Programa 2

Proceso Personal para el Desarrollo de Software

Este material fue realizado en base a material del curso “Personal Software Process for Engineers: PartI”, dictado por The Software Engineering Institute (SEI)

Proceso Personal para el Desarrollo de Software

Instructivo para el Programa 2

Descripción

Descripción

El presente instructivo cubre los siguientes temas

Programa 2 Requerimientos

Requerimientos Programa 2

Usando PSP0.1, escribir un programa que cuente en líneas de código (LOC)
- Tamaño total programa
- Tamaño total de cada clase
- Número de métodos de cada clase

La salida del programa será
- Tamaño total del programa
- Tamaño, cantidad de métodos para cada clase, junto con el nombre de la clase.

Utilice los estándares de codificación y conteo vistos en clase

Probar a fondo el programa. Como mínimo, probar el programa contando los tamaños totales de los programas 1 y 2 y sus clases.

Número de Programa	Nombre de Clase	Numero de Items	Tamaño de Clase	Tamaño Total
1	ABC	3	86	
	DEF	2	8	
	GHI	4	92	
				212
2	…			

Tabla 1

Sugerencias Programa 2

Sugerencias

No intentar escribir un programa de cuenta sofisticado.

Para el conteo de LOC, seguir la estrategia de cuenta sugerida en clase.

Si las clases, funciones, o todo lo demás es difícil de identificar en el lenguaje de programación utilizado, considere incluir comentarios especiales para identificar tales cosas en el programa de cuenta.
Recordar modificar los estándares de codificación y conteo para incluir estos comentarios. También se tendrá que incluir tales comentarios en cada programa, incluyendo el programa 1.

Instrucciones

Instrucciones

Antes de comenzar el programa 2, repasar el proceso PSP0.1 para asegurarse de comprenderlo. También, asegurarse de tener todas las entradas requeridas antes de comenzar con la fase de planificación.

PSP0.1 Process Script

Purpose	To guide the development of module-level programs
Entry Criteria	- Problem description - PSP0.1 Project Plan Summary form - Time and Defect Recording logs - Defect Type, ***Coding, and Size Counting*** standards - Stopwatch (optional)

Step	Activities	Description
1	Planning	- Produce or obtain a requirements statement. - ***Estimate the added and modified size of this program.*** - Estimate the required development time. - Enter the plan data in the Project Plan Summary form. - Complete the Time Recording log.
2	Development	- Design the program. - Implement the design. - Compile the program, and fix and log all defects found. - Test the program, and fix and log all defects found. - Complete the Time Recording log.
3	Postmortem	Complete the Project Plan Summary form with actual time, defect, and size data.

Exit Criteria	- A thoroughly tested program - Completed Project Plan Summary form with estimated and actual data - ***Completed PIP forms*** - Completed Time and Defect Recording logs

Fase de Planificación Planificar el desarrollo del programa 2 siguiendo el script para la fase de planificación

PSP0.1 Planning Script

Purpose	To guide the PSP planning process
Entry Criteria	- Problem description - PSP0.1 Project Plan Summary form - Time Recording log

Step	Activities	Description
1	Program Requirements	- Produce or obtain a requirements statement for the program. - Ensure that the requirements statement is clear and unambiguous. - Resolve any questions.
2	***Size Estimate***	- ***Make your best estimate of the added and modified size of this program.*** - ***Enter the plan size data in the Project Plan Summary form.***
3	Resource Estimate	- Make your best estimate of the time required to develop this program. - Enter the plan time data in the Project Plan Summary form. - ***Using the To Date % from the most recently developed program as a guide, distribute the development time over the planned project phases.***

Exit Criteria	- Documented requirements statement - Completed Project Plan Summary form with estimated ***program size and*** development time data - Completed Time Recording log

Verifique que ha cumplido con las condiciones de salida de la fase de planificación.

Fase de Desarrollo Desarrollar el programa 2 siguiendo el script para la fase de desarrollo

PSP0.1 Development Script

Purpose	To guide the development of small programs
Entry Criteria	- Requirements statement - Project Plan Summary form with estimated program ***size and*** development time - Time and Defect Recording logs - Defect Type standard ***and Coding standard***

Step	**Activities**	**Description**
1	Design	- Review the requirements and produce a design to meet them. - Record in the Defect Recording log any requirements defects found. - Record time in the Time Recording log.
2	Code	- Implement the design ***following the Coding standard.*** - Record in the Defect Recording log any requirements or design defects found. - Record time in the Time Recording log.
3	Compile	- Compile the program until there are no compile errors. - Fix all defects found. - Record defects in the Defect Recording log. - Record time in the Time Recording log.
4	Test	- Test until all tests run without error. - Fix all defects found. - Record defects in the Defect Recording log. - Record time in the Time Recording log.

Exit Criteria	- A thoroughly tested program ***that conforms to the Coding standard*** - Completed Time and Defect Recording logs

Verifique que ha cumplido con todas las condiciones de salida de la fase de desarrollo. Luego pase a la fase de Postmortem.

Fase de Postmortem Realice la fase de Postmorten siguiendo el script indicado para ello.

PSP0.1 Postmortem Script

Purpose	To guide the PSP postmortem process
Entry Criteria	- Problem description and requirements statement - Project Plan Summary form with program size and development time data - Completed Time and Defect Recording logs - A tested and running program ***that conforms to the coding and size counting standards***

Step	Activities	Description
1	Defect Recording	- Review the Project Plan Summary to verify that all of the defects found in each phase were recorded. - Using your best recollection, record any omitted defects.
2	Defect Data Consistency	- Check that the data on every defect in the Defect Recording log are accurate and complete. - Verify that the numbers of defects injected and removed per phase are reasonable and correct. - Using your best recollection, correct any missing or incorrect defect data.
3	***Size***	- ***Count the size of the completed program.*** - ***Determine the size of the base, deleted, modified, reused, total, and new reusable code. (Note: The size of added and modified code is calculated by the SEI student workbook.)*** - ***Enter these data in the Project Plan Summary form.***
4	Time	- Review the completed Time Recording log for errors or omissions. - Using your best recollection, correct any missing or incomplete time data.

Exit Criteria	- A thoroughly tested program ***that conforms to the coding and size counting standards*** - Completed Project Plan Summary form - ***Completed PIP forms describing process problems, improvement suggestions, and lessons learned*** - Completed Time and Defect Recording logs

Verifique que ha cumplido con todas las condiciones de salida de la fase de Postmortem.

Entregables

Cuando complete la etapa de postmortem, arme un archivo .zip para enviar al instructor, conteniendo lo siguiente.

- El archivo .mdb con sus datos del ejercicio.
- Código fuente del programa.
- Impresión de pantalla de las pruebas realizadas mostrando el resultado de la mismas.
- Captura de pantalla de la salida del contador de LOC aplicada al ejercicio 2.
- Archivo de texto o documento que contenga la descripción de qué clases/módulos/unidades del código construído contienen las distintas categorías (Added, Modified, Delete, etc). Utilizando el contador de LOC que muestra el tamaño de dichas unidades de código.

Ejemplo:

36 LOC Base -> Clase Base.java y muestre el tamaño de la clase utilizando el contador de LOCS aplicado a la clase.

3 LOC Deleted -> En la clase Base.java

10 LOC Modified -> En la clase Base.java

15 LOC Added -> En la clase Base.java

40 LOC Reused -> Clases LibUno.java, LibDos.java y muestre el tamaño de la las mismas utilizando el contador de LOCS aplicado ambas clases.

5 LOC New Reused -> agregadas en NuevaLib.java

Tenga en cuenta que:

.- La lectura y preparación de los estándares de codificación y conteo no entra dentro del tiempo de planificación del ejercicio. Deben estar prontos antes de comenzar el ejercicio.

.- El ejercicio no pide construir un parser, sino un contador de LOC que se basa en la definición de estándares de codificación y conteo que equipara líneas lógicas a líneas físicas. De manera que contar una línea física implique contar una línea lógica.

.- Para quienes programan en C++ recuerden que:
– se deben contar tanto las LOC en los .h como en los .cpp.
– los TAD son "clases" a los efectos del conteo. Como no hay palabra clave que los identifique se pueden crear una.

.- Si en la programación del ejercicio se utiliza otro lenguaje donde ocurre algo similar que en C++, vale la misma aclaración anterior.

Criterios de evaluación para el programa 2

Criterios de Evaluación	Su reporte debe ser • completo	Los datos de su proceso deben ser • exactos • precisos • consistentes
Sugerencias	Recuerde entregar su reporte en fecha. Mantenga la simplicidad del programa. Si no esta seguro de algo, consulte a su instructor Debe producir y reportar sus propias estimaciones y datos reales, desarrollar su propio código y llevar adelante su propio juego de pruebas.	

