



Tecnológico de Monterrey

Campus Monterrey

Escuela de Ingeniería y Tecnologías de Información
Departamento de Ciencias Computacionales

Scripts del PSPSM

Clase TC-3045 “Calidad y Pruebas de Software”

SM “Personal Software Process” y “PSP” son marcas de servicio de la Universidad de Carnegie Mellon

Índice:

Script del Proceso PSP0:.....	3
Script del Proceso PSP1:.....	5
Script del Proceso PSP2:.....	8
Script del Proceso PSP2.1:.....	12

Script del Proceso PSP0¹:

Propósito	Guiar el proceso de desarrollo de un programa computacional, utilizando el Process Dashboard como herramienta de captura de métricas personales.	
Criterios de Entrada	<ul style="list-style-type: none"> - Descripción del Problema - Process Dashboard con las fases del PSP0 asignadas al programa a desarrollar - Forma de Reporte de Casos de Prueba - Estándar de Tipo de Defectos 	
#	Paso	Descripción
1	<u>Planeación</u> ² (<i>Planning</i>) ³	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Obtener la descripción de los requerimientos - Estudiar y entender los requerimientos - Estimar el tiempo que tomará el desarrollo del programa y capturarlo en la forma Resumen del Plan (<i>Project Plan Summary</i>) - Detener el reloj y marcar la fase como terminada
2	Diseño (<i>Design</i>)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Utilizando como base los requerimientos: <ul style="list-style-type: none"> ▪ Diseñar en forma completa el programa ▪ Diseñar todos los casos de pruebas unitarias - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
3	Codificación (<i>Code</i>)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Escribir el código de acuerdo al diseño realizado - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
4	Compilación (<i>Compile</i>)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Compilar el código escrito (repetir hasta que el compilador no marque errores) - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
5	Pruebas Unitarias (<i>Test</i>)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Ejecutar todos los casos de prueba diseñados (repetir hasta que ya no se encuentren errores) - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
6	<u>Postmortem</u> (<i>Postmortem</i>)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Revisar que se hayan registrado en forma completa y correcta todos los tiempos y los defectos - Detener el reloj y marcar la fase y el programa como terminados
Criterios de Salida	<ul style="list-style-type: none"> - Programa pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos - Resumen del Plan (<i>Project Plan Summary</i>) lleno con los datos estimados y reales - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas - Reporte de pruebas completo 	

¹ Traducido y adaptado del Proceso PSP0 que se encuentra descrito en la página 18 del libro PSP A Self-Improvement Process for Software Engineers del autor Watts S. Humphrey de la editorial Addison Wesley.

² Los pasos que se encuentran subrayados tienen un sub-script que los detalla.

³ Entre paréntesis y en *italica* se muestra el nombre en inglés utilizado por Process Dashboard.

Script PSP0 para la Planeación:

Propósito		Guiar el proceso de planeación en PSP
Criterios de Entrada		- Descripción del Problema - Process Dashboard con las fases del PSP0 asignadas al programa a desarrollar
#	Paso	Descripción
1	Requerimientos del Programa	- Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Obtener o producir el documento de los requerimientos del programa a desarrollar - Asegurar que los requerimientos son claros y no ambiguos - Obtener respuesta a todas las dudas
2	Estimado de Tiempo	- Hacer el mejor estimado del tiempo que tomará desarrollar el programa - Capturar el tiempo estimado en la forma Resumen del Plan (<i>Project Plan Summary</i>) - Detener el reloj y marcar la fase de planeación como terminada
Criterios de Salida		- Documento de requerimientos completo y entendido - Forma Resumen del Plan (<i>Project Plan Summary</i>) con los datos del tiempo estimado - Bitácora de tiempo (<i>Time Log</i>) completa para esta fase

Script PSP0 para el Postmortem:

Propósito		Guiar el proceso de postmortem en PSP
Criterios de Entrada		- Documento de requerimientos completo y entendido - Programa pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos - Process Dashboard con todos los datos recolectados al estar realizando el programa
#	Paso	Descripción
1	Registro de Defectos	- Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Revisar el Resumen del Plan (<i>Project Plan Summary</i>) para verificar que todos los defectos encontrados en cada fase están registrados - Utilizando el mejor recuerdo, registrar en la bitácora de defectos (<i>Defect Log</i>) aquellos defectos que se hayan olvidado registrar
2	Consistencia de los datos de los defectos	- Verificar que en la bitácora de defectos (<i>Defect Log</i>) los datos de cada defecto estén correctos y completos - Verificar que en el Resumen del Plan (<i>Project Plan Summary</i>) la cantidad de defectos inyectados y removidos por fase sean razonables y completos - Utilizando el mejor recuerdo, corregir cualquier dato que falte o esté incorrecto
3	Registro de tiempo	- Verificar la bitácora de tiempo (<i>Time Log</i>) buscando errores u omisiones - Utilizando el mejor recuerdo, corregir cualquier dato de tiempo que falte o esté incorrecto - Detener el reloj y marcar la fase de postmortem como terminada - Marcar el programa como terminado
Criterios de Salida		- Programa pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos - Resumen del Plan (<i>Project Plan Summary</i>) lleno con los datos estimados y reales - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas - Reporte de pruebas completo

Script del Proceso PSP1⁴:

Propósito	Guiar el proceso de desarrollo de un programa computacional, utilizando el Process Dashboard como herramienta de captura de métricas personales.	
Criterios de Entrada	<ul style="list-style-type: none"> - Descripción del Problema - Process Dashboard con las fases del PSP1 asignadas al programa a desarrollar - Forma de Reporte de Casos de Prueba - Estándar de Tipo de Defectos - Estándares de Codificación y Contabilización - Datos históricos de tiempo y tamaño 	
#	Paso	Descripción
1	<u>Planeación</u> (Planning)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Obtener la descripción de los requerimientos - Estudiar y entender los requerimientos - Utilizar el método PROBE para estimar las LDC agregadas y modificadas - Utilizar el método PROBE para estimar el tiempo que tomará el desarrollo del programa - Capturar los estimados en la Plantilla de Estimación de Tamaño (Size Estimating Template) - Verificar en la forma Resumen del Plan (Project Plan Summary) que la información planeada es correcta - Detener el reloj y marcar la fase como terminada
2	Diseño (Design)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Utilizando como base los requerimientos: <ul style="list-style-type: none"> ▪ Diseñar en forma completa el programa ▪ Diseñar todos los casos de pruebas unitarias - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (Defect Log) - Detener el reloj y marcar la fase como terminada
3	Codificación (Code)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Escribir el código de acuerdo al diseño y siguiendo los estándares de codificación y contabilización - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (Defect Log) - Detener el reloj y marcar la fase como terminada
4	Compilación (Compile)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Compilar el código escrito (repetir hasta que el compilador no marque errores) - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (Defect Log) - Detener el reloj y marcar la fase como terminada
5	Pruebas Unitarias (Test)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Ejecutar todos los casos de prueba diseñados (repetir hasta que ya no se encuentren errores) - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (Defect Log) - Detener el reloj y marcar la fase como terminada
6	<u>Postmortem</u> (Postmortem)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Capturar en la Plantilla de Estimación de Tamaño (Size Estimating Template) y en el Resumen del Plan (Project Plan Summary) el tamaño real del programa - Revisar que se hayan registrado en forma completa y correcta todos los tiempos y defectos - Elaborar una Propuesta de Mejora de Proceso (Process Improvement Proposal) - Detener el reloj y marcar la fase y el programa como terminados
Criterios de Salida	<ul style="list-style-type: none"> - Programa que pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos y con los estándares de codificación y contabilización - Resumen del Plan (Project Plan Summary) lleno con los datos estimados y reales - Plantilla de Estimación de Tamaño (Size Estimating Template) completa - Bitácora de tiempo (Time Log) y de defectos (Defect Log) completas - Reporte de pruebas completo - Formato de Propuesta de Mejora de Proceso (Process Improvement Proposal) completo 	

⁴ Traducido y adaptado del Proceso PSP 1.0 del material académico proporcionado por el SEI.

Script PSP1 para la Planeación:

Propósito		Guiar el proceso de planeación en PSP
Criterios de Entrada		<ul style="list-style-type: none"> - Descripción del Problema - Process Dashboard con las fases del PSP1 asignadas al programa a desarrollar - Datos históricos de tiempo y tamaño
#	Paso	Descripción
1	Requerimientos del Programa	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Obtener o producir el documento de los requerimientos del programa a desarrollar - Asegurar que los requerimientos son claros y no ambiguos - Obtener respuesta a todas las dudas
2	Estimado de Tamaño	<ul style="list-style-type: none"> - Realizar un diseño conceptual del programa a desarrollar - Capturar los estimados de LDC de todas las partes en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Utilizar el método PROBE para estimar las LDC agregadas y modificadas para este programa (en Process Dashboard utilizar el "<i>PROBE Wizard</i>")
3	Estimado de Tiempo	<ul style="list-style-type: none"> - Utilizar el método PROBE para estimar el tiempo que tomará desarrollar el programa - Capturar el tiempo estimado en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Verificar en la forma Resumen del Plan (<i>Project Plan Summary</i>) que la información planeada es correcta - Detener el reloj y marcar la fase de planeación como terminada
Criterios de Salida		<ul style="list-style-type: none"> - Documento de requerimientos completo y entendido - Diseño conceptual del programa - Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) completa - Forma Resumen del Plan (<i>Project Plan Summary</i>) con estimados de tiempo y tamaño completos - Bitácora de tiempo (<i>Time Log</i>) completa para esta fase

Script PSP1 para el Postmortem:

Propósito		Guiar el proceso de postmortem en PSP
Criterios de Entrada		<ul style="list-style-type: none"> - Documento de requerimientos completo y entendido - Programa que pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos y con los estándares de codificación y contabilización - Process Dashboard con todos los datos recolectados al estar realizando el programa
#	Paso	Descripción
1	Registro de Defectos	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Revisar el Resumen del Plan (<i>Project Plan Summary</i>) para verificar que todos los defectos encontrados en cada fase están registrados - Utilizando el mejor recuerdo, registrar en la bitácora de defectos (<i>Defect Log</i>) aquellos defectos que se hayan olvidado registrar
2	Consistencia de los datos de los defectos	<ul style="list-style-type: none"> - Verificar que en la bitácora de defectos (<i>Defect Log</i>) los datos de cada defecto estén correctos y completos - Verificar que en el Resumen del Plan (<i>Project Plan Summary</i>) la cantidad de defectos inyectados y removidos por fase sean razonables y completos - Utilizando el mejor recuerdo, corregir cualquier dato que falte o esté incorrecto
3	Tamaño	<ul style="list-style-type: none"> - Medir el tamaño real del programa - Determinar el tamaño de las partes base (LDC base, borradas, modificadas, agregadas), partes nuevas (cantidad de ítems y tamaño en LDC) y partes reusadas (LDC). - Capturar estos tamaños en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Calcular el Tamaño TOTAL del programa y capturarlo en el Resumen del Plan (<i>Project Plan Summary</i>)
4	Registro de tiempo	<ul style="list-style-type: none"> - Verificar la bitácora de tiempo (<i>Time Log</i>) buscando errores u omisiones - Utilizando el mejor recuerdo, corregir cualquier dato de tiempo que falte o esté incorrecto
5	Mejora de Proceso	<ul style="list-style-type: none"> - Analizar las métricas de la forma Resumen del Plan para encontrar una o más áreas de mejora en el proceso personal - Documentar la(s) propuesta(s) de mejora en la forma Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) describiendo el (los) problema(s), la(s) sugerencia(s) de mejora y/o la(s) lección(es) aprendida(s) - Detener el reloj y marcar la fase de postmortem como terminada - Marcar el programa como terminado
Criterios de Salida		<ul style="list-style-type: none"> - Programa que pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos y con los estándares de codificación y contabilización - Resumen del Plan (<i>Project Plan Summary</i>) lleno con los datos estimados y reales - Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) completa - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas - Reporte de pruebas completo - Formato de Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) completo

Script del Proceso PSP2⁵:

Propósito	Guiar el proceso de desarrollo de un programa computacional, utilizando el Process Dashboard como herramienta de captura de métricas personales.	
Criterios de Entrada	<ul style="list-style-type: none"> - Descripción del Problema - Process Dashboard con las fases del PSP2 asignadas al programa a desarrollar - Forma de Reporte de Casos de Prueba - Estándar de Tipo de Defectos - Estándares de Codificación y Contabilización - Checklists de revisión de diseño y de revisión de código - Datos históricos de tiempo y tamaño 	
#	Paso	Descripción
1	<u>Planeación</u> (Planning)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Obtener la descripción de los requerimientos - Estudiar y entender los requerimientos - Utilizar el método PROBE para estimar las LDC agregadas y modificadas - Utilizar el método PROBE para estimar el tiempo que tomará el desarrollo del programa - Capturar los estimados en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Verificar en la forma Resumen del Plan (<i>Project Plan Summary</i>) que la información planeada es correcta - Detener el reloj y marcar la fase como terminada
2	Diseño (Design)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Utilizando como base los requerimientos: <ul style="list-style-type: none"> ▪ Diseñar en forma completa el programa ▪ Diseñar todos los casos de pruebas unitarias - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
3	<u>Revisión del Diseño</u> (Design Review)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Revisar el diseño siguiendo el script de revisión de diseño y el checklist de revisión de diseño - Corregir todos los defectos que se encuentren - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
4	Codificación (Code)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Escribir el código de acuerdo al diseño y siguiendo los estándares de codificación y contabilización - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
5	<u>Revisión del Código</u> (Code Review)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Revisar el código siguiendo el script de revisión de código y el checklist de revisión de código - Corregir todos los defectos que se encuentren - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
6	Compilación (Compile)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Compilar el código escrito (repetir hasta que el compilador no marque errores) - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
7	Pruebas Unitarias (Test)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Ejecutar todos los casos de prueba diseñados (repetir hasta que ya no se encuentren errores) - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada

(Continúa en la siguiente página)

⁵ Traducido y adaptado del Proceso PSP 2.0 del material académico proporcionado por el SEI.

Script del Proceso PSP2: *(continuación)*

8	<u>Postmortem</u> (Postmortem)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Capturar en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) y en el Resumen del Plan (<i>Project Plan Summary</i>) el tamaño real del programa - Revisar que se hayan registrado en forma completa y correcta todos los tiempos y defectos - Elaborar una Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) - Detener el reloj y marcar la fase y el programa como terminados
Criterios de Salida		<ul style="list-style-type: none"> - Programa que pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos y con los estándares de codificación y contabilización - Resumen del Plan (<i>Project Plan Summary</i>) lleno con los datos estimados y reales - Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) completa - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas - Reporte de pruebas completo - Checklists de revisión de diseño y de revisión de código completos - Formato de Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) completo

Script PSP2 para la Planeación:

Propósito		Guiar el proceso de planeación en PSP
Criterios de Entrada		<ul style="list-style-type: none"> - Descripción del Problema - Process Dashboard con las fases del PSP2 asignadas al programa a desarrollar - Datos históricos de tiempo y tamaño
#	Paso	Descripción
1	Requerimientos del Programa	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Obtener o producir el documento de los requerimientos del programa a desarrollar - Asegurar que los requerimientos son claros y no ambiguos - Obtener respuesta a todas las dudas
2	Estimado de Tamaño	<ul style="list-style-type: none"> - Realizar un diseño conceptual del programa a desarrollar - Capturar los estimados de LDC de todas las partes en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Utilizar el método PROBE para estimar las LDC agregadas y modificadas para este programa (en Process Dashboard utilizar el "PROBE Wizard")
3	Estimado de Tiempo	<ul style="list-style-type: none"> - Utilizar el método PROBE para estimar el tiempo que tomará desarrollar el programa - Capturar el tiempo estimado en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Verificar en la forma Resumen del Plan (<i>Project Plan Summary</i>) que la información planeada es correcta
4	Estimado de defectos	<ul style="list-style-type: none"> - Basado en los datos históricos de defectos totales y los inyectados y removidos en cada fase, Process Dashboard calcula el cantidad de defectos a inyectar y remover en cada fase de este programa - Ajustar los estimados anteriores para incluir los defectos a remover en las revisiones del diseño y del código - Detener el reloj y marcar la fase de planeación como terminada
Criterios de Salida		<ul style="list-style-type: none"> - Documento de requerimientos completo y entendido - Diseño conceptual del programa - Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) completa - Forma Resumen del Plan (<i>Project Plan Summary</i>) con estimados de tiempo y tamaño completos - Bitácora de tiempo (<i>Time Log</i>) completa para esta fase

Script PSP2 para la revisión del diseño:

Propósito	Guiar el proceso de revisión del diseño detallado	
Criterios de Entrada	<ul style="list-style-type: none"> - Diseño del programa completo - Diseño de los casos de prueba completos - Checklist de revisión de diseño - Documento de requerimientos completo y entendido - Estándar de Tipo de Defectos 	
#	Paso	Descripción
1	Preparación	<ul style="list-style-type: none"> - Descansar un momento antes de revisar el diseño - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Analizar la complejidad del diseño y el checklist de revisión, para decidir la mejor estrategia de revisión
2	Revisión	<ul style="list-style-type: none"> - Seguir el checklist de revisión de diseño - Revisar parte por parte - Dentro de cada parte, revisar categoría por categoría - Cada vez que se encuentra un defecto, corregirlo y registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Marcar cada categoría al terminar de revisarla
3	Revisión de Correcciones	<ul style="list-style-type: none"> - Revisar que se hayan corregido todos los defectos encontrados - Revisar todos los cambios realizados - Si se encuentra un defecto, corregirlo y registrarlo en la bitácora de defectos y, si procede, llenar el campo "<i>Fix Defect</i>" del defecto que se estaba arreglando cuando se inyectó este nuevo defecto - Detener el reloj y marcar la fase de revisión de diseño como terminada
Criterios de Salida	<ul style="list-style-type: none"> - Diseño del programa completamente revisado - Diseño de las pruebas completamente revisado - Checklist de revisión marcado en forma completa de acuerdo a las partes que se revisaron - Todos los defectos encontrados fueron corregidos - Todos los cambios fueron revisados - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas 	

Script PSP2 para la revisión del código:

Propósito	Guiar el proceso de revisión del código	
Criterios de Entrada	<ul style="list-style-type: none"> - Código del programa completo - Checklist de revisión de código - Documento de requerimientos completo y entendido - Estándar de Tipo de Defectos - Estándares de Codificación y Contabilización 	
#	Paso	Descripción
1	Preparación	<ul style="list-style-type: none"> - Descansar un momento antes de revisar el código - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Analizar la complejidad del código y el checklist de revisión, para decidir la mejor estrategia de revisión
2	Revisión	<ul style="list-style-type: none"> - Seguir el checklist de revisión de código - Revisar parte por parte - Dentro de cada parte, revisar categoría por categoría - Cada vez que se encuentra un defecto, corregirlo y registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Marcar cada categoría al terminar de revisarla

(Continúa en la siguiente página)

Script PSP2 para la revisión del código: *(continuación)*

3	Revisión de Correcciones	<ul style="list-style-type: none"> - Revisar que se hayan corregido todos los defectos encontrados - Revisar todos los cambios realizados - Si se encuentra un defecto, corregirlo y registrarlo en la bitácora de defectos y, si procede, llenar el campo "Fix Defect" del defecto que se estaba arreglando cuando se inyectó este nuevo defecto - Detener el reloj y marcar la fase de revisión de código como terminada
Criterios de Salida		<ul style="list-style-type: none"> - Código del programa completamente revisado - Checklist de revisión marcado en forma completa de acuerdo a las partes que se revisaron - Todos los defectos encontrados fueron corregidos - Todos los cambios fueron revisados - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas

Script PSP2 para el Postmortem:

Propósito		Guiar el proceso de postmortem en PSP
Criterios de Entrada		<ul style="list-style-type: none"> - Documento de requerimientos completo y entendido - Programa que pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos y con los estándares de codificación y contabilización - Process Dashboard con todos los datos recolectados al estar realizando el programa - Checklists de revisión de diseño y de revisión de código completos
#	Paso	Descripción
1	Registro de Defectos	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Revisar el Resumen del Plan (<i>Project Plan Summary</i>) para verificar que todos los defectos encontrados en cada fase están registrados - Utilizando el mejor recuerdo, registrar en la bitácora de defectos (<i>Defect Log</i>) aquellos defectos que se hayan olvidado registrar
2	Consistencia de los datos de los defectos	<ul style="list-style-type: none"> - Verificar que en la bitácora de defectos (<i>Defect Log</i>) los datos de cada defecto estén correctos y completos - Verificar que en el Resumen del Plan (<i>Project Plan Summary</i>) la cantidad de defectos inyectados y removidos por fase sean razonables y completos - Calcula el yield del proceso y verifica que el valor es razonable y correcto - Utilizando el mejor recuerdo, corregir cualquier dato que falte o esté incorrecto
3	Tamaño	<ul style="list-style-type: none"> - Medir el tamaño real del programa - Determinar el tamaño de las partes base (LDC base, borradas, modificadas, agregadas), partes nuevas (cantidad de ítems y tamaño en LDC) y partes reusadas (LDC). - Capturar estos tamaños en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Calcular el Tamaño TOTAL del programa y capturarlo en el Resumen del Plan (<i>Project Plan Summary</i>)
4	Registro de tiempo	<ul style="list-style-type: none"> - Verificar la bitácora de tiempo (<i>Time Log</i>) buscando errores u omisiones - Utilizando el mejor recuerdo, corregir cualquier dato de tiempo que falte o esté incorrecto
5	Mejora de Proceso	<ul style="list-style-type: none"> - Analizar las métricas de la forma Resumen del Plan para encontrar una o más áreas de mejora en el proceso personal - Documentar la(s) propuesta(s) de mejora en la forma Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) describiendo el (los) problema(s), la(s) sugerencia(s) de mejora y/o la(s) lección(es) aprendida(s) - Detener el reloj y marcar la fase de postmortem como terminada - Marcar el programa como terminado
Criterios de Salida		<ul style="list-style-type: none"> - Programa que pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos y con los estándares de codificación y contabilización - Resumen del Plan (<i>Project Plan Summary</i>) lleno con los datos estimados y reales - Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) completa - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas - Reporte de pruebas completo - Checklists de revisión de diseño y de revisión de código completos - Formato de Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) completo

Script del Proceso PSP2.1⁶:

Propósito	Guiar el proceso de desarrollo de un programa computacional, utilizando el Process Dashboard como herramienta de captura de métricas personales.
Criterios de Entrada	<ul style="list-style-type: none"> - Descripción del Problema - Process Dashboard con las fases del PSP2.1 asignadas al programa a desarrollar - Forma de Reporte de Casos de Prueba - Estándar de Tipo de Defectos - Estándares de Codificación y Contabilización - Checklists de revisión de diseño y de revisión de código - Datos históricos de tiempo y tamaño

#	Paso	Descripción
1	<u>Planeación</u> (Planning)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Obtener la descripción de los requerimientos - Estudiar y entender los requerimientos - Utilizar el método PROBE para estimar las LDC agregadas y modificadas y, en el caso de los métodos A y B, el intervalo de predicción del tamaño - Utilizar el método PROBE para estimar el tiempo que tomará el desarrollo del programa y, en el caso de los métodos A y B, el intervalo de predicción del tiempo - Capturar los estimados en la Plantilla de Estimación de Tamaño (Size Estimating Template) - Verificar en la forma Resumen del Plan (Project Plan Summary) que la información planeada es correcta - Detener el reloj y marcar la fase como terminada
2	Diseño (Design)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Estudiar los requerimientos y producir una especificación externa para cumplirlos - Registrar esta especificación externa en las plantillas de especificación operacional y funcional - Producir un diseño para cumplir con esta especificación externa - Registrar este diseño en las plantillas de especificación de estados y lógica - Basado en la especificación operacional diseñar todos los casos de pruebas unitarias - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (Defect Log) - Detener el reloj y marcar la fase como terminada
3	<u>Revisión del Diseño</u> (Design Review)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Revisar el diseño siguiendo el script de revisión de diseño y el checklist de revisión de diseño - Corregir todos los defectos que se encuentren - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (Defect Log) - Detener el reloj y marcar la fase como terminada
4	Codificación (Code)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Escribir el código de acuerdo al diseño y siguiendo los estándares de codificación y contabilización - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (Defect Log) - Detener el reloj y marcar la fase como terminada
5	<u>Revisión del Código</u> (Code Review)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Revisar el código siguiendo el script de revisión de código y el checklist de revisión de código - Corregir todos los defectos que se encuentren - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (Defect Log) - Detener el reloj y marcar la fase como terminada
6	Compilación (Compile)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Compilar el código escrito (repetir hasta que el compilador no marque errores) - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (Defect Log) - Detener el reloj y marcar la fase como terminada

(Continúa en la siguiente página)

⁶ Traducido y adaptado del Proceso PSP 2.1 del material académico proporcionado por el SEI.

Script del Proceso PSP2.1: *(continuación)*

7	Pruebas Unitarias (Test)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Ejecutar todos los casos de prueba diseñados (repetir hasta que ya no se encuentren errores) - Cada vez que se corrige un defecto, registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Detener el reloj y marcar la fase como terminada
8	<u>Postmortem</u> (Postmortem)	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Capturar en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) y en el Resumen del Plan (<i>Project Plan Summary</i>) el tamaño real del programa - Revisar que se hayan registrado en forma completa y correcta todos los tiempos y defectos - Elaborar una Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) - Detener el reloj y marcar la fase y el programa como terminados
Criterios de Salida		<ul style="list-style-type: none"> - Programa que pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos y con los estándares de codificación y contabilización - Resumen del Plan (<i>Project Plan Summary</i>) lleno con los datos estimados y reales - Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) completa - Plantillas de diseño completas - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas - Reporte de pruebas completo - Checklists de revisión de diseño y de revisión de código completos - Formato de Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) completo

Script PSP2.1 para la Planeación:

Propósito		Guiar el proceso de planeación en PSP
Criterios de Entrada		<ul style="list-style-type: none"> - Descripción del Problema - Process Dashboard con las fases del PSP2.1 asignadas al programa a desarrollar - Datos históricos de tiempo y tamaño
#	Paso	Descripción
1	Requerimientos del Programa	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Obtener o producir el documento de los requerimientos del programa a desarrollar - Asegurar que los requerimientos son claros y no ambiguos - Obtener respuesta a todas las dudas
2	Estimado de Tamaño	<ul style="list-style-type: none"> - Realizar un diseño conceptual del programa a desarrollar - Capturar los estimados de LDC de todas las partes en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Utilizar el método PROBE para estimar las LDC agregadas y modificadas para este programa (en Process Dashboard utilizar el "PROBE Wizard") - Para los métodos A y B, calcular el intervalo de 70% de predicción del tamaño (en Process Dashboard utilizar el "PROBE Wizard")
3	Estimado de Tiempo	<ul style="list-style-type: none"> - Utilizar el método PROBE para estimar el tiempo que tomará desarrollar el programa - Capturar el tiempo estimado en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Para los métodos A y B, calcular el intervalo de 70% de predicción del tiempo (en Process Dashboard utilizar el "PROBE Wizard") - Verificar en la forma Resumen del Plan (<i>Project Plan Summary</i>) que la información planeada es correcta
4	Estimado de defectos	<ul style="list-style-type: none"> - Basado en los datos históricos de defectos totales y los inyectados y removidos en cada fase, Process Dashboard calcula el cantidad de defectos a inyectar y remover en cada fase de este programa - Detener el reloj y marcar la fase de planeación como terminada

(Continúa en la siguiente página)

Script PSP2.1 para la Planeación: *(continuación)*

Criterios de Salida	<ul style="list-style-type: none"> - Documento de requerimientos completo y entendido - Diseño conceptual del programa - Plantilla de Estimación de Tamaño (Size Estimating Template) completa - Forma Resumen del Plan (Project Plan Summary) con estimados de tiempo y tamaño completos y con intervalos de predicción del tamaño y del tiempo - Bitácora de tiempo (Time Log) completa para esta fase
----------------------------	--

Script PSP2.1 para la revisión del diseño:

Propósito	Guiar el proceso de revisión del diseño detallado	
Criterios de Entrada	<ul style="list-style-type: none"> - Diseño del programa completo - Diseño de los casos de prueba completos - Checklist de revisión de diseño - Documento de requerimientos completo y entendido - Estándar de Tipo de Defectos 	
#	Paso	Descripción
1	Preparación	<ul style="list-style-type: none"> - Descansar un momento antes de revisar el diseño - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Analizar la complejidad del diseño y el checklist de revisión, para decidir la mejor estrategia de revisión
2	Revisión	<ul style="list-style-type: none"> - Seguir el checklist de revisión de diseño - Revisar parte por parte - Dentro de cada parte, revisar categoría por categoría - Utilizar tablas de rastreo u otros métodos para verificar que el diseño es correcto - Cada vez que se encuentra un defecto, corregirlo y registrarlo en la bitácora de defectos (Defect Log) - Marcar cada categoría al terminar de revisarla
3	Revisión de Correcciones	<ul style="list-style-type: none"> - Revisar que se hayan corregido todos los defectos encontrados - Revisar todos los cambios realizados - Si se encuentra un defecto, corregirlo y registrarlo en la bitácora de defectos y, si procede, llenar el campo "Fix Defect" del defecto que se estaba arreglando cuando se inyectó este nuevo defecto - Detener el reloj y marcar la fase de revisión de diseño como terminada
Criterios de Salida	<ul style="list-style-type: none"> - Diseño del programa completamente revisado - Diseño de las pruebas completamente revisado - Checklist de revisión marcado en forma completa de acuerdo a las partes que se revisaron - Resultados documentados del análisis del diseño - Todos los defectos encontrados fueron corregidos - Todos los cambios fueron revisados - Bitácora de tiempo (Time Log) y de defectos (Defect Log) completas 	

Script PSP2.1 para la revisión del código:

Propósito		Guiar el proceso de revisión del código
Criterios de Entrada		<ul style="list-style-type: none"> - Código del programa completo - Checklist de revisión de código - Documento de requerimientos completo y entendido - Estándar de Tipo de Defectos - Estándares de Codificación y Contabilización
#	Paso	Descripción
1	Preparación	<ul style="list-style-type: none"> - Descansar un momento antes de revisar el código - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Analizar la complejidad del código y el checklist de revisión, para decidir la mejor estrategia de revisión
2	Revisión	<ul style="list-style-type: none"> - Seguir el checklist de revisión de código - Revisar parte por parte - Dentro de cada parte, revisar categoría por categoría - Cada vez que se encuentra un defecto, corregirlo y registrarlo en la bitácora de defectos (<i>Defect Log</i>) - Marcar cada categoría al terminar de revisarla
3	Revisión de Correcciones	<ul style="list-style-type: none"> - Revisar que se hayan corregido todos los defectos encontrados - Revisar todos los cambios realizados - Si se encuentra un defecto, corregirlo y registrarlo en la bitácora de defectos y, si procede, llenar el campo "Fix Defect" del defecto que se estaba arreglando cuando se inyectó este nuevo defecto - Detener el reloj y marcar la fase de revisión de código como terminada
Criterios de Salida		<ul style="list-style-type: none"> - Código del programa completamente revisado - Checklist de revisión marcado en forma completa de acuerdo a las partes que se revisaron - Todos los defectos encontrados fueron corregidos - Todos los cambios fueron revisados - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas

Script PSP2.1 para el Postmortem:

Propósito		Guiar el proceso de postmortem en PSP
Criterios de Entrada		<ul style="list-style-type: none"> - Documento de requerimientos completo y entendido - Programa que pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos y con los estándares de codificación y contabilización - Process Dashboard con todos los datos recolectados al estar realizando el programa - Checklists de revisión de diseño y de revisión de código completos - Plantillas de diseño completas
#	Paso	Descripción
1	Registro de Defectos	<ul style="list-style-type: none"> - Iniciar el reloj (pausarlo cada vez que haya una interrupción) - Revisar el Resumen del Plan (<i>Project Plan Summary</i>) para verificar que todos los defectos encontrados en cada fase están registrados - Utilizando el mejor recuerdo, registrar en la bitácora de defectos (<i>Defect Log</i>) aquellos defectos que se hayan olvidado registrar
2	Consistencia de los datos de los defectos	<ul style="list-style-type: none"> - Verificar que en la bitácora de defectos (<i>Defect Log</i>) los datos de cada defecto estén correctos y completos - Verificar que en el Resumen del Plan (<i>Project Plan Summary</i>) la cantidad de defectos inyectados y removidos por fase sean razonables y completos - Calcular el yield del proceso y verifica que el valor es razonable y correcto - Utilizando el mejor recuerdo, corregir cualquier dato que falte o esté incorrecto

(Continúa en la siguiente página)

Script PSP2.1 para el Postmortem: *(continuación)*

3	Tamaño	<ul style="list-style-type: none"> - Medir el tamaño real del programa - Determinar el tamaño de las partes base (LDC base, borradas, modificadas, agregadas), partes nuevas (cantidad de ítems y tamaño en LDC) y partes reusadas (LDC). - Capturar estos tamaños en la Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) - Calcular el Tamaño TOTAL del programa y capturarlo en el Resumen del Plan (<i>Project Plan Summary</i>)
4	Registro de tiempo	<ul style="list-style-type: none"> - Verificar la bitácora de tiempo (<i>Time Log</i>) buscando errores u omisiones - Utilizando el mejor recuerdo, corregir cualquier dato de tiempo que falte o esté incorrecto
5	Mejora de Proceso	<ul style="list-style-type: none"> - Analizar las métricas de la forma Resumen del Plan para encontrar una o más áreas de mejora en el proceso personal - Documentar la(s) propuesta(s) de mejora en la forma Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) describiendo el (los) problema(s), la(s) sugerencia(s) de mejora y/o la(s) lección(es) aprendida(s) - Detener el reloj y marcar la fase de postmortem como terminada - Marcar el programa como terminado
Criterios de Salida		<ul style="list-style-type: none"> - Programa que pasa exitosamente todas las pruebas unitarias y que cumple con los requerimientos y con los estándares de codificación y contabilización - Resumen del Plan (<i>Project Plan Summary</i>) lleno con los datos estimados y reales - Plantilla de Estimación de Tamaño (<i>Size Estimating Template</i>) completa - Bitácora de tiempo (<i>Time Log</i>) y de defectos (<i>Defect Log</i>) completas - <i>Plantillas de diseño completas</i> - Reporte de pruebas completo - Checklists de revisión de diseño y de revisión de código completos - Formato de Propuesta de Mejora de Proceso (<i>Process Improvement Proposal</i>) completo