reales del proyecto y los factores de calibración.

## ► Construx estimate

- [https://web.archive.org/web/20170920093850/https://www.construx.com/Thought\\_Leadership/Books/Survival\\_Guide/Resources\\_By\\_Chapter/Construx\\_Estimate/](https://web.archive.org/web/20170920093850/https://www.construx.com/Thought_Leadership/Books/Survival_Guide/Resources_By_Chapter/Construx_Estimate/)

# Modelos empíricos

$$\text{esfuerzo} = A + B * \text{tam}^c$$

## EJEMPLOS

$$E = 5,2KLOC^{0,91}$$

Walston-Felix

$$E = 3,2KLOC^{1,05}$$

Boehm

$$E = 5,288KLOC^{1,047}$$

Doty (KLOC > 9)

$$E = 5,5 + 0,73KLOC^{1,16}$$

Bailey-Basili

$$E = -0,91 + 0,355FP$$

Albretch & Gaffney

$$E = -37 + 0,96FP$$

Kemerer

IMPORTANTE: Todos estos modelos deben usarse calibrados para el entorno concreto del proyecto.

# COCOMO II

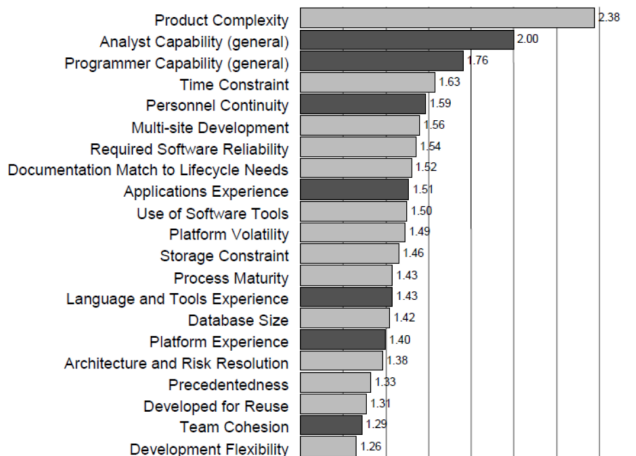
Es una jerarquía de modelos de estimación que tienen en cuenta las diferentes “etapas” del proyecto software.

- ▶ Estimación del tamaño: KLOC, FP, puntos “de objetos” (OP).
- ▶ Ajuste por reutilización:  $NOP = OP * (100 - \%reuse) / 100$
- ▶ Productividad:  $PROD = NOP / pm$
- ▶ Esfuerzo estimado:  $EFF = NOP / PROD$

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

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

# COCOMO II





# Comentarios finales

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

- ▶ Tú, como gestor de proyecto (capaz de destruir la motivación).
- ▶ Las suposiciones utilizadas al realizar la estimación pueden no ser las adecuadas.
- ▶ La probabilidad (la estimación es solo un punto en una distribución de probabilidad).

# ¿Quién debería hacer las estimaciones?

Es recomendable que el propio desarrollador responsable de la tarea sea quien proporcione una estimación, en lugar de el gestor.

Es el que va a realizar el trabajo y será más productivo con un plazo autoimpuesto (o, al menos, consensuado). Además, tendrá mayor conocimiento y experiencia en la tarea a realizar que el gestor del proyecto, por lo que las estimaciones deberían ser más precisas.

## 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

## 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

## 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
- ▶ Richar 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

## Clásicos

- ▶ Freferic 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 Managemenet Rules** NASA Software Engineering Laboratory, SEL-91-001, 1991

# Planificación y Gestión de Proyectos Informáticos

Técnicas de estimación

Jose M. Moyano

`jmoyano@ugr.es`

**Departamento de Ciencias de la Computación e Inteligencia Artificial**

**Universidad de Granada**

Curso 2022-23



**DaSCI**

Instituto Andaluz de Investigación en  
Data Science and Computational Intelligence



**UNIVERSIDAD  
DE GRANADA**