Proceso Personal para el Desarrollo de Software

Medir Tamaño de Programa

Temario

- Medidas
- Medir Tamaño de Programa
  - Por qué medir tamaño de programa.
  - Criterios de medida
- Contar Tamaño de Programa
  - Estándares de codificación.
  - Estándares de conteo.
  - Contadores de tamaño.

Medición - Propósito

- Por qué medir?
  - Predecir o planificar para el futuro.
  - Comparar un producto o proceso
  - Determinar apego a los estándares.
  - Proporcionar una base para el control.
  - Entender y administrar el cambio.
- Se mide para conocer.
- Se mide como fuente de información.

Medición - Objetivos

- Las medidas en sí, son solamente números
- Para ser útiles, deben
  - Guardar relación con los objetivos de negocio.
  - Ser obtenidas con un propósito específico.
  - Estar explícitamente definidas.
  - Ser interpretas correctamente.
- Si las razones por las cuales se eligen determinadas medidas, no se comprenden
  - pueden ser recopilados datos incorrectos.
  - los datos pueden no ser utilizados correctamente.

Medición - Principios

- Medir un proceso no lo mejora.
- La mejora proviene de evaluar los valores obtenidos y actuar en consecuencia.
  - Tomar las acciones apropiadas.
- Actualizar el proceso, en función de las medidas, permite alcanzar la mejora continua.

Medición - Tipos

- Se necesitan medidas objetivas y explícitas.

- Debe existir una relación que las vincule.
  - Tamaño de programa versus horas de desarrollo.
  - Distribución del costo.
  - Densidad de defectos.

PSP - Medidas

- Medidas básicas de PSP
  - Tamaño de programa.
  - Tiempo dedicado por fase.
  - Defectos inyectados y defectos removidos, por fase.

- Se busca mantener el proceso controlado y predecible.
  - Mejorar calidad o reducir defectos.
  - Mejorar productividad.

PSP - Tamaño de Programa

- Objetivos al medir tamaño
  - Hacer mejores estimaciones de tamaño.
  - Establecer una base para normalizar medidas de tiempo y defectos.

- Algunas de las preguntas que estos datos pueden ayudar a contestar
  - ¿Qué tamaño tendrá el programa que se piensa desarrollar?.
  - ¿Qué tan buena fue la estimación de tamaño realizada?
  - ¿Cuál es el tamaño real del programa terminado ?

PSP - Medidas

- De las tres medidas básicas de PSP, se derivan otras, que son de utilidad para la mejora del proceso.
  - To Date %
  - Densidad de Defectos.
  - Proporción de Desarrollo de Tiempo.
  - Productividad
  - CPI (Cost Performance Index)
  - % Reuse y % New Reusable.

PSP – Medidas Derivadas

- To Date %
  - Distribución del tiempo o defectos por fase.
  - Proporciona la información para proyectar la distribución de tiempo de un nuevo proyecto, basado en la distribución de tiempo de proyectos anteriores

- Densidad de Defectos.
  - Mide la cantidad de defectos cada 1000 LOC (KLOC)

PSP – Medidas Derivadas

- Proporción de Desarrollo de Tiempo.
  - Proporción entre dos fases del proceso.
  - Tiempo de Diseño vs Tiempo de Codificación.

- Productividad
  - Tamaño de producto desarrollado por hora.

PSP – Medidas Derivadas

- CPI (Cost Performance Index).
  - Indica el grado en el cual se está cumpliendo con los costos comprometidos
  - Cociente Tiempo planificado a la fecha / Tiempo real a la fecha.
- % Reuse y % New Reusable.
  - Comprender el grado de reutilización del código previamente desarrollado.
  - Conocer la tasa de incorporación de nuevo código reutilizable.

PSP – Unidades de Medida

- Tiempo – Minutos
- Defectos – Cantidad (Enteros)
- Tamaño ??

- Al elegir una medida de tamaño, esta debe ser
  - Útil para la planificación.
  - Precisa.
  - Medible en forma automatizada.

PSP – Medida de Tamaño

• Útil para Planificación

- Las medidas intuitivas de tamaño son generalmente necesarias para los planes iniciales.
  - Para una casa, los pies cuadrados predicen costo.
  - Pocas personas pueden visualizar una casa en términos de pies cuadrados de espacio.
  - El número de cuartos es más intuitivos o los metros cuadrados.

PSP – Medida de Tamaño

Útil para Planificación

- Para ser útil para planificación
  - La medida de tamaño del producto debe guardar correlación con el tiempo dedicado a desarrollarlo.
- Si la medida de tamaño no se relaciona directamente con el esfuerzo de desarrollo, no es digna de utilizar.

PSP – Medida de Tamaño

Útil para Planificación

- Hay muchas medidas posibles.
  - Elementos de la base de datos.
  - Líneas del código físicas o lógicas (LOC).
  - Puntos funcionales.
  - Páginas, Pantallas, Reportes

PSP – Medida de Tamaño
Útil para Planificación

- LOC Lógico: cuenta elementos del lenguaje
  - Invariantes a los cambios de edición.
  - Están correlacionadas con el esfuerzo de desarrollo.
  - Complejas para contar.
    - Ej: If … then … else …
    - Ej: If …

      Then …

      Else …
- LOC Físico: cuenta líneas de texto.
  - Fáciles de contar.
  - Son variantes a los cambios de edición

PSP – Medida de Tamaño

- Para la medición de tamaño, debemos ser precisos en la definición de la medida.

- Si se usa LOC
  - Se cuenta LOC lógico o físico?
  - Se cuentan las líneas en blanco?
  - Se cuentan los comentarios?
  - Cómo se cuentan los constructores del lenguaje?

PSP – Medida de Tamaño

Medición automatizada.

- La medición manual de tamaño es muy costosa en tiempo, tediosa e inexacta.
  - Contadores automatizados.

- Los contadores pueden ser complejos, dependiendo de
  - La definición de tamaño seleccionada.
  - El método de conteo utilizado.

PSP – Medida de Tamaño

- PSP sugiere el uso de LOC Lógico (elementos del lenguaje) como medida de tamaño de software.
  - Uso extendido.
  - Se demostró su correlación con el esfuerzo de desarrollo.
  - Permite medir en función de la lógica y no de la cantidad de texto.

PSP – Medida de Tamaño

- Cómo contar LOC Lógico con PSP?
  - Definir un estándar de codificación y respetarlo.
  - Escribir el código fuente utilizando una línea física por cada constructor lógico del lenguaje utilizado.
  - Entonces contar líneas físicas equivale a contar líneas lógicas.

PSP – Medida de Tamaño

- Contar todas las sentencias o constructores, lógicos.
  - if, then, else, for, etc.
  - Declaraciones, Directivas, cabezales, etc.
- No contar líneas en blanco
- No contar comentarios
- No contar código generado en forma automática.

PSP – Framework de Medición (SEI)

Example C++ Size Counting Standard

Definition Name:	Example C++ LOC std.	Language:	C++
Author	W.S. Humphrey	Date	12/20/93

Count Type Physical/Logical	**Type** Logical	**Comments**
Statement Type	**Included**	**Comments**
Executable	Yes	
Nonexecutable:		
Declarations	Yes,Notes 3, 4	
Compiler Directives	Yes, Note 4	
Comments	No	
Blank lines	No	
Other elements		

PSP – Framework de Medición (SEI)

Clarifications		Examples/Cases
Empty statements	yes	";", lone {'s, etc.
Begin...end	note 1	
Expression evaluation	yes	when used as sub program arguments
End symbols	notes 1,2	for terminating executable statements, declarations, bodies
Then, else, otherwise	note 1	
Elseif	yes	
Keywords	yes	procedure division, interface, implementation
Labels	yes	branch destinations when on separate lines
Note 1		Count once every occurrence of the following key words: CASE, DO, ELSE, ENUM, FOR, IF, PRIVATE, PUBLIC, STRUCT, SWITCH, UNION, WHILE
Note 2		count once every occurrence of the following: ; , {} or }
Note 3		count each variable or parameter declaration
Note 4		count once each //define, //ifdef, //include, etc. statement

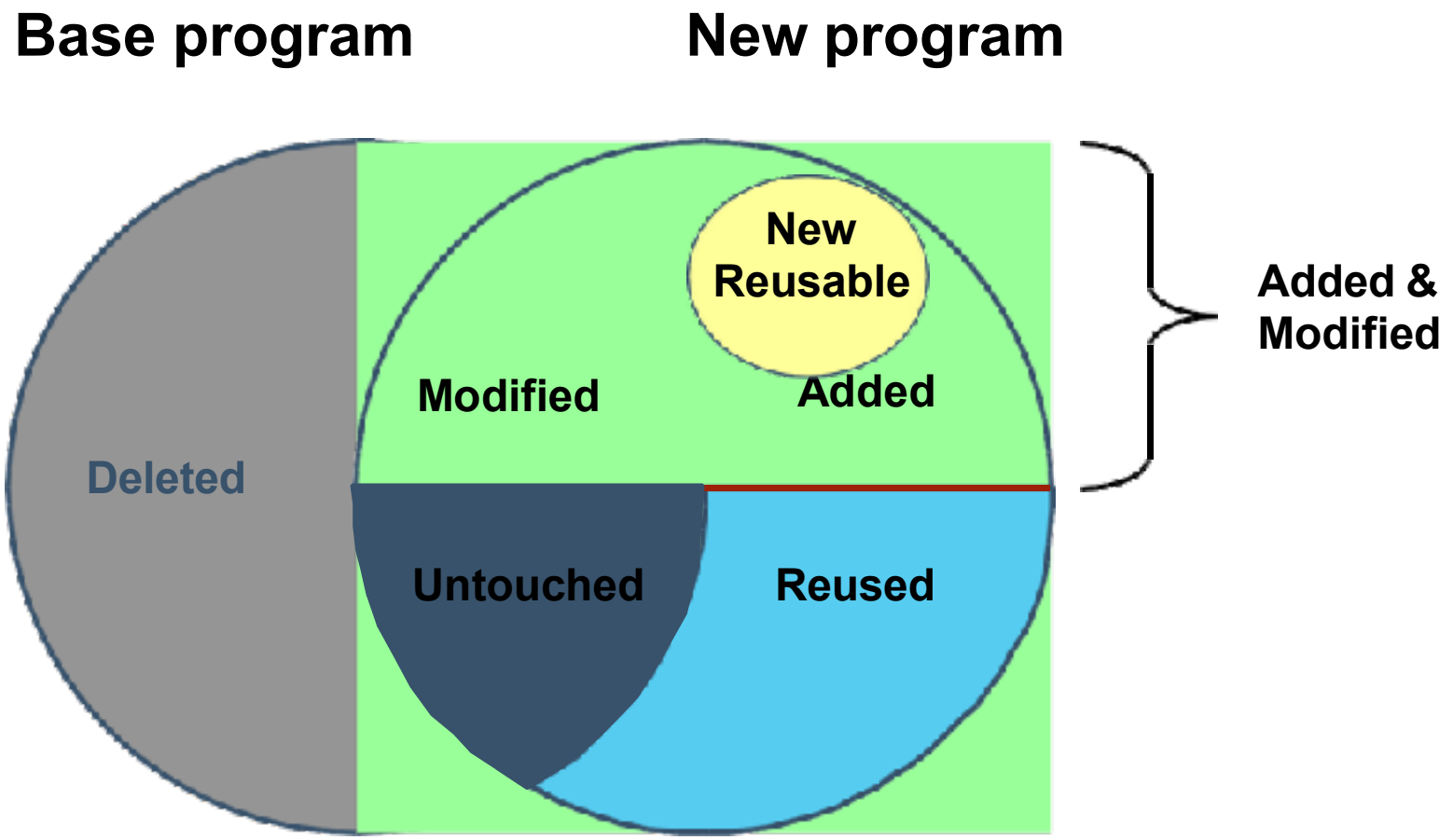
PSP – Conteo de Tamaño en LOC

- Categorías de LOC
  - **Base(B):** Cuando un producto existente será modificado o actualizado, LOC Base es el tamaño de la versión original del producto.
  - **Added (A):** Código escrito para un programa nuevo o agregado a un programa pre-existente.
  - **Modified (M):** Líneas modificadas del LOC Base de un programa.
  - **Deleted (D):** Líneas eliminadas del LOC Base de un programa.

PSP – Conteo de Tamaño en LOC

- **Added and Modified:** A+M. Es el valor que se utiliza para las estimaciones de tamaño.
- **Reuse(R):** Código que se toma de librerías y se utiliza sin modificación alguna.
- **New Reuse:** Código que se crea durante el desarrollo del producto actual, pero pensado para formar parte de una librería.
- **Total(T):** Es el tamaño total de programa en LOC.

PSP – Conteo de Tamaño en LOC



PSP – Conteo de Tamaño en LOC

- El tamaño total de programa incluye código agregado, modificado, borrado, base y reutilizado.
- Cuando se modifica un programa existente, el código base es el tamaño de dicho programa sin modificaciones.
  - Se cuenta como código base, no como código reutilizado.
- Código base es una forma de reutilización, pero PSP solo cuenta como reutilización el código no modificado utilizado desde librerías.

PSP – Conteo de Tamaño en LOC

- Cuando se modifican programas resulta necesario realizar un seguimiento de su tamaño.

- Se necesita un método para identificar la cantidad de:
  - Líneas Agregadas
  - Líneas Modificadas
  - Líneas Eliminadas

PSP – Ejemplos de Conteo

Se realizó una nueva versión de un producto de 100.000 LOC

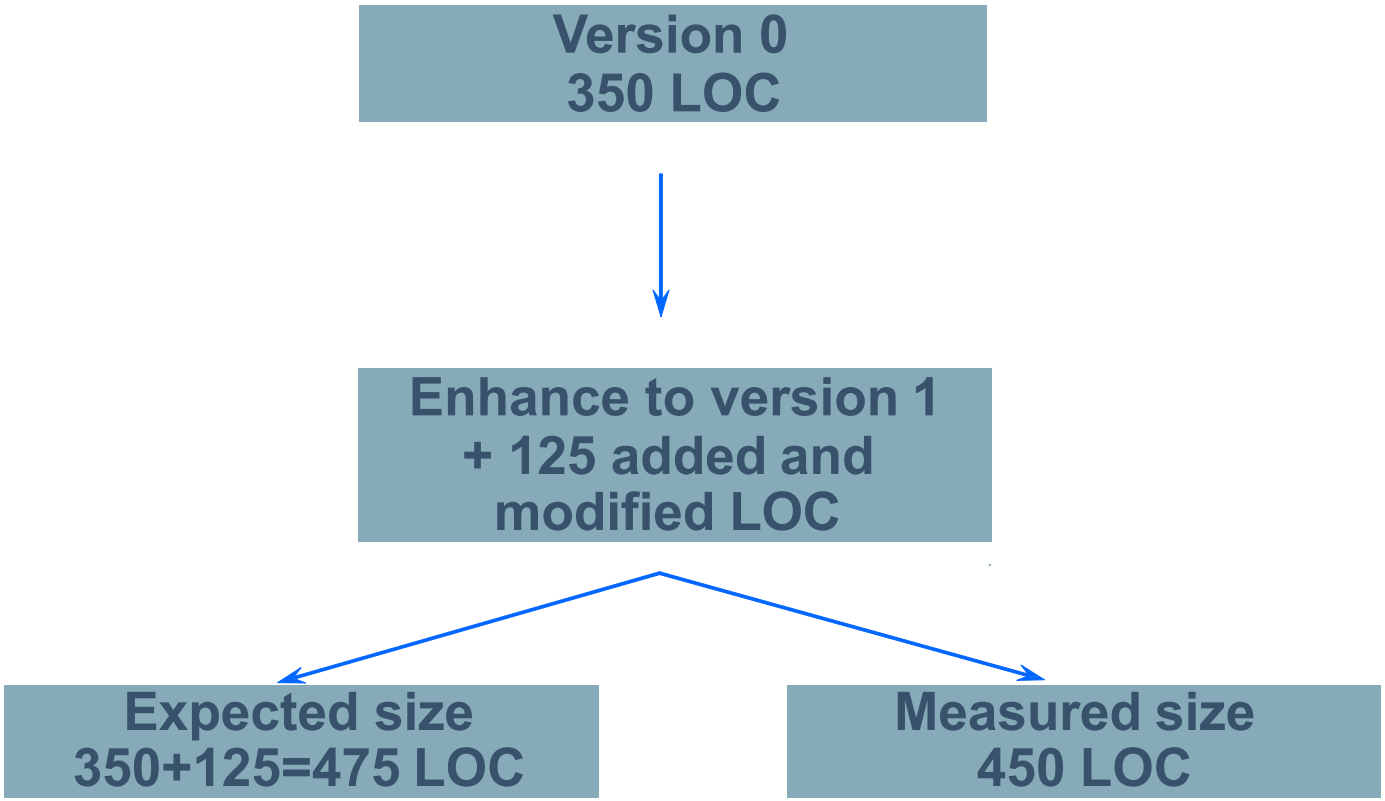
- LOC Base = 100.000 LOC
- LOC Deleted = 12.000 LOC
- LOC Added = 23.000 LOC
- LOC Modified = 5.000 LOC
- LOC Reused = 3.000 LOC

Resultados:

LOC Total = Base – Deleted + Added + Reused =
100.000 - 12.000 + 23.000 + 3.000 = 114.000 LOC

LOC A&M = 23.000 + 5.000 = 28.000

PSP – Conteo de Tamaño en LOC



Qué pasó?

Resumen

- Hay muchas razones por las cuales medir tamaño. La más importante es para poder PLANIFICAR.
- La medida de tamaño elegida, debe correlacionarse con el esfuerzo de desarrollo requerido.
- Debe poder definirse en forma precisa y ser contable automáticamente.
- Si la medida de tamaño no se correlaciona, ni es contable automáticamente, no será muy útil.
- La clave para realizar mediciones efectivas de tamaño es asegurarse que se ajustan a sus necesidades personales.

Referencias

- *“A Discipline for Software Engineering”,* Humphrey, W. MA: Addison-*Wesley, 1995*
- “The Personal Software Process”, Humphrey, W. *Technical Report, 2000*