

PSP 1
Estimación de tamaño
Reporte de pruebas

El proceso de Planeación III
Estimación del Tamaño del Software

15/01/2009

Proceso de Software Personal
Carlos Mojica

1 - III

¿Por qué estimar?

- *Para hacer mejores planes**
 - Para dimensionar de mejor manera el trabajo.
 - Para dividir el trabajo.
- *Para poder darle mejor seguimiento al progreso del trabajo.**
 - Medir mejor el trabajo.

15/01/2009

Proceso de Software Personal
Carlos Mojica

2 - III

Introducción

**¿En que momento podemos predecir –*estimar*–
con mayor exactitud?**

15/01/2009

Proceso de Software Personal
Carlos Mojica

3 - III

Antecedentes en la Estimación

- + Modelos de estimación en otras áreas**
 - Basados en largas bases de datos históricas.
 - Requieren estimaciones de tamaño como entradas.
- + Experiencias con la estimación del tamaño en el SW**
 - Los errores del 100% son normales.
 - Pocos desarrolladores realizan estimaciones.
 - Aún menos, utilizan métodos *formales*.

15/01/2009

Proceso de Software Personal
Carlos Mojica

4 - III

Principios de Estimación del Tamaño [1/4]

- + La estimación es un proceso incierto.
 - Nadie sabe que tan grande va a ser el producto.
 - Entre más temprano se haga la estimación, menos se sabe.
 - Las estimaciones pueden *truquearse* debido a presiones.
- + La estimación es un proceso intuitivo.
 - La habilidad se mejora con la experiencia y con datos.
 - Algunos son mejores que otros en la estimación.

15/01/2009

Proceso de Software Personal
Carlos Mojica

5 - III

Principios de Estimación del Tamaño [2/4]

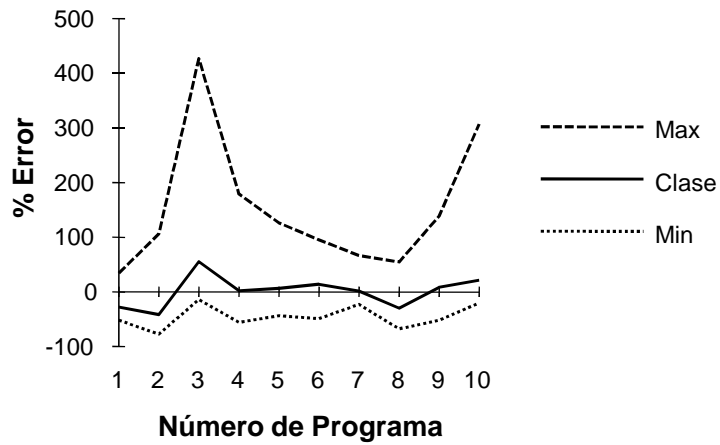
- + La estimación es una habilidad
 - La mejoría se obtiene gradualmente.
 - Quizá, nunca llegues a ser muy bueno.
- + El objetivo, sin embargo, es ser consistente
 - Sólo así, podrás entender las variaciones de tus estimaciones.
 - Tendrás que balancear tus estimaciones (entre las de arriba y las de abajo).

15/01/2009

Proceso de Software Personal
Carlos Mojica

6 - III

Rango de Error en la Estimación del Tamaño

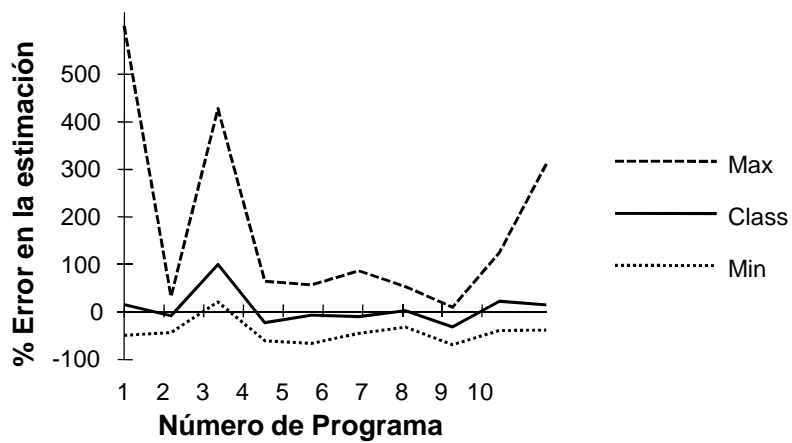


15/01/2009

Proceso de Software Personal
Carlos Mojica

7 - III

Precisión en la Estimación del Tiempo - % de Error



15/01/2009

Proceso de Software Personal
Carlos Mojica

8 - III

Principios de Estimación del Tamaño [3/4]

+ Las ventajas principales al usar un método de estimación definido son:

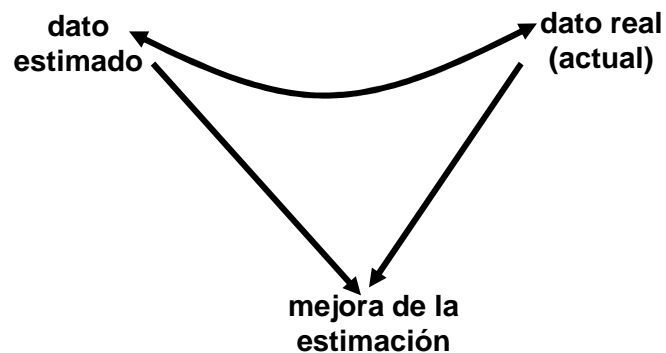
- Provee un marco de trabajo para la recolección de los datos de estimación.
- Al usar un método consistente y datos históricos, tus estimaciones serán más consistentes.

15/01/2009

Proceso de Software Personal
Carlos Mojica

9 - III

Principios de Estimación del Tamaño [4/4]

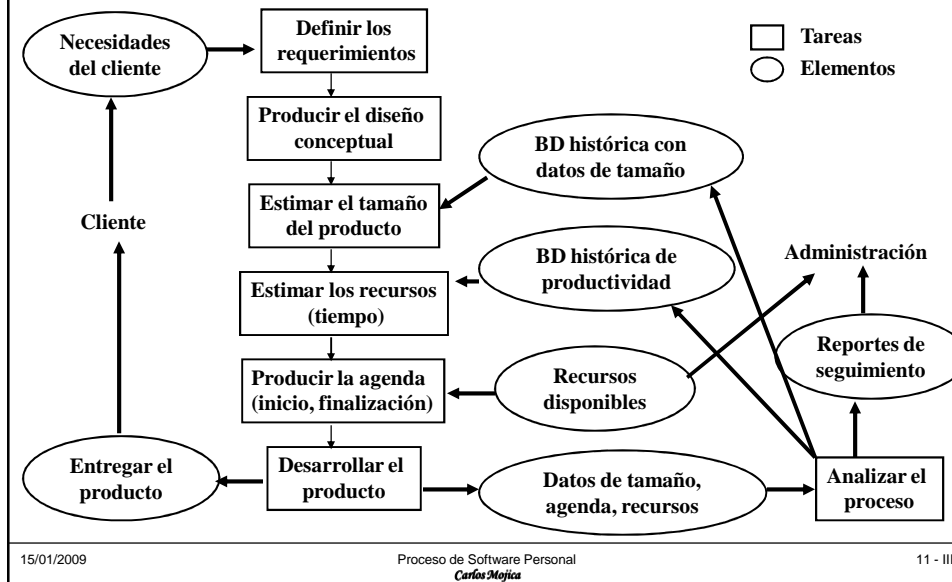


15/01/2009

Proceso de Software Personal
Carlos Mojica

10 - III

Marco de Trabajo para la Estimación del Tamaño



En resumen

La razón principal para estimar el tamaño del producto de SW es ayudarte a planear el desarrollo del producto.

La calidad del PDS, depende de las estimaciones de tamaño. La mala estimación es una de las razones principales de que los proyectos estén en problemas.

Con un buen plan:

- Se tienen las bases para presupuestar y armar el equipo de personas para el trabajo.
- Se sabe qué se tiene que hacer, cuándo y por quién.
- También se sabe cuánto tomará entender las dependencias críticas y otras restricciones.

Algunos métodos de estimación

Ya que el objetivo principal de la estimación del tamaño es tener estimaciones exactas para hacer mejores planes, debes experimentar con varios métodos de estimación y usar el que mejor se ajuste.

Entre algunos de estos métodos están:

- Delphi.**
- Lógica Difusa.**
- Puntos Funcionales.**
- Por componentes.**

15/01/2009

Proceso de Software Personal
Carlos Mojica

13 - III

Wideband-Delphi [1/5]

- + Varias personas estiman.**
 - Cada una hace una estimación en forma independiente.**
 - Cada una provee su estimación al coordinador.**
- + Coordinador**
 - Calcula el promedio de la estimación.**
 - Registra en la forma: el promedio y las otras estimaciones.**
- + Cuando se estabilizan las reestimaciones**
 - El promedio es la estimación final**

15/01/2009

Proceso de Software Personal
Carlos Mojica

14 - III

Wideband-Delphi [2/5]

Ejemplo

+ Se les pide a 3 personas realizar sus estimaciones sobre un producto.

+ Las estimaciones iniciales son:

- A = 13,800 LOC**
- B = 15,700 LOC**
- C = 21,000 LOC**

+ Luego, el coordinador

- Calcula el promedio, 16,833 LOC**
- Proporciona a cada persona este promedio y las otras estimaciones sin decir de quien es cual.**

15/01/2009

Proceso de Software Personal
Carlos Mojica

15 - III

Wideband-Delphi [3/5]

Ejemplo

+ Las personas que realizaron las estimaciones se reúnen y discuten las estimaciones.

+ Sus segundas estimaciones son:

- A = 18,500 LOC**
- B = 19,500 LOC**
- C = 20,000 LOC**

+ Luego, el coordinador

- Calcula el promedio, 19,333 LOC**
- Pregunta a los que realizaron las estimaciones si están de acuerdo con esta estimación**

15/01/2009

Proceso de Software Personal
Carlos Mojica

16 - III

Wideband-Delphi [4/5]

+ Ventajas

- Puede producir resultados muy precisos.
- Se usan habilidades de la organización.
- Trabaja para cualquier tamaño de producto.

+ Desventajas

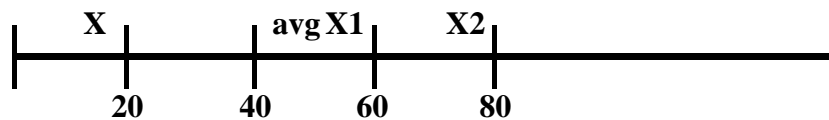
- Recae sobre pocos expertos.
- Consume tiempo.
- Está sujeta a percepciones personales.

15/01/2009

Proceso de Software Personal
Carlos Mojica

17 - III

Wideband-Delphi [5/5]



15/01/2009

Proceso de Software Personal
Carlos Mojica

18 - III

Lógica Difusa [1/10]

- + Recolectar datos históricos del tamaño de los programas desarrollados anteriormente.**
- + Subdividir estos datos en categorías de tamaño:**
 - Muy largos, Largos, Medianos, Pequeños y Muy pequeños.**
 - Establecer rangos de tamaño.**
- + Subdividir estos rangos en subcategorías.**

15/01/2009

Proceso de Software Personal
Carlos Mojica

19 - III

Lógica Difusa [2/10]

- + Asignar los datos disponibles a las categorías.**
- + Establecer subcategorías de rangos de tamaño.**
- + Cuando se estime un programa nuevo, juzgar a que categoría y subcategoría se acerca.**

15/01/2009

Proceso de Software Personal
Carlos Mojica

20 - III

Lógica Difusa [3/10] Ejemplo

+ Se tienen datos históricos de 5 programas:

- Una utilidad para manejo de archivos de 173 LOC.
- Un administrador de archivos de 5,834 LOC.
- Un programa de registro de datos de personal de 6,845 LOC.
- Un paquete generador de reportes de 8,386 LOC.
- Un programa administrador de inventarios de 10,341 LOC.

15/01/2009

Proceso de Software Personal
Carlos Mojica

21 - III

Lógica Difusa [4/10] Ejemplo

+ Se debe establecer 5 rangos de categorías:

- $\text{Log}(173) = 2.238$
- $\text{Log}(10,341) = 4.015$
- La diferencia es 1.777
- $1.777 / 4 = 0.444$
- Los rangos son espaciados 0.444
- Los límites de estos rangos están a 0.222 por arriba y por debajo de cada punto medio de cada rango ($0.444/2$).

2.016 - 2.238 – 2.460

2.460 - 2.682 – 2.904

2.904 - 3.126 – 3.348

3.348 - 3.570 – 3.792

3.792 - 4.015 – 4.236

15/01/2009

Proceso de Software Personal
Carlos Mojica

22 - III

Lógica Difusa [5/10] Ejemplo

Very Small	104	173	288
Small	288	481	802
Medium	802	1338	2230
Large	2230	3719	6202
Very Large	6202	10341	17243

$$\text{Log}_a n = b \Rightarrow a^b = n$$

15/01/2009

Proceso de Software Personal
Carlos Mojica

23 - III

Lógica Difusa [6/10] Ejemplo

+ El nuevo programa tiene los siguientes requerimientos:

- Analizar el rendimiento del mercado.
- Proyectar las ventas en cada categoría de producto.
- Asignar estas ventas a cada región de venta y los períodos de tiempo.
- Producir un reporte mensual de estas proyecciones y los resultados actuales.

15/01/2009

Proceso de Software Personal
Carlos Mojica

24 - III

Lógica Difusa [7/10]

Ejemplo

+ Al comparar el programa nuevo con los datos históricos se hacen los siguientes juicios:

- Es una aplicación más compleja que el administrador de archivos o el de personal.**
- No es tan compleja como el programa administrador de inventarios.**
- Parece tener más funcionalidades que el paquete de reportes.**

+ Se concluye que el programa nuevo está entre 6,202 y 10,341 LOC.

15/01/2009

Proceso de Software Personal
Carlos Mojica

25 - III

Lógica Difusa [8/10]

Ejemplo

+ Para realizar una estimación con lógica difusa:

- 1.- Dividir los datos históricos de tamaño de los productos en rangos de tamaño.**
- 2.- Comparar el producto planeado con los productos anteriores.**
- 3.- Con esta comparación, seleccionar el tamaño que más se aproxime al producto nuevo.**

15/01/2009

Proceso de Software Personal
Carlos Mojica

26 - III

Lógica Difusa [9/10] Ventajas

- + Está basado en datos históricos relevantes.**
- + Es fácil de usar.**
- + No requiere de herramientas o entrenamientos especiales.**
- + Provee estimaciones razonablemente buenas, cuando el trabajo es similar a algo anterior.**

15/01/2009

Proceso de Software Personal
Carlos Mojica

27 - III

Lógica Difusa [10/10] Desventajas

- + Requiere de muchos datos.**
- + Quien realice las estimaciones debe estar familiarizado con los programas que han sido desarrollados.**
- + Sólo provee una estimación cruda.**
- + No es útil para nuevos tipos de programas.**
- + No es útil para programas más largos o más pequeños que los datos históricos.**

15/01/2009

Proceso de Software Personal
Carlos Mojica

28 - III

Puntos Funcionales [1/9]

+ Un Punto Funcional es una unidad arbitraria

- Basada en la funcionalidad de la aplicación.
 - * Entradas, salidas, archivos, preguntas, interfases.

- Escalada por simple, promedio, compleja.

+ Si hay complejidad

- Ajustar $\pm 35\%$.

15/01/2009

Proceso de Software Personal
Carlos Mojica

29 - III

Puntos Funcionales [2/9]

+ Procedimiento

- Determinar el número de cada tipo de función en la aplicación.

- Juzgar la escala y complejidad de cada función.

- Calcular los puntos funcionales totales.

- Usar datos históricos en el desarrollo del costo por punto funcional para realizar las estimaciones.

- Multiplicar los puntos funcionales por el costo para obtener la estimación.

15/01/2009

Proceso de Software Personal
Carlos Mojica

30 - III

Puntos Funcionales [3/9]

Ejemplo

+ El nuevo programa tiene los siguientes requerimientos:

- Analizar el rendimiento del mercado.
- Proyectar las ventas en cada categoría de producto.
- Asignar estas ventas a cada región de venta y los períodos de tiempo.
- Producir un reporte mensual de estas proyecciones y los resultados actuales.

15/01/2009

Proceso de Software Personal
Carlos Mojica

31 - III

Puntos Funcionales [4/9]

Ejemplo

+ La primera estimación tiene los siguientes puntos funcionales:

- 12 entradas: $12 \times 4 = 48$
- 7 salidas: $7 \times 5 = 35$
- 0 preguntas: $0 \times 4 = 0$
- 3 archivos lógicos: $3 \times 10 = 30$
- 2 interfases: $2 \times 7 = 14$
- Puntos Funcionales antes del ajuste: 127

15/01/2009

Proceso de Software Personal
Carlos Mojica

32 - III

Puntos Funcionales [5/9] Ejemplo

+ Se ajusta siguiendo los factores de influencia:

- Comunicación de datos: 4
- Entrada de datos en línea: 4
- Complejidad del procesamiento: 3
- Facilidad operacional: 5
- Facilidad para el cambio: 5
- Total en factores de influencia: 21
- Complejidad = $0.65 + 21 \times 0.01 = 0.86$

+ El total de puntos funcionales = $127 \times 0.86 = 109.22$

+ Usando los datos históricos de horas/punto funcional, se puede calcular el tiempo de desarrollo del proyecto.

15/01/2009

Proceso de Software Personal
Carlos Mojica

33 - III

Puntos Funcionales [6/9] Ventajas

+ Pueden ser usadas en las etapas iniciales del proyecto.

+ Son independientes del lenguaje de programación, diseño del producto o estilo de desarrollo.

+ Existe mucha información histórica.

+ Es un método bien documentado.

+ Existe un grupo activo de usuarios de PF.

15/01/2009

Proceso de Software Personal
Carlos Mojica

34 - III

Puntos Funcionales [7/9] Desventajas

- + No se puede contar directamente los puntos funcionales del contenido de una aplicación.
- + Los puntos funcionales no reflejan el lenguaje de programación, diseño o diferencias de estilo.
- + Los puntos funcionales están diseñados para la estimación de aplicaciones de procesamiento de datos.

15/01/2009

Proceso de Software Personal
Carlos Mojica

35 - III

Puntos Funcionales [8/9] Cálculo

Número	Tipos de funciones	Peso	Total
	Entradas	X 4	
	Salidas	X 5	
	Peticiones	X 4	
	Archivos de Datos	X 10	
	Interfases	X 7	
Total de Puntos Funcionales no ajustados			

$$PF = \text{Complejidad} * PF \text{ no ajustados}$$

15/01/2009

Proceso de Software Personal
Carlos Mojica

36 - III

Puntos Funcionales [9/9] Cálculo

Factor de influencia	Peso
Comunicación de datos	2
Frecuencia de transacción	4
Actualización en línea	2
Facilidad de instalación	3
Facilidad de operación	4
Facilidad de cambios	3
Suma	41
Complejidad	$.65 + .01 * \sum \text{factores}$

15/01/2009

Proceso de Software Personal
Carlos Mojica

37 - III

Basada en Componentes [1/6]

- + Establecer los niveles principales del tamaño del producto.
 - Componentes, módulos, pantallas, etc.
 - Determinar el tamaño típico de cada nivel.
- + Para un nuevo producto
 - Determinar el nivel de componente en el cual la estimación sea práctica.
 - Estimar cuántos de estos componentes estarán en el producto.
 - Determinar el número máximo y mínimo posible.

15/01/2009

Proceso de Software Personal
Carlos Mojica

38 - III

Basada en Componentes [2/6]

- + Calcular el tamaño
 - Número de componentes de cada tipo.
 - Multiplicar por el tamaño típico de cada tipo.
 - Totalizar para obtener el tamaño.
- + Calcular el tamaño máximo y el mínimo de los componentes.
- + Calcular el tamaño como
 - $(\text{máximo} + 4 * (\text{lo probable}) + \text{mínimo}) / 6$

15/01/2009

Proceso de Software Personal
Carlos Mojica

39 - III

Basada en Componentes [3/6] Ejemplo

- + Se tienen los siguientes datos históricos de un número de componentes estándar
 - Entrada de datos: 1,108 LOC
 - Salida: 675 LOC
 - Archivo: 1,585
 - Control: 2,550 LOC
 - Cálculo: 475 LOC
- + Primero, estimar el número máximo, mínimo y más probable de componentes que se van a utilizar
 - De entrada = 1, 4, 7
 - De salida = 1, 3, 5
 - De archivo = 2, 4, 8
 - De control = 1, 2, 3
 - De cálculo = 1, 3, 7

15/01/2009

Proceso de Software Personal
Carlos Mojica

40 - III

Basada en Componentes [4/6] Ejemplo

+ Luego, calcular el tamaño máximo, mínimo y el más probable de los componentes

- De entrada = 1108, 4432, 7756
- De salida = 675, 2025, 3375
- De archivo = 3170, 6340, 12680
- De control = 2550, 5100, 7650
- De cálculo = 475, 1425, 3325

+ Luego, totalizar el máximo, mínimo y más probable

- Máximo = 34786
- Más probable = 19322
- Mínimo = 7978

15/01/2009

Proceso de Software Personal
Carlos Mojica

41 - III

Basada en Componentes [5/6] Ejemplo

+ El tamaño estimado es

- $LOC = (7978 + 4 * 19322 + 34786) / 6 = 18,679 \text{ LOC}$
- La desviación estándar es $(34786 - 7978) / 6 = 4468$
- El rango estimado es: 14,211 a 23,147 LOC

15/01/2009

Proceso de Software Personal
Carlos Mojica

42 - III

Basada en Componentes [6/6] Ventajas y Desventajas

+ Ventajas

- Basado en datos históricos relevantes.
- Fácil de usar.
- No requiere de herramientas o entrenamiento especial.
- Provee una estimación cruda.

+ Desventajas

- Se deben usar componentes grandes al principio del proyecto.
- Datos limitados de componentes grandes.

15/01/2009

Proceso de Software Personal
Carlos Mojica

43 - III

Basada en Proxies [1/10]

+ La regla básica

- Las buenas estimaciones son detalladas.
- Los novatos, raramente piensan en detalle.

+ Alternativas

- Esperar a estimar hasta que se tenga el detalle.
- Realizar la mejor conjetura.
- Identificar un proxy adecuado.

15/01/2009

Proceso de Software Personal
Carlos Mojica

44 - III

Basada en Proxies [2/10]

+ Un buen proxy debe:

- Estar relacionado al esfuerzo.
- Ser capaz de contarse en forma automática al igual que su contenido.
- Ser fácil de visualizar al principio del proyecto.
- Ser adecuado a las necesidades de la Organización.
- Ser sensitivo a las variaciones de implementación.

+ Algunos ejemplos de proxies

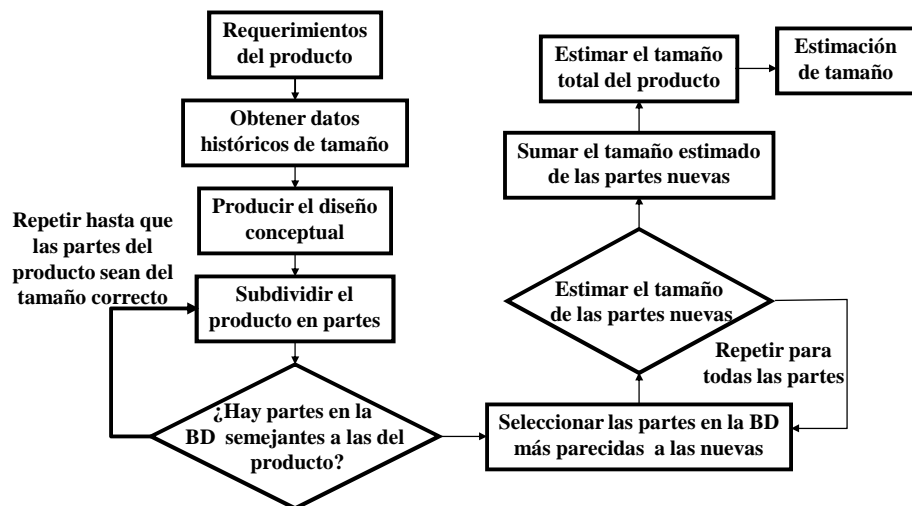
- Puntos Funcionales.
- Objetos.
- Elementos del producto
 - * Componentes.
 - * Pantallas, reportes, scripts, archivos.
 - * Capítulos de libros.

15/01/2009

Proceso de Software Personal
Carlos Mojica

45 - III

Basada en Proxies [3/10]



15/01/2009

Proceso de Software Personal
Carlos Mojica

46 - III

Basada en Proxies [4/10] Puntos Funcionales

- + Los datos muestran que los puntos funcionales se correlacionan bien con el tiempo de desarrollo.
- + Los puntos funcionales se pueden visualizar en los inicios del desarrollo.
- + Para usar propiamente los puntos funcionales, se requiere de personal entrenado.
- + Los puntos funcionales no pueden ser contados directamente.
- + El grupo de usuarios de puntos funcionales (IFPUG), está refinando el método de conteo.

15/01/2009

Proceso de Software Personal
Carlos Mojica

47 - III

Basada en Proxies [5/10] Componentes

- + La correlación entre los componentes y el desarrollo depende de los componentes.
- + Se requieren muchos datos.
- + El conteo de componentes es difícil de visualizar al inicio del desarrollo.
- + El conteo de componentes es fácil de automatizar.

15/01/2009

Proceso de Software Personal
Carlos Mojica

48 - III

Basada en Proxies [6/10] Objetos

- + Correlación con las horas de desarrollo**
 - El número de objetos se correlaciona razonablemente bien.
 - Las líneas de código (LOC), se aproximan a la correlación.
 - Las LOC del objeto pueden ser estimadas usando el método estándar de estimación de componentes.
 - Luego se pueden estimar las LOC de la relación histórica entre las LOC del objeto y las LOC del programa.

15/01/2009

Proceso de Software Personal
Carlos Mojica

49 - III

Basada en Proxies [7/10] Objetos

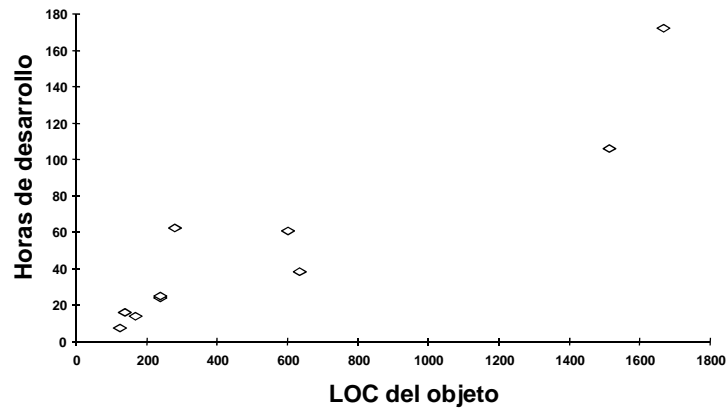
- + Cuando los objetos son seleccionados como entidades de la aplicación, pueden ser visualizados durante las primeras fases del desarrollo.**
- + Funciones y procedimientos pueden ser estimados de la misma manera.**
- + Objetos, funciones, procedimientos y sus líneas de código pueden ser contabilizadas automáticamente.**

15/01/2009

Proceso de Software Personal
Carlos Mojica

50 - III

Basada en Proxies [8/10] Correlación

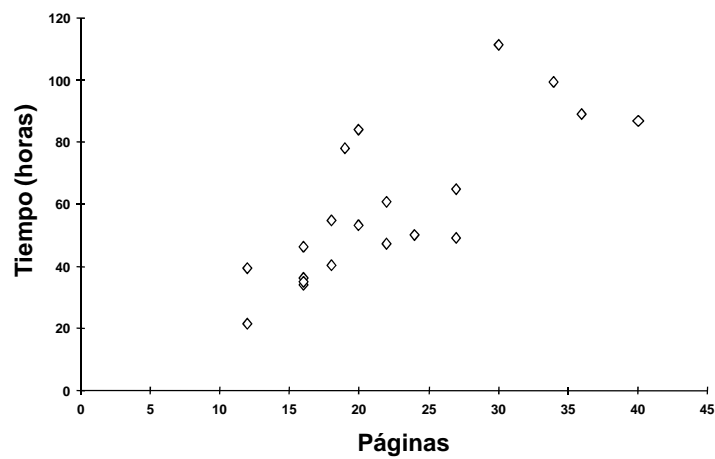


15/01/2009

Proceso de Software Personal
Carlos Mojica

51 - III

Basada en Proxies [9/10] Correlación



15/01/2009

Proceso de Software Personal
Carlos Mojica

52 - III

Basada en Proxies [10/10] Ejemplo de objetos para proxies

Objetos en Pascal y sus LOC					
Categoría	VS	S	M	L	VL
Control	4.24	8.68	17.79	36.46	74.71
Despliegue	4.72	6.35	8.55	11.50	15.46
Archivo	3.30	6.23	11.74	22.14	41.74
Lógico	6.41	12.42	24.06	46.60	90.27
Impresión	3.38	5.86	10.15	17.59	30.49
Texto	4.63	8.73	16.48	31.09	58.62

15/01/2009

Proceso de Software Personal
Carlos Mojica

53 - III

Reporte R3 Análisis de Defectos

Objetivos de la asignación:

- Entender la densidad y tipos de defectos introducidos y encontrados mientras se desarrollaron los primeros ejercicios.
- Demostrar la importancia de recolectar, registrar y reportar los datos del proceso.

15/01/2009

Proceso de Software Personal
Carlos Mojica

54 - III

Reporte R3 Análisis de Defectos

Tarea: Analizar los defectos encontrados en los ejercicios y producir un reporte que contenga:

1 Una tabla que muestre

- a) El número total de defectos encontrados.
- b) Las LOC nuevas y cambiadas.
- c) Los defectos por KLOC

2 Una tabla que muestre

- a) El número total de defectos encontrados en compilación.
- b) El número de defectos encontrados en pruebas.
- c) El número de defectos por KLOC encontrados en compilación
- d) El número de defectos por KLOC encontrados en pruebas.

3 Producir una tabla que muestre el tiempo promedio de corrección por defecto encontrado en compilación, en pruebas, inyectados en diseño, en codificación.

Usar 1A, 2A y 3A

15/01/2009

Proceso de Software Personal
Carlos Mojica

55 - III

Reporte R3 Análisis de Defectos

# de programa	LOC Nuevas y Cambiadas	Número de defectos	Defectos/KLOC
1A			
2A			
3A			
Totales			

# de programa	LOC Nuevas y Cambiadas	Defectos encontrados en Compilación	Defectos encontrados en Pruebas	Defectos en Compilación por KLOC	Defectos en Pruebas por KLOC
1A					
2A					
3A					
Totales					

15/01/2009

Proceso de Software Personal
Carlos Mojica

56 - III

Reporte R3 Análisis de Defectos

	Defectos encontrados en Compilación	Defectos encontrados en Pruebas	Total de Defectos encontrados
Defectos inyectados en Diseño	Para los defectos inyectados en diseño y encontrados en compilación: Tiempo total de corrección: Número total de defectos: Tiempo promedio de corrección:	Para los defectos inyectados en diseño y encontrados en pruebas: Tiempo total de corrección: Número total de defectos: Tiempo promedio de corrección:	Para todos los defectos inyectados en diseño: Tiempo total de corrección: Número total de defectos: Tiempo promedio de corrección:
Defectos inyectados en Codificación	Para los defectos inyectados en codificación y encontrados en compilación: Tiempo total de corrección: Número total de defectos: Tiempo promedio de corrección:	Para los defectos inyectados en codificación y encontrados en pruebas: Tiempo total de corrección: Número total de defectos: Tiempo promedio de corrección:	Para todos los defectos inyectados en codificación: Tiempo total de corrección: Número total de defectos: Tiempo promedio de corrección:
Total de Defectos inyectados	Para todos los defectos encontrados en compilación: Tiempo total de corrección: Número total de defectos: Tiempo promedio de corrección:	Para todos los defectos encontrados en pruebas: Tiempo total de corrección: Número total de defectos: Tiempo promedio de corrección:	Para todos los defectos: Tiempo total de corrección: Número total de defectos: Tiempo promedio de corrección:

15/01/2009

Proceso de Software Personal
Carlos Mojica

57 - III

Ejercicio 3A Usar PSP0.1

Escribir un programa para contar las LOC totales del programa, las LOC totales de cada objeto que contenga el programa y el número de métodos en cada objeto. Usar 1A, 2A y 3A. Se puede modificar o reutilizar el programa 2A.

# de programa	Nombre del Objeto	Número de métodos	LOC del Objeto	LOC totales del programa
1A	ABC	3	86	
	DEF	2	8	
	GHI	4	92	
				212
2A	...			

# de programa	Nombre de la func/proc	Número de métodos	LOC de la fun/proc	LOC totales del programa
1A	ABC	1	86	
	DEF	1	8	
	GHI	1	92	
				212
2A	...			

15/01/2009

Proceso de Software Personal
Carlos Mojica

58 - III

Orden de entrega de las formas

Resumen del Plan del Proyecto nivel PSP0.1.
FRPP proyecto anterior.
Formas de PIPs.
Forma de Registro de Tiempos.
Forma de Registro de Defectos.
Código Fuente del Programa.
Reporte R3.
Pantallas de la interfase gráfica.
Pantallas de los resultados.
Tabla de resultados.