

Personal Software Process - ISO/IEC 12207

Oscar León Trureo
Sebastián Menéndez Sáez
Claudio Piña Novoa

Universidad Tecnológica Metropolitana

26 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

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 el PSP, saber como apoyarlos y como monitorear sus avances, sin un adecuado monitoreo los ingenieros caeran otra vez en los malos habitos.
- 4 Después de ser bien entrenados y bien administrados lo que sigue es optimizar la interaccion 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 el PSP, saber como apoyarlos y como monitorear sus avances, sin un adecuado monitoreo los ingenieros caeran otra vez en los malos habitos.
- 4 Después de ser bien entrenados y bien administrados lo que sigue es optimizar la interaccion 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 el PSP, saber como apoyarlos y como monitorear sus avances, sin un adecuado monitoreo los ingenieros caeran otra vez en los malos habitos.
- 4 Después de ser bien entrenados y bien administrados lo que sigue es optimizar la interaccion 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 el PSP, saber como apoyarlos y como monitorear sus avances, sin un adecuado monitoreo los ingenieros caeran otra vez en los malos habitos.
- 4 Después de ser bien entrenados y bien administrados lo que sigue es optimizar la interaccion 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 (PSP), tiene un ciclo de vida el cual consta de 7 fases, 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
- 6 Integración
- 7 Pruebas

Ciclo de Vida de PSP

Personal Software Process (PSP), tiene un ciclo de vida el cual consta de 7 fases, 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
- 6 Integración
- 7 Pruebas

Ciclo de Vida de PSP

Personal Software Process (PSP), tiene un ciclo de vida el cual consta de 7 fases, 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
- 6 Integración
- 7 Pruebas

Ciclo de Vida de PSP

Personal Software Process (PSP), tiene un ciclo de vida el cual consta de 7 fases, 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
- 6 Integración
- 7 Pruebas

Ciclo de Vida de PSP

Personal Software Process (PSP), tiene un ciclo de vida el cual consta de 7 fases, 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
- 6 Integración
- 7 Pruebas

Ciclo de Vida de PSP

Personal Software Process (PSP), tiene un ciclo de vida el cual consta de 7 fases, 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
- 6 Integración
- 7 Pruebas

Ciclo de Vida de PSP

Personal Software Process (PSP), tiene un ciclo de vida el cual consta de 7 fases, 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
- 6 Integración
- 7 Pruebas

Ciclo de Vida de PSP

Personal Software Process (PSP), tiene un ciclo de vida el cual consta de 7 fases, 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
- 6 Integración
- 7 Pruebas

Introducción

contenido introducción

Historia

contenido historia OSCAL

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>