

# Personal Software Process<sup>SM</sup>

## Estimación con PROBE I

This material is approved for public release. Distribution is limited by the Software Engineering Institute to attendees.

Sponsored by the U.S. Department of Defense

© 2006 by Carnegie Mellon University



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Tópicos

Revisión de la Planificación

Por qué estimar el tamaño?

Principios de estimación del tamaño

El método de estimación PROBE

Tamaño de la estimación de proxies

Examples de Estimación



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

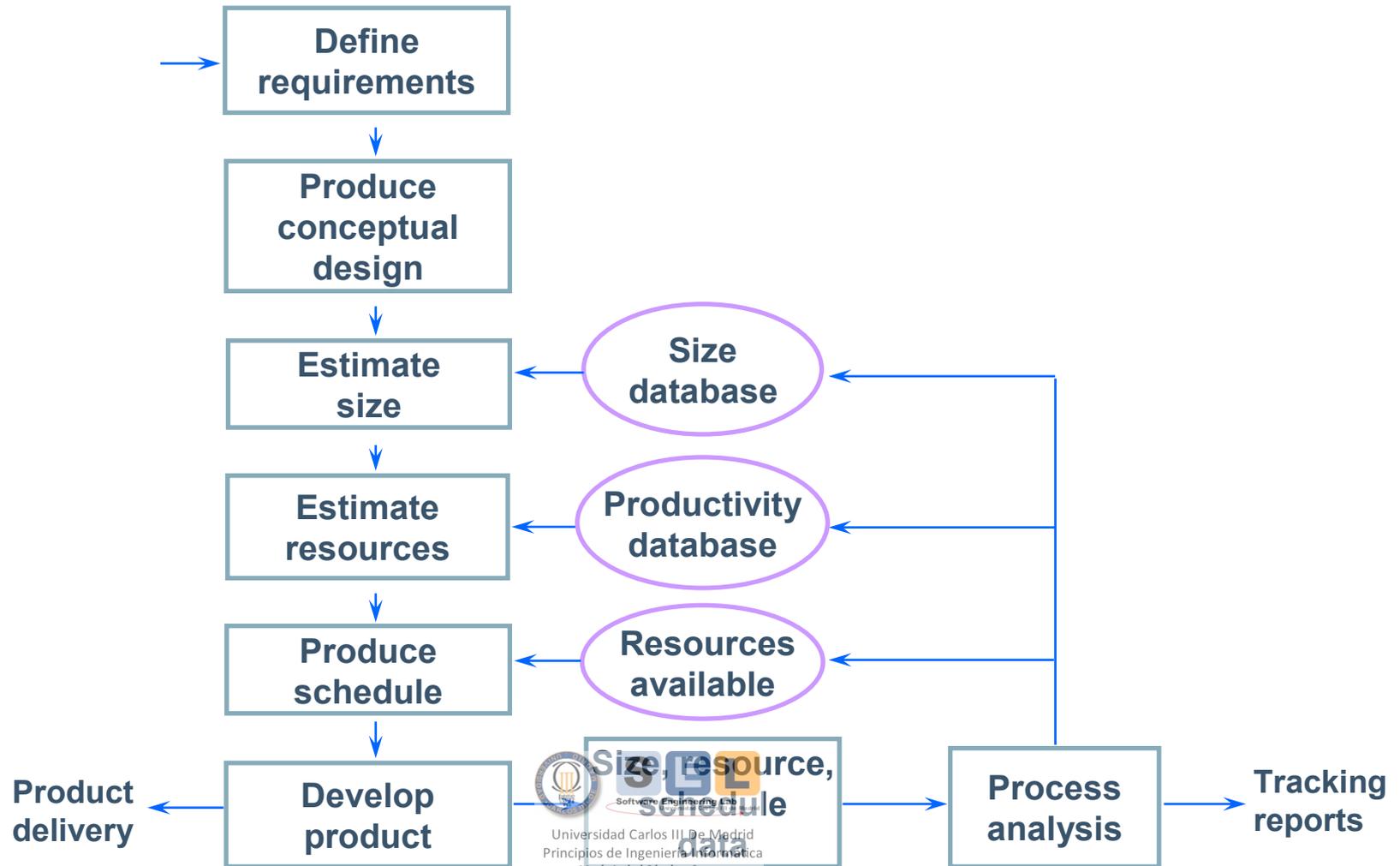
# Revisión de la Planificación

## Planes

- Permitir realizar compromisos que se puedan cumplir
- Proveer las bases para acuerdos sobre el ámbito del trabajo, cronograma y recursos
- Guiar el trabajo
- Facilitar el progreso del seguimiento y reportes
- Ayudar a asegurar que las tareas claves no se pasen por alto



# El Framework de Planificación del Proyecto



# El Proceso de Planificación

Antes de realizar el plan, se debe de tener un requisito.

- Entre mejor sea el requisito, mejor el plan
- Con requisitos mal definidos, espere realizar frecuentes actualizaciones del plan.

Planes son más precisos cuando están basados en estimaciones del tamaño e información histórica.

Con una estimación del tamaño e información histórica, se puede

- Identificar la información
- Identificar la información de la mayoría de los trabajos anteriores.
- Basar la estimación del recurso sobre esta información



# Principios de Estimación del Tamaño -1

Estimación es un proceso incierto.

- Nadie sabe cuan grande será el producto
- Entre más temprana la estimación, es menos conocida.
- Estimación puede ser influenciada por negocios y otras presiones.

Estimación es un proceso de aprendizaje intuitivo.

- Mejora de habilidades con experiencia e información.
- Algunas personas serán mejores al estimar que otras.



# Principios de Estimación del Tamaño -2

Estimación es una habilidad.

- La mejora puede ser gradual
- Nunca podrá ser muy buena

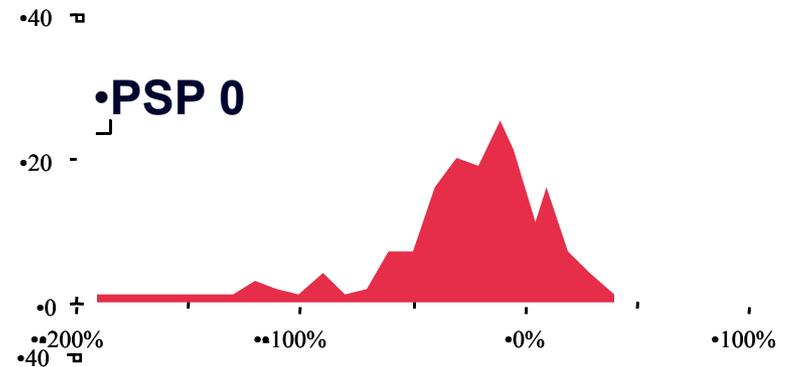
El objetivo es llegar a ser consistente.

- Entender la variabilidad de sus estimaciones
- Buscar un balance entre sub y sobreestimado

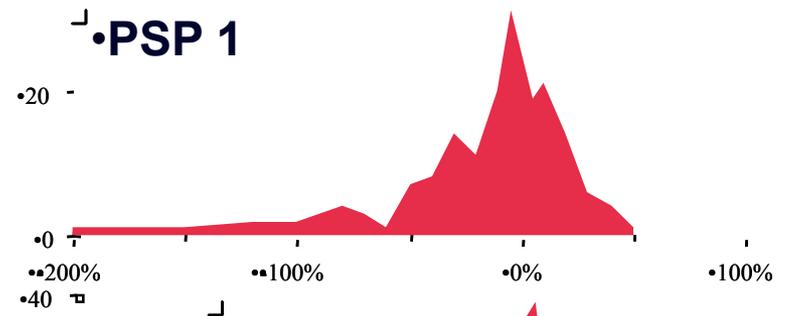


# Estimaciones Balanceadas

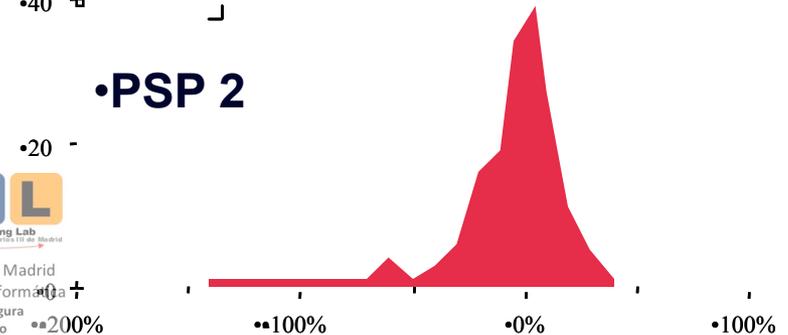
Mayormente son sub estimaciones



Balancear sobre y sub estimaciones



Muchos balances ajustados alrededor de cero



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Principios de Estimación del Tamaño -3

Las ventajas de usar métodos de estimación definidos son aquellas que:

- Se tienen prácticas conocidas que se pueden mejorar
- Tienen un framework para recolectar información de estimación
- Pueden consistentemente usar información histórica para producir estimaciones balanceadas



# Estimando con PROBE

PSP usa el método PROBE para estimar y planificar proyectos.

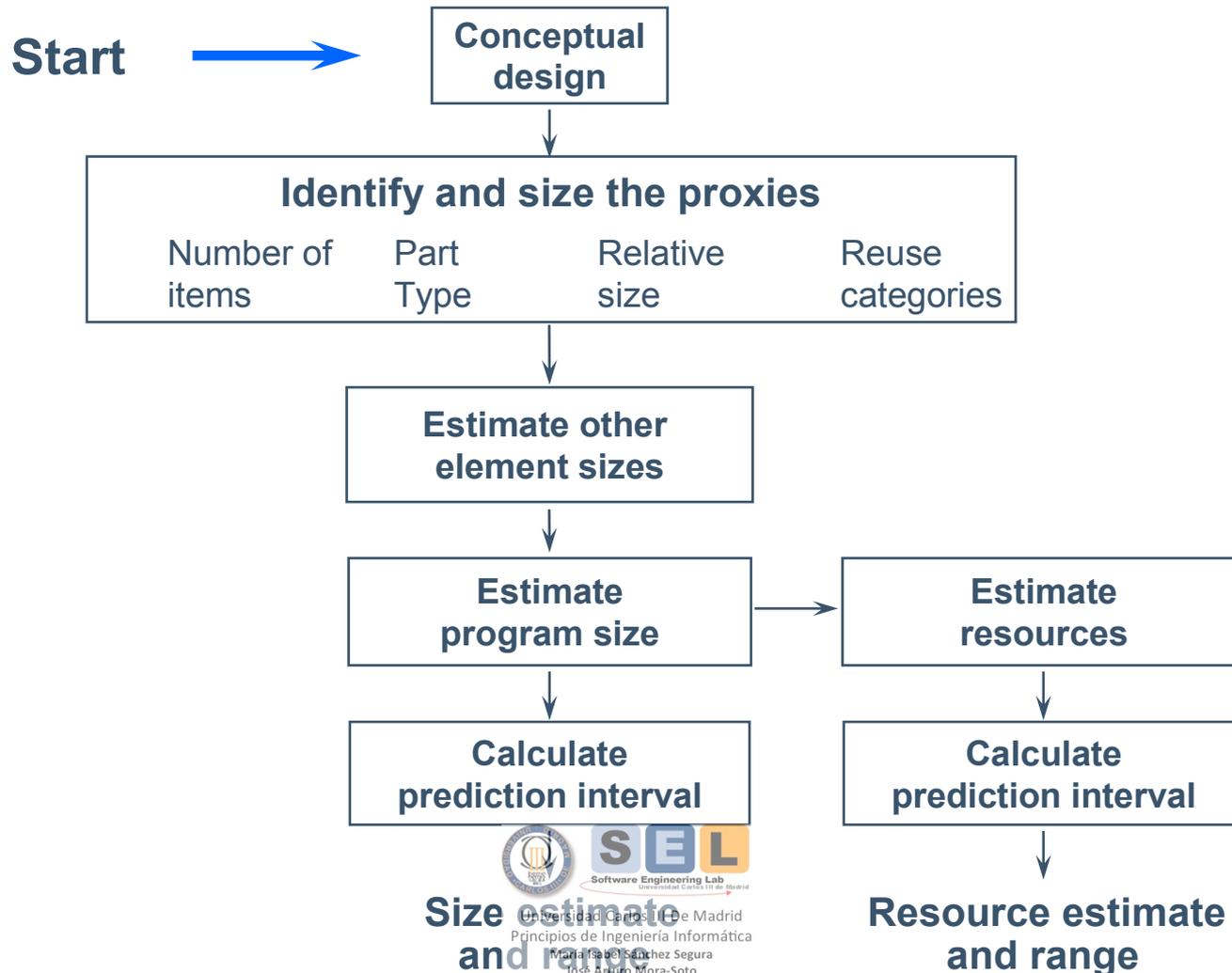
PROBE sirve de apoyo para la estimación basada en PROxy.

PROBE usa proxies para estimar el tamaño del programa y el tiempo de desarrollo.

Un buen proxy ayudará a realizar estimaciones precisas.



# El Método de Estimación PROBE



# Diseño Conceptual -1

El primer paso de estimación es hacer un diseño conceptual.

- Relacionar los requisitos para el producto
- Definir los elementos del producto que producirán las funciones deseadas
- Estimar el tamaño de lo que se planea construir

Para muchos proyectos, el diseño conceptual puede ser relativamente producido rápidamente.

Para los programas de los ejercicios de PSP, intenta limitar el tiempo del diseño conceptual a 10, o a los mucho 20 minutos.



# Diseño Conceptual -2

Para realizar un diseño conceptual, identificar las funciones del producto y las partes del programa necesarias para producirlas.

De hecho, dices: “Si tuviera las siguientes partes, podría construir este producto.”

Entonces, compara estas partes para programas que tienes ya escritos y estima sus tamaños.

Finalmente, combina las partes estimadas para obtener el tamaño total.

Si no entiendes el producto bastante bien para realizar un diseño conceptual, no sabes lo suficiente para hacer un plan.

# Tamaño de la estimación de Proxies -1

La cuestión básica

- Buenas medidas de tamaño son detalladas.
- Es generalmente difícil visualizar detalles iniciales del producto en un proyecto.

Alternativas

- Esperar a estimar hasta que se tenga el detalle.
- Realiza tu mejor conjetura
- Identificar un proxy apropiado



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Tamaño de la estimación de Proxies -2

Un buen proxy debería correlacionar cercanamente a los costes de desarrollo.

Un buen proxy debería ser fácil para visualizar tempranamente en el desarrollo

Debería también ser una entidad física que puede ser medida.



# Ejemplo: Costes de Construcción

## Problema

- El constructor necesita saber el área del piso para estimar el coste de la construcción.
- Clientes normalmente no pueden describir sus necesidades en pies cuadrados.
- Ellos usualmente pueden describir el tipo y número de salas que ellos desean.

## Solución

- Usar salas como un proxy para el tamaño
- Usar información histórica para traducir salas a pies cuadrados.



# Ejemplo: Requisitos del Cliente

Dormitorios: 1 grande, 2 medianos, y 1 pequeño

Baños: 1 grande y 2 medianos

Cocina: 1 mediano

Sala: 1 grande

Comedor: 1 mediano

Sala familiar: 1 grande

Almacén de herramientas: 1 mediano



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Datos históricos de Construcción

	Pequeño	Mediano	Grande
Dormitorios	90	140	200
Baños	25	60	120
Cocinas	100	130	160
Salas	150	250	400
Comedores	100	140	200
Salas familiares	150	240	340
Almacén de herramientas		50	80

# Cálculo del Proxy

Dormitorio	1 grande	=	1 x 200	=	200
Dormitorio	2 mediano	=	2 x 140	=	280
Dormitorio	1 pequeño	=	1 x 90	=	90
Baño	1 grande	=	1 x 120	=	120
Baño	2 mediano	=	2 x 60	=	120
Cocina	1 mediano	=	1 x 130	=	130
Sala	1 grande	=	1 x 400	=	400
Comedor	1 mediano	=	1 x 140	=	140
Sala familiar	1 grande	=	1 x 340	=	340
Almacén de herramientas	1 mediano	=	1 x 50	=	50



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

**Total (pies<sup>2</sup>.) = 1870**

# Ejemplo: La estimación del constructor

El primer paso de estimación provee al constructor con la información del proxy para el tamaño del ambiente.

Sin embargo, hay muchos otros elementos de coste en la construcción de la casa.

Constructores típicamente tienen amplia información para relacionar el tamaño de un ambiente con los costes de construcción.

Con planes iniciales acordados, los constructores típicamente requieren detalladas especificaciones arquitecturales y estimaciones antes de citar un precio.

# Ejemplo Proxies Producto

Clases, funciones, y procedimientos

Elementos del producto

- Elementos de la base de datos
- Pantallas, reportes, scripts, archivos
- Capítulos del libro



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Clases como Proxies -1

Correlación con horas de desarrollo

- Números de clases que correlacionan razonablemente bien
- Tamaño de la clase que correlaciona muy cercanamente
- Tamaño de la clase puede ser estimada usando datos históricos
- Estimación del tamaño del programa es luego calculada a partir de la relación histórica entre el tamaño de la clase total y el tamaño del programa

Con una buena correlación, calcule el tamaño del programa a partir de la relación entre el tamaño de la clase y el tamaño del programa

# Clases como Proxies -2

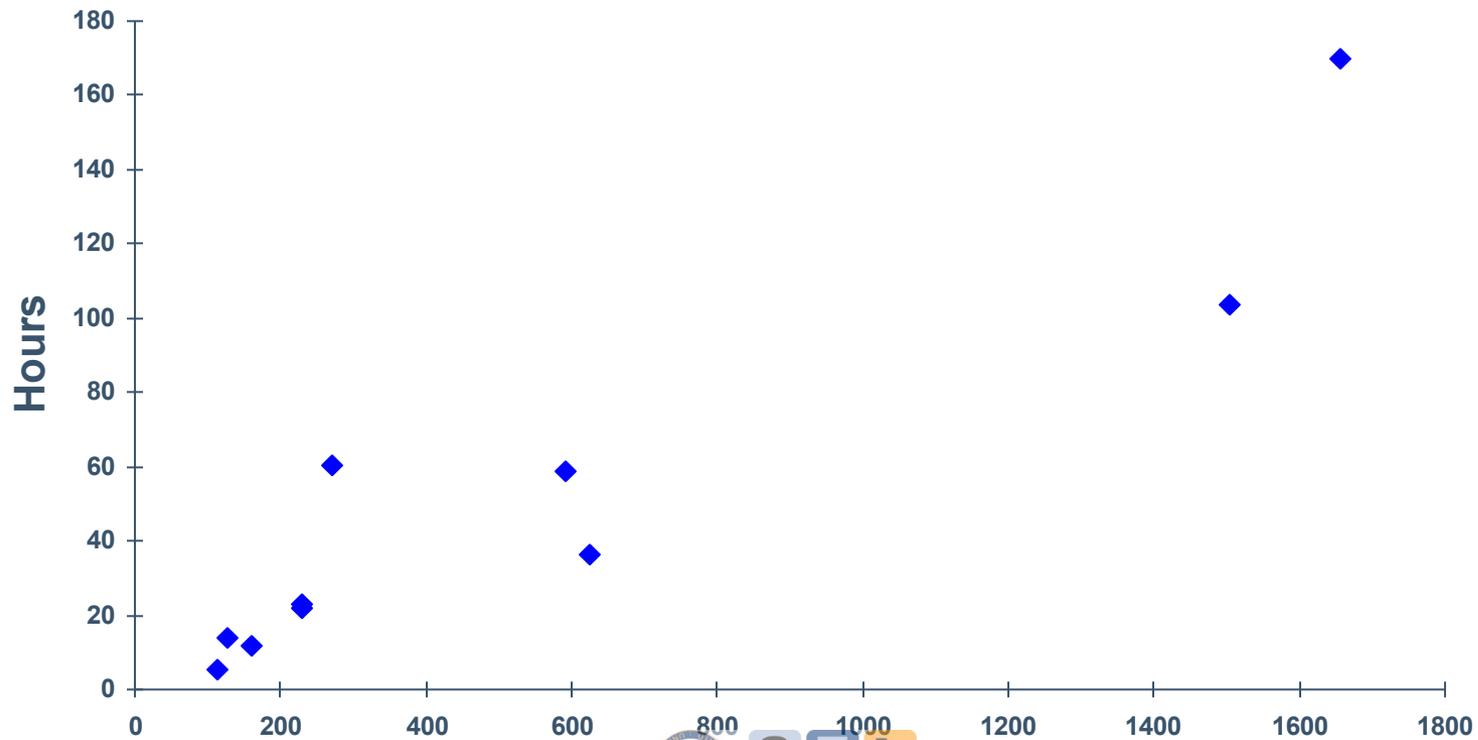
Cuando las clases son seleccionadas como entidades de aplicación, pueden ser visualizadas empezando el desarrollo

Funciones y procedimientos pueden frecuentemente ser estimados de la misma forma

Clases, funciones, procedimientos, y sus tamaños pueden ser automáticamente contados.



# Correlación entre LOC Clases y Horas de Desarrollo



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Otros Ejemplos de Proxies

Posibles candidatos

- Pantallas, reportes, scripts, archivos
- Capítulos del libro

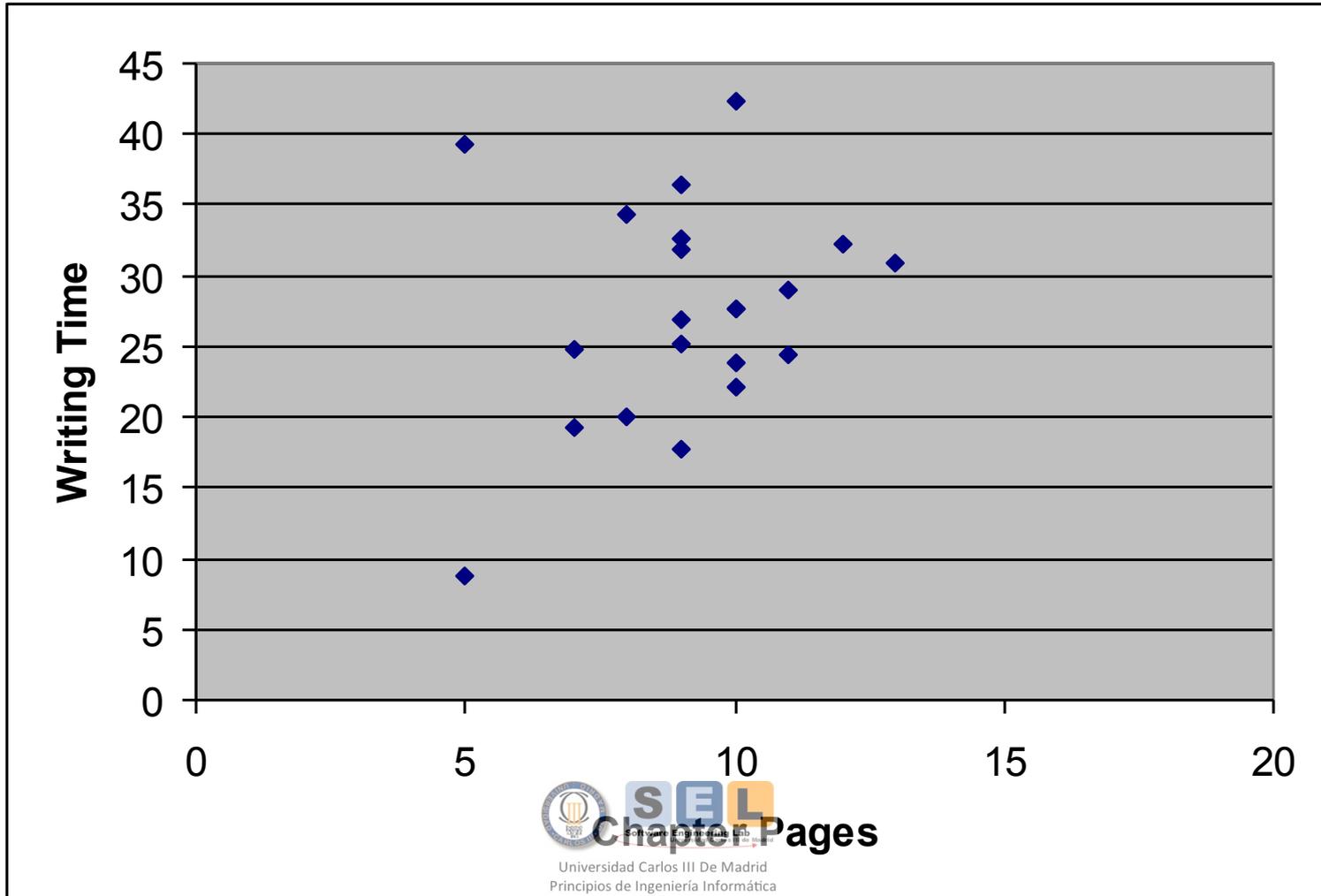
Si el número de items correlaciona con el desarrollo, estime el número de items.

Con un apropiado tamaño del proxy, se puede estimar el tamaño del proxy repetidas veces.



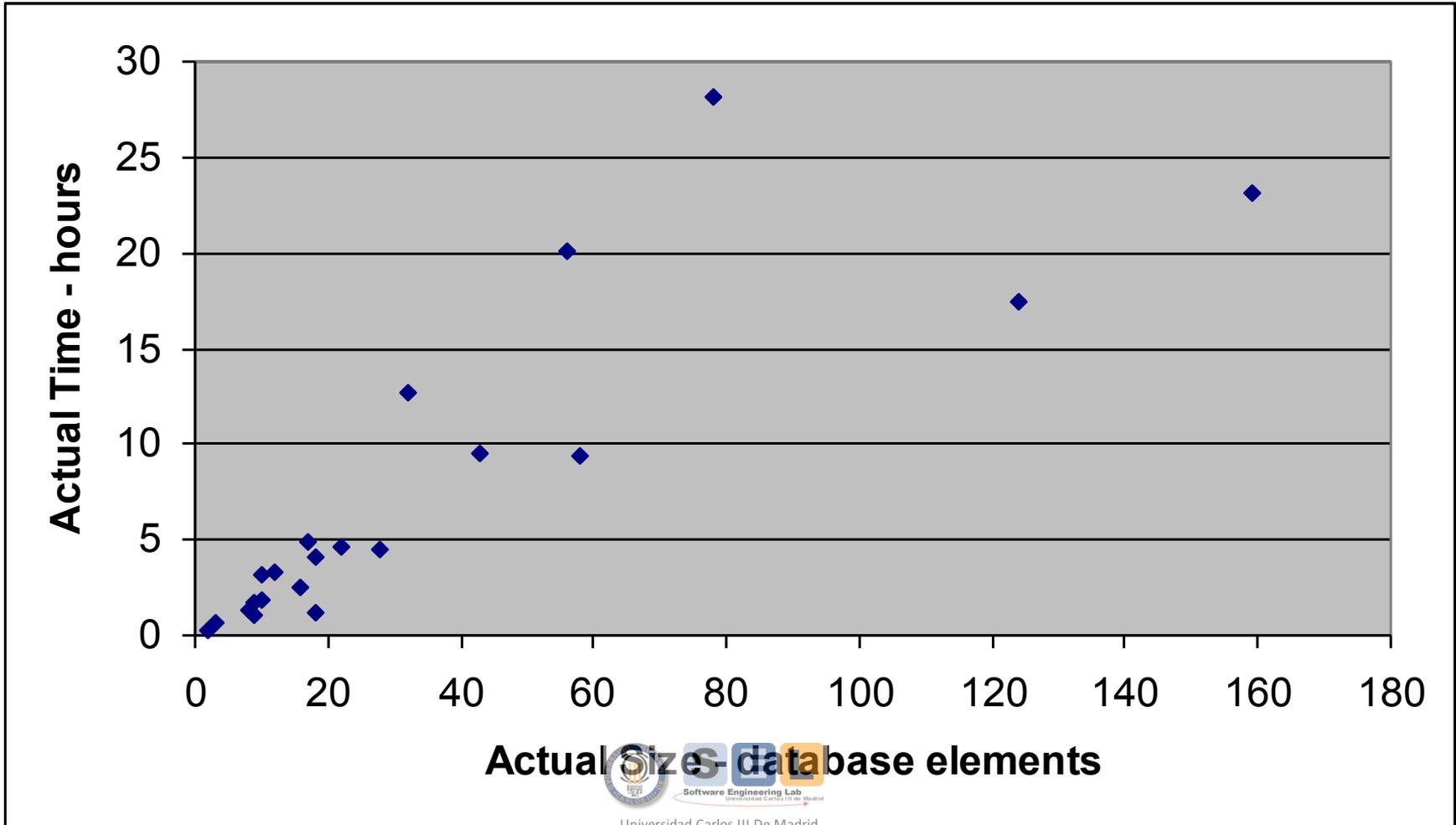
Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Páginas del capítulo Versus Tiempo



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Elementos de la base de datos Versus Tiempo



# Estimando con Proxies

Una vez que se ha seleccionado un proxy, debe

- Obtener información del proxy
- Organizar la información para usarla en la estimación
- Usar la información para estimar el tamaño de los proxies en el nuevo programa
- Combinar las estimaciones del proxy con la estimación del producto
- Realizar una estimación del recurso
- Producir un plan del proyecto

El método PROBE muestra cómo realizar estos pasos.



# Organizando datos del Proxy

Una forma común para estimar es

- Fraccionar el producto planificado en partes
- Relacionar estas partes planificadas con partes ya construidas
- Usar el tamaño de las partes previamente construidas para estimar los tamaños de las nuevas partes

Para hacer esto, se necesitan rangos de tamaño para los tipos de partes que típicamente desarrolla.

Para cada tipo de producto, también se necesitan rangos de tamaños para ayudar a tener una idea de los tamaños de las nuevas partes



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Ejemplo Rangos de tamaño de clase en C++

## LOC por ítem

Tipo	VS	S	M	L	VL
Cálculo	2.34	5.13	11.25	24.66	54.04
Datos	2.60	4.79	8.84	16.31	30.09
I/O	9.01	12.06	16.15	21.62	28.93
Lógica	7.55	10.98	15.98	23.25	33.83
Set-up	3.88	5.04	6.56	8.53	11.09
Texto	3.75	8.90	17.07	36.41	77.66



# Estimando el tamaño del programa

Precisamente como las casas tienen pies cuadrados que no están en habitaciones, los programas tienen código que no está en partes del programa.

- inclusiones
- declaraciones
- otras funciones en general

El trabajo de desarrollo, sin embargo, debe también producir código general.

El tamaño de este código adicional general es usualmente proporcional al tamaño de las partes del programa.



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Estimando el Tiempo de Desarrollo

Con métodos de estimación acertados, el tamaño real del programa estará cercanamente relacionado al tamaño del programa estimado.

Las diferencias se deberán al código general y al error de estimación.

El tiempo de desarrollo real es también frecuentemente relacionado al tamaño del programa estimado.

De nuevo, con métodos acertados, las diferencias serán mayormente debido a las actividades generales y de estimación del error.



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Estimaciones Basadas Estadísticamente

PROBE usa datos históricos, regresión lineal, y el intervalo de predicción para producir estimaciones de precisión conocida.

La regresión provee el mejor ajuste, o mínima varianza, de una línea para estos datos.

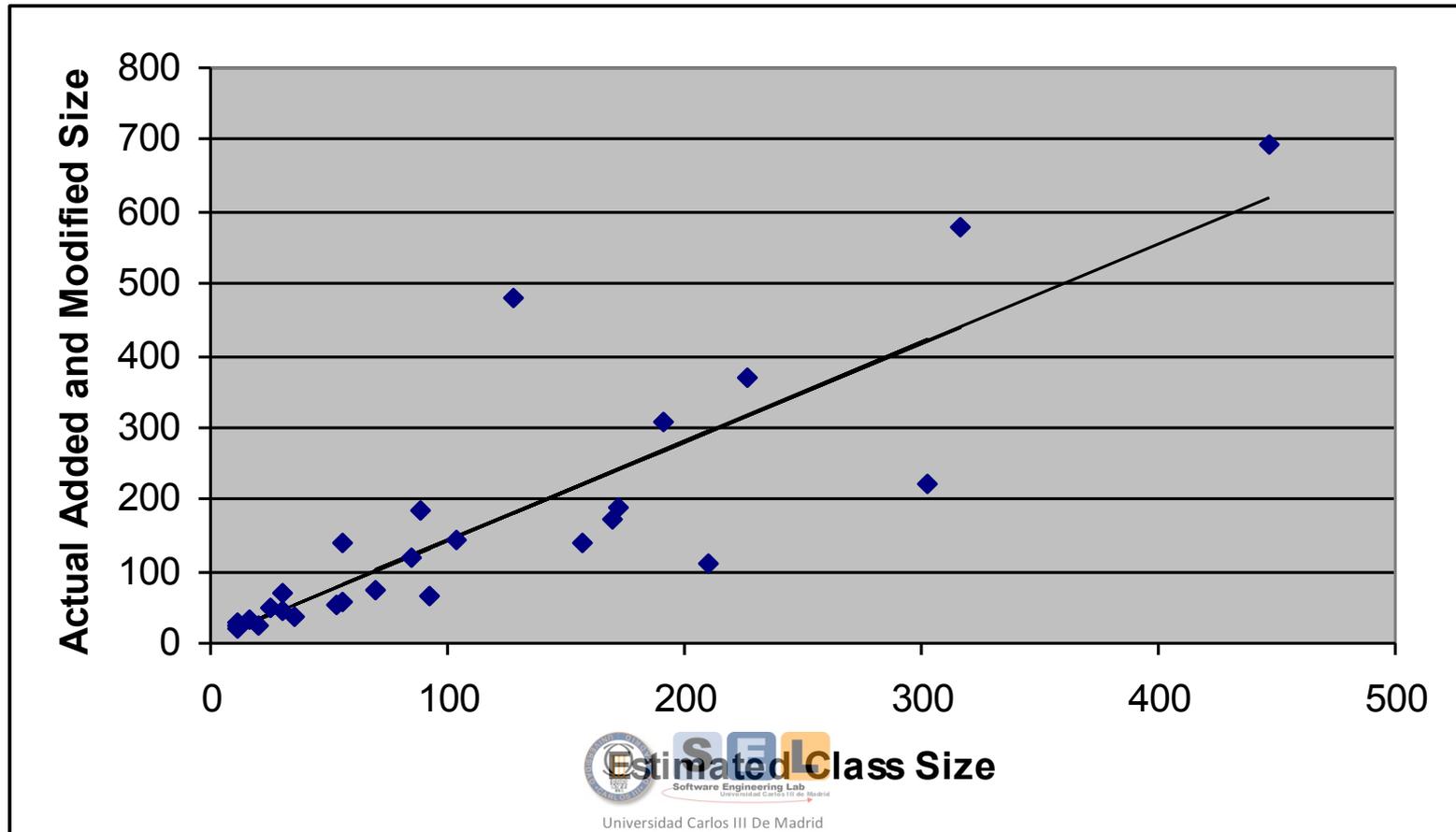
La varianza de los datos es usada para determinar la probabilidad de estimación del error.

A mayor varianza, mayor probabilidad de error.



# Línea de regresión para el tamaño del programa

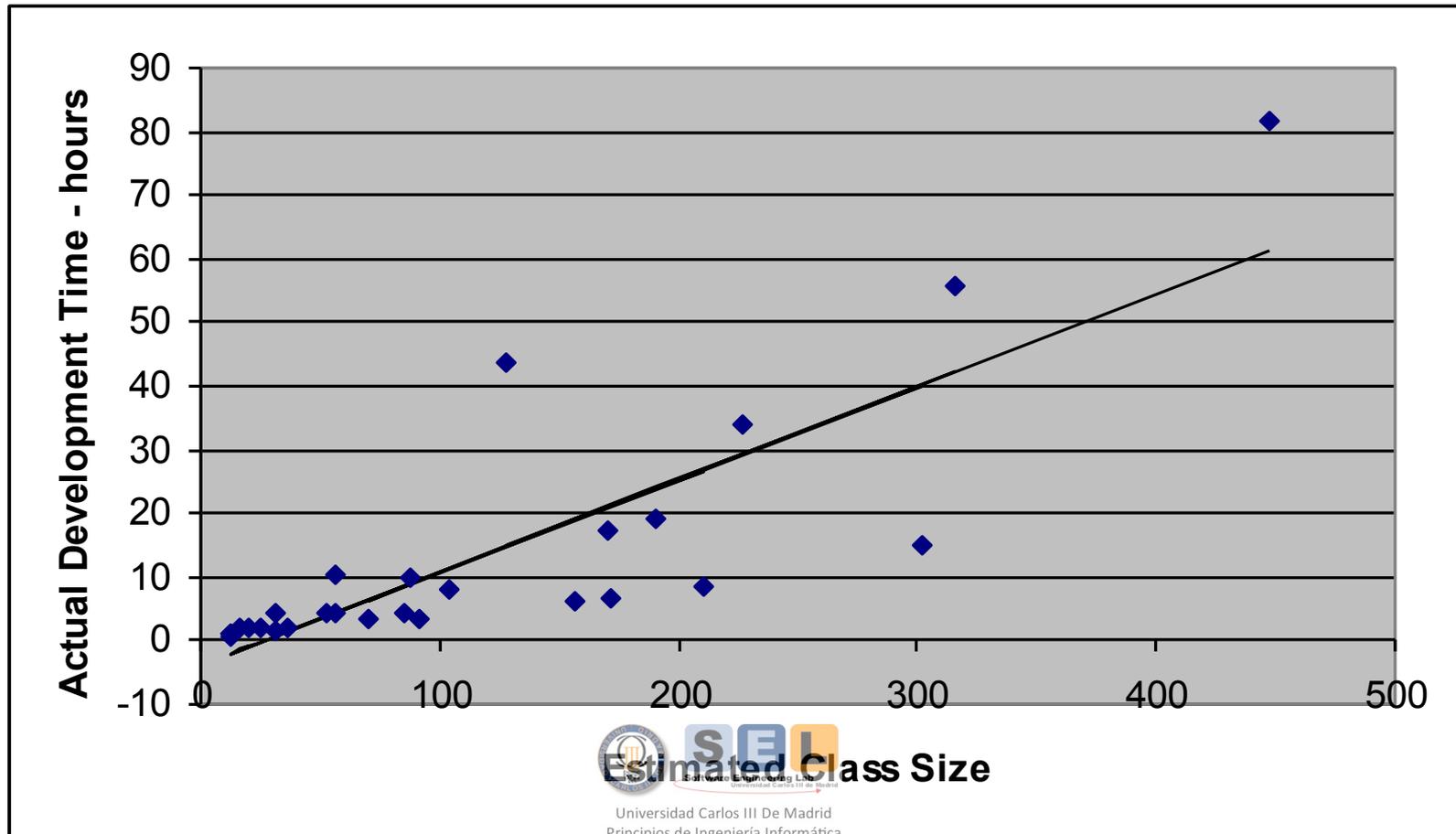
27 programas C++



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Línea de regresión para el tiempo de desarrollo

27 programas C++



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Estimando el Tamaño y el Tiempo -1

Para proyectar el tamaño o tiempo de desarrollo del programa, halle la ecuación para la regresión lineal.

Luego use la fórmula de regresión para proyectar el tamaño y el tiempo.

E es el tamaño del proxy estimado.

$$y_k = \beta_0 + \beta_1 * x_k$$

$$\textit{Added \& Modified \_ Size} = \beta_{0size} + \beta_{1size} * (E)$$

$$\textit{Development \_ Time} = \beta_{0time} + \beta_{1time} * (E)$$



# Estimando el Tamaño y el Tiempo -2

Calcular los parámetros de regresión  $\beta_0$  y  $\beta_1$  a partir de la información de los programas previamente desarrollados.

- Para  $x$ , use el tamaño estimado del proxy (E).
- Para  $y$ , use el actual
  - Tamaño añadido y modificado para la estimación del tamaño
  - Tiempo total de desarrollo para la estimación del tiempo

Calcular dos conjuntos de los parámetros de regresión lineal  $\beta_0$  y  $\beta_1$ : uno para el tamaño y otro para el tiempo.



# Calculando el tamaño total del Programa

El tamaño total del programa incluye código añadido, modificado, borrado, base y reusado.

Al modificar un programa existente, el código base es el tamaño del programa existente sin modificación.

Al modificar programas, incluye su tamaño sin modificar en el código base y no en el reuso.

Mientras que el código base es una forma de reuso, PSP solo cuenta código sin modificar de la librería de reuso como reusado.



# Completando la Estimación -1

Para completar el tamaño de estimación, calcular

- El tamaño añadido y modificado proyectado con los parámetros de regresión del tamaño
- El tamaño total del programa, incluyendo código añadido, modificado, borrado, base y reusado.
- Código nuevo reusable estimado a ser añadido a la librería de reuso

La completa estimación incluye tiempo de desarrollo estimado calculado con los parámetros de regresión del tiempo



# Completando la Estimación -2

Con el tamaño y tiempo estimados, calcular

- Intervalos de predicción superiores (UPI) e inferiores (LPI) al 70% para el tamaño del programa proyectado
- Intervalos de predicción superiores (UPI) e inferiores (LPI) al 70% para el tiempo del programa proyectado



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán

# Ejemplo Completado -1

Programa Base (B)	695 LOC
Borrado (D)	0 LOC
Modificado (M)	5 LOC
Adiciones a la Base (BA)	0 LOC
Partes añadidas (PA)	$115+197+49 = 361$ LOC
Partes reusadas (R)	169 LOC
Tamaño estimado E	$= BA + PA + M = 366$ LOC



# Ejemplo completado -2

Empezando con  $E = 366$ , use los parámetros de regresión del tamaño para calcular el tamaño proyectado (P).

$$\beta_{0\text{Size}} = 62 \text{ y } \beta_{1\text{Size}} = 1.3$$

$$\text{Tamaño proyectado } P = 62 + 1.3 * 366 = 538 \text{ LOC}$$

$$\text{Tamaño Total } T = 538 + 695 - 5 + 169 = 1397 \text{ LOC}$$

$$\text{Nuevo Reusable Total Estimado} = 49 \text{ LOC} *$$

**Nota: El tamaño modificado es ~~subtraído~~ así que no es incluido doblemente.**

# Ejercicio completado -3

Empezando con  $E = 366$ , use los parámetros de regresión del tiempo para calcular el tiempo de desarrollo.

$$\textit{Development\_Time} = \beta_{0\textit{time}} + \beta_{1\textit{time}} * (E)$$

$$\beta_{0\textit{Time}} = 108 \text{ y } \beta_{1\textit{Time}} = 2.95$$

$$\text{Tiempo de Desarrollo} = 108 + 2.95 * 366 = 1186 \text{ min.}$$

# Mensajes a Recordar

Estimaciones precisas del tamaño ayudarán a realizar mejores planes de desarrollo.

La habilidad de la estimación se mejora con la práctica.

Un proceso medido y definido provee unas bases repetibles para la mejora.

Para realizar estimaciones precisas, debe usar información histórica y seguir métodos convenientes.

El método PROBE muestra como hacerlo.



Universidad Carlos III De Madrid  
Principios de Ingeniería Informática  
María Isabel Sánchez Segura  
José Arturo Mora-Soto  
Juan Carlos Alonso Durán