

Personal Software Process - ISO/IEC 12207

Oscar León Trureo
Sebastián Menéndez Sáez
Claudio Piña Novoa

Universidad Tecnológica Metropolitana

28 de octubre de 2012

Índice

- 1 Personal Software Process
 - Introducción
 - Historia
 - PSP
- 2 ISO/IEC 12207
 - Introducción
 - Historia
 - Desarrollo1
 - Desarrollo2
- 3 Conclusión
- 4 Bibliografía

Introducción

Dentro del mundo del desarrollo en Software, podemos encontrar las metodologías, las cuales son un conjunto de buenas prácticas, las cuales nos permiten:

- Crear mejores aplicaciones.
- Llevar un mejor proceso de desarrollo.
- Agilizar el desarrollo en sí.
- etc . . .

Introducción

Dentro del mundo del desarrollo en Software, podemos encontrar las metodologías, las cuales son un conjunto de buenas prácticas, las cuales nos permiten:

- Crear mejores aplicaciones.
- Llevar un mejor proceso de desarrollo.
- Agilizar el desarrollo en sí.
- etc . . .

Introducción

Dentro del mundo del desarrollo en Software, podemos encontrar las metodologías, las cuales son un conjunto de buenas prácticas, las cuales nos permiten:

- Crear mejores aplicaciones.
- Llevar un mejor proceso de desarrollo.
- Agilizar el desarrollo en sí.
- etc . . .

Introducción

Dentro del mundo del desarrollo en Software, podemos encontrar las metodologías, las cuales son un conjunto de buenas prácticas, las cuales nos permiten:

- Crear mejores aplicaciones.
- Llevar un mejor proceso de desarrollo.
- Agilizar el desarrollo en sí.
- etc ...

Introducción

Dentro del mundo del desarrollo en Software, podemos encontrar las metodologías, las cuales son un conjunto de buenas prácticas, las cuales nos permiten:

- Crear mejores aplicaciones.
- Llevar un mejor proceso de desarrollo.
- Agilizar el desarrollo en sí.
- etc . . .

Introducción

Existen una gran cantidad de metodologías, las cuales pueden estar enfocadas al desarrollo en sí, a la gestión, a la calidad, al desarrollador, como también puede ser una mezcla.

Personal Software Process, es una metodología enfocada a la calidad del desarrollo del software a nivel personal, la cual se basa en factores que veremos más adelante.

Historia

- Creado en el año 1995 por Watt's S. Humphrey en la Universidad de Carnegie Mellon, en Pittsburgh, Pennsylvania.
- El primer curso fue impartió en la Universidad de Carnegie Mellon.
- fue plasmado en el libro “A Discipline for SW Engineering” de Humphrey.

Historia

- Creado en el año 1995 por Watt's S. Humphrey en la Universidad de Carnegie Mellon, en Pittsburgh, Pennsylvania.
- El primer curso fue impartió en la Universidad de Carnegie Mellon.
- fue plasmado en el libro “A Discipline for SW Engineering” de Humphrey.

Historia

- Creado en el año 1995 por Watt's S. Humphrey en la Universidad de Carnegie Mellon, en Pittsburgh, Pennsylvania.
- El primer curso fue impartió en la Universidad de Carnegie Mellon.
- fue plasmado en el libro “A Discipline for SW Engineering” de Humphrey.

Historia

- Creado en el año 1995 por Watt's S. Humphrey en la Universidad de Carnegie Mellon, en Pittsburgh, Pennsylvania.
- El primer curso fue impartió en la Universidad de Carnegie Mellon.
- fue plasmado en el libro “A Discipline for SW Engineering” de Humphrey.

“La calidad del software está dada por la cantidad de procesos usados para desarrollarlo y mantenerlo”.

– Watts S. Humphrey, *Creador de Personal Software Process*

PSP

Personal Software Process, que en español significa **Proceso Personal de Software** (PSP), es un conjunto de buenas prácticas las cuales se enfocan al control del tiempo y en la productividad de los Ingenieros en Software, ya sea en la mantencion de sistemas o en tareas de desarrollo.

PSP fue diseñado para ayudar a los profesionales del software para que utilicen constantemente prácticas sanas de ingeniería del software, enseñándoles a planificar y dar seguimiento a un trabajo, utilizar un proceso bien definido y medido, a establecer metas medurables y finalmente a rastrear constantemente para obtener las metas definidas.

Principios de PSP

Personal Software Process, se caracteriza porque es de uso personal, es por esto que se definen principios enfocados a la individualidad:

- Cada ingeniero es diferente, para ser más eficiente, debe planificar su trabajo basándose en su experiencia personal.
- Cada ingeniero debe utilizar procesos bien definidos y medibles.
- Los ingenieros deben asumir la responsabilidad personal de la calidad de sus productos.
- Cuanto antes se detecten y corrijan los errores, menor será el esfuerzo necesario para cumplir la meta.
- Es más efectivo prevenir los defectos, que detectarlos y corregirlos.

Principios de PSP

Personal Software Process, se caracteriza porque es de uso personal, es por esto que se definen principios enfocados a la individualidad:

- Cada ingeniero es diferente, para ser más eficiente, debe planificar su trabajo basándose en su experiencia personal.
- Cada ingeniero debe utilizar procesos bien definidos y medibles.
- Los ingenieros deben asumir la responsabilidad personal de la calidad de sus productos.
- Cuanto antes se detecten y corrijan los errores, menor será el esfuerzo necesario para cumplir la meta.
- Es más efectivo prevenir los defectos, que detectarlos y corregirlos.

Principios de PSP

Personal Software Process, se caracteriza porque es de uso personal, es por esto que se definen principios enfocados a la individualidad:

- Cada ingeniero es diferente, para ser más eficiente, debe planificar su trabajo basándose en su experiencia personal.
- Cada ingeniero debe utilizar procesos bien definidos y medibles.
- Los ingenieros deben asumir la responsabilidad personal de la calidad de sus productos.
- Cuanto antes se detecten y corrijan los errores, menor será el esfuerzo necesario para cumplir la meta.
- Es más efectivo prevenir los defectos, que detectarlos y corregirlos.

Principios de PSP

Personal Software Process, se caracteriza porque es de uso personal, es por esto que se definen principios enfocados a la individualidad:

- Cada ingeniero es diferente, para ser más eficiente, debe planificar su trabajo basándose en su experiencia personal.
- Cada ingeniero debe utilizar procesos bien definidos y medibles.
- Los ingenieros deben asumir la responsabilidad personal de la calidad de sus productos.
- Cuanto antes se detecten y corrijan los errores, menor será el esfuerzo necesario para cumplir la meta.
- Es más efectivo prevenir los defectos, que detectarlos y corregirlos.

Principios de PSP

Personal Software Process, se caracteriza porque es de uso personal, es por esto que se definen principios enfocados a la individualidad:

- Cada ingeniero es diferente, para ser más eficiente, debe planificar su trabajo basándose en su experiencia personal.
- Cada ingeniero debe utilizar procesos bien definidos y medibles.
- Los ingenieros deben asumir la responsabilidad personal de la calidad de sus productos.
- Cuanto antes se detecten y corrijan los errores, menor será el esfuerzo necesario para cumplir la meta.
- Es más efectivo prevenir los defectos, que detectarlos y corregirlos.

PSP no es un estándar, es más bien una alternativa que permite mejorar la forma en la que se construye el software, pero con un enfoque “individual”, por lo que es muy recomendada para los desarrolladores que estén interesados en mejorar en lo que llamamos “desarrollo individual”.



Figura: Niveles de la Organización

Pasos para la implementación de PSP

- 1 Los ingenieros deben ser entrenados por un instructor calificado de PSP.
- 2 La Capacitacion es sobre grupos o equipos, y seran grupos que asi lo han sido y seguiran siendo.
- 3 Requiere un fuerte soporte de administración, en este sentido es necesario que los administradores entiendan PSP, saber como apoyarlos y como monitorear sus avances, sin un adecuado monitoreo los ingenieros caerán otra vez en los malos hábitos.
- 4 Después de ser bien entrenados y bien administrados lo que sigue es optimizar la interacción entre equipos y aquí entraría Team Software Process, el TSP extiende y refina los metodos de CMM y PSP sobre desarrollo y mantenimiento de equipos, y llegar a lo que se le llama un equipo autodirigido.

Pasos para la implementación de PSP

- 1 Los ingenieros deben ser entrenados por un instructor calificado de PSP.
- 2 La Capacitacion es sobre grupos o equipos, y seran grupos que asi lo han sido y seguiran siendo.
- 3 Requiere un fuerte soporte de administración, en este sentido es necesario que los administradores entiendan PSP, saber como apoyarlos y como monitorear sus avances, sin un adecuado monitoreo los ingenieros caerán otra vez en los malos hábitos.
- 4 Después de ser bien entrenados y bien administrados lo que sigue es optimizar la interacción entre equipos y aquí entraría Team Software Process, el TSP extiende y refina los metodos de CMM y PSP sobre desarrollo y mantenimiento de equipos, y llegar a lo que se le llama un equipo autodirigido.

Pasos para la implementación de PSP

- 1 Los ingenieros deben ser entrenados por un instructor calificado de PSP.
- 2 La Capacitacion es sobre grupos o equipos, y seran grupos que asi lo han sido y seguiran siendo.
- 3 Requiere un fuerte soporte de administración, en este sentido es necesario que los administradores entiendan PSP, saber como apoyarlos y como monitorear sus avances, sin un adecuado monitoreo los ingenieros caerán otra vez en los malos hábitos.
- 4 Después de ser bien entrenados y bien administrados lo que sigue es optimizar la interacción entre equipos y aquí entraría Team Software Process, el TSP extiende y refina los metodos de CMM y PSP sobre desarrollo y mantenimiento de equipos, y llegar a lo que se le llama un equipo autodirigido.

Pasos para la implementación de PSP

- 1 Los ingenieros deben ser entrenados por un instructor calificado de PSP.
- 2 La Capacitacion es sobre grupos o equipos, y seran grupos que asi lo han sido y seguiran siendo.
- 3 Requiere un fuerte soporte de administración, en este sentido es necesario que los administradores entiendan PSP, saber como apoyarlos y como monitorear sus avances, sin un adecuado monitoreo los ingenieros caerán otra vez en los malos hábitos.
- 4 Después de ser bien entrenados y bien administrados lo que sigue es optimizar la interacción entre equipos y aquí entraría Team Software Process, el TSP extiende y refina los metodos de CMM y PSP sobre desarrollo y mantenimiento de equipos, y llegar a lo que se le llama un equipo autodirigido.

Ciclo de Vida de PSP

Personal Software Process, tiene un ciclo de vida el cual consta de 5 fases principales, las cuales son fundamentales para el desarrollo de esta metodología:

- 1 Planeación
- 2 Diseño de Alto Nivel
- 3 Revisión de Alto Nivel
- 4 Desarrollo Ciclico
- 5 Post Mortem

Ciclo de Vida de PSP

Personal Software Process, tiene un ciclo de vida el cual consta de 5 fases principales, las cuales son fundamentales para el desarrollo de esta metodología:

- 1 Planeación
- 2 Diseño de Alto Nivel
- 3 Revisión de Alto Nivel
- 4 Desarrollo Ciclico
- 5 Post Mortem

Ciclo de Vida de PSP

Personal Software Process, tiene un ciclo de vida el cual consta de 5 fases principales, las cuales son fundamentales para el desarrollo de esta metodología:

- 1 Planeación
- 2 Diseño de Alto Nivel
- 3 Revisión de Alto Nivel
- 4 Desarrollo Ciclico
- 5 Post Mortem

Ciclo de Vida de PSP

Personal Software Process, tiene un ciclo de vida el cual consta de 5 fases principales, las cuales son fundamentales para el desarrollo de esta metodología:

- 1 Planeación
- 2 Diseño de Alto Nivel
- 3 Revisión de Alto Nivel
- 4 Desarrollo Ciclico
- 5 Post Mortem

Ciclo de Vida de PSP

Personal Software Process, tiene un ciclo de vida el cual consta de 5 fases principales, las cuales son fundamentales para el desarrollo de esta metodología:

- 1 Planeación
- 2 Diseño de Alto Nivel
- 3 Revisión de Alto Nivel
- 4 Desarrollo Ciclico
- 5 Post Mortem

Ciclo de Vida de PSP

Personal Software Process, tiene un ciclo de vida el cual consta de 5 fases principales, las cuales son fundamentales para el desarrollo de esta metodología:

- 1 Planeación
- 2 Diseño de Alto Nivel
- 3 Revisión de Alto Nivel
- 4 Desarrollo Ciclico
- 5 Post Mortem

Fase de Planeación

Entrada

Descripción del problema, resumen del proyecto, resumen cíclico, tamaño estimado, tiempo estimado, formas de planeación.

Actividad

Requerimientos, tamaño estimado, desarrollo estrategia, estimados de recursos, planificación y programas de tareas, estimación de defectos.

Salida

Diseño conceptual, resumen plan, resumen del ciclo, patrones de estimados de tamaño y planeación de tareas, programas de patrones de planeación, registro de tiempos.

Fase de Diseño de Producto

Entrada

Tipificación requerimientos, diseño conceptual, patrones de estimaciones de tamaño, resumen parte ciclico, seguimiento.

Actividad

Especificaciones externas, diseño modular, prototipos, estrategia de desarrollo y documentación, seguimiento.

Salida

Diseño de programa, escenarios operacionales, especificación de funciones y lógica, resumen cíclico, seguimiento y estrategias de pruebas.

Fase Revisión o Validación del Diseño

Entrada

Programa de diseño, escenarios operacionales, especificación de funciones y lógica, resumen ciclico, seguimiento y estrategia de pruebas y ciclo.

Actividad

Diseño de apariencia, verificación de máquinas y lógica, consistencia del diseño, reuso, estrategia de verificación, detectar errores.

Salida

Diseño de alto nivel, registro de seguimiento, tiempos y defectos.

Fase de Desarrollo o Implementación

Entrada

Diseño de alto nivel, registro de seguimiento, tiempos y defectos, ciclo de desarrollo, estrategia de pruebas, patrones de operación y función.

Actividad

Diseño de módulos, revisión de diseño, código, revisión de código, compilación, pruebas, aseguramiento de calidad y del ciclo.

Salida

Modulos de sw, patrón de diseño, lista de verificación de código y diseño, resumen del ciclo, patrón de reporte de pruebas, registro de tiempo, defectos y seguimiento.

Fase PostMortem, Evaluación Ciclo

Entrada

Definición de problema y requerimientos, plan de proyecto y de ciclo, producto de software, patrón de diseño, lista de verificación de código y diseño, resumen del ciclo, patrón de reporte de pruebas, registro de tiempo, defectos y seguimiento.

Actividad

Defectos previstos, removidos, tamaño, tiempo del producto.

Salida

Producto, listas de verificación, plan de proyecto y ciclo, patrón de reporte de pruebas y diseño, forma con propuesta de mejora, registro seguimiento pruebas y tiempo.

Ciclo de Vida

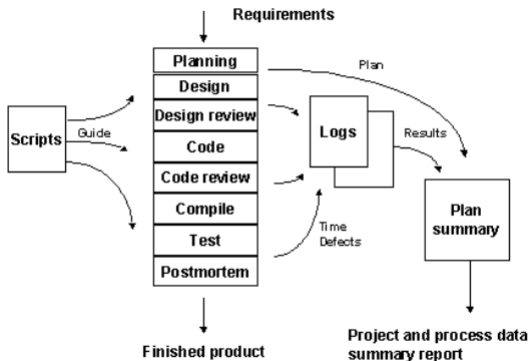


Figura: Ciclo de Vida PSP

Requisitos

En esta fase se definen los requisitos los cuales son definidos claramente en psp como:

- Descripción del problema.
- Especificación de componentes.
- Formas de procesos.
- Estimadores del tamaño del producto y tiempos en base a históricos.

Ciclo de Vida

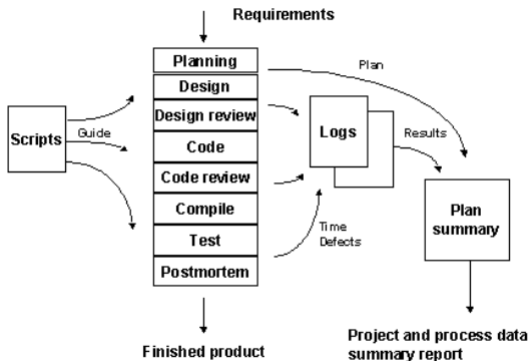


Figura: Ciclo de Vida PSP

Introducción

contenido introducción

Historia

contenido historia ISO

Desarrollo1

contenido Desarrollo1

Desarrollo2

contenido Desarrollo2

Conclusión

aquí va la conclusión

Bibliografía



Victor M. Fleites Sabido

Personal Software Process,

<http://www.slideshare.net/Tonymx/introduccion-a-personal-software-process>.



Armando David Espinoza Robles

Metodologías de Desarrollo de Software,

<http://www.slideshare.net/juliopari/4-clase-metodologia-de-desarrollo-de-software>.

<http://calidadesoftware.wordpress.com/2012/02/23/personal-software-process/>

<http://es.pdfsb.com/readonline/5a56464364516835575846394358706118608>

<http://ingsw.ccbas.uaa.mx/sitio/images/material/psp.htm>