

PSP

LISSET CRISTINA ALFONSO FELICIANO

JULIO 02 2022

1 INTRODUCCION

En este trabajo vamos a ver que el Personal Software Process (PSP) ayuda a los desarrolladores de software a mejorar su funcionamiento disciplinando la manera en que desarrollan software. De acuerdo con las prácticas encontradas en el modelo de la madurez de la capacidad integrado (CMMI), el PSP puede ser utilizado como guía a un acercamiento disciplinado y estructurado para el desarrollo de software. El PSP se puede aplicar a muchas partes del proceso del desarrollo del software, incluyendo el análisis, definición de requerimientos, desarrollo del programa, documentación, las pruebas del sistema, y mantenimiento. PSP es un proceso de software diseñado para ser utilizado por un Desarrollador de Software. Fue desarrollado por Watts Humphrey y se describe en su libro “A discipline for Software Engineering”. Es utilizado para guiar el planeamiento y desarrollo de módulos de software o pequeños programas, pero es adaptable a otras tareas personales.

2 ¿QUE ES PSP?

El Proceso Personal Software, conocido por sus siglas como PSP, es un proceso de reciente creación, proveniente del Instituto de Ingeniería del Software (SEI). PSP es una alternativa dirigida a los ingenieros de sistemas, que les permite mejorar la forma en la que construyen software. Considerando aspectos como la planeación, calidad, estimación de costos y productividad, PSP es un proceso que vale la pena revisar cuando el ingeniero de software está interesado en aumentar la calidad de los productos de software que desarrolla dentro de un contexto de trabajo individual. Atendiendo a la premisa de que existe una fuerte relación entre las habilidades de los ingenieros de software y la calidad de los productos que desarrollan, las actividades establecidas en PSP están orientadas al conocimiento, administración y mejora de sus habilidades al construir programas. En PSP todas las tareas y actividades que el ingeniero de software debe realizar durante el proceso de desarrollo de un producto de software, están puntualmente definidas en un conjunto de documentos conocidos como scripts. Los scripts son el punto medular de PSP, por lo que se hace mucho énfasis en que deben ser seguidos en forma disciplinada, ya que de ello dependerá el éxito

de la mejora que se busca. Gran parte de las tareas y actividades definidas en los scripts generará en su realización un conjunto de datos, fundamentalmente de carácter estadístico. La aplicación de PSP en varios procesos de desarrollo, y el análisis de la información estadística generada en cada uno de éstos, permitirán al ingeniero de software identificar, tanto sus fortalezas como sus debilidades, y crecer a través de un proceso de autoaprendizaje y auto-mejora. Los scripts se organizan en cuatro niveles, identificados del 0 al 3, atendiéndose en cada nivel un conjunto de aspectos a mejorar del proceso de desarrollo de software. Al primer nivel se le conoce como 0 o de medición personal, al segundo como nivel 1 o de planeación personal, al tercero, como nivel 2 o de calidad personal, y al cuarto, como nivel 3 o cíclico personal. Cada uno de estos niveles, con excepción del 3, tiene una versión que los extiende, introduciendo tareas y actividades para un mejor manejo de los aspectos de interés en nivel, o bien para incluir nuevos aspectos.

La disciplina del PSP provee un marco estructurado para desarrollar habilidades personales y métodos que se necesitarán más adelante para ir forjando al ingeniero de software. Es importante que la calidad del software desarrollado abarque hasta el más mínimo detalle, por muy pequeño que éste sea, ya que, si no se hace así, puede dañar el sistema entero. El Personal Software Process, PSP, indica a los profesionales de software cómo:

- •Administrar la calidad de los proyectos de desarrollo de software
- •Reducir defectos en los productos
- •Estimar y planear el trabajo

PSP, fue diseñado para ayudar al ingeniero de software a hacer bien el trabajo, aplicar métodos avanzados de ingeniería en su labor diaria, a utilizar métodos detallados de planificación y estimación y controlar rendimientos frente a los tiempos planeados. PSP es la disciplina del trabajo con alta calidad. El trabajo del ingeniero de software según PSP, se puede resumir en planificar el trabajo, hacer el trabajo de acuerdo con el plan y producir productos de máxima calidad. El costo del personal representa más del 70% El PSP puede ser usado por los ingenieros de software como una guía disciplinada y estructurada para el desarrollo de software. El PSP es un prerrequisito para una compañía que planea introducir el TSP. El PSP puede ser aplicado en cualquier parte del proceso de desarrollo de software, como por ejemplo desarrollo de pequeños programas, definición de requerimientos, documentación, aseguramiento de calidad y mantenimiento de software.

3 PRINCIPIOS Y OBJETIVOS DE PSP

El PSP es un conjunto ordenado de procesos definidos que orientan a los ingenieros de software a medir, evaluar y monitorear la manera de hacer sus tareas. Los principales objetivos del PSP son:

- 1. Mejorar las estimaciones
- 2. Mejorar la planeación y acompañamiento de cronogramas
- 3. Proteger contra el exceso de compromisos
- 4. Crear un compromiso personal con la calidad
- 5. Compromiso del desarrollador en la mejora continua del proceso de desarrollo
- 6. Aumento de la calidad a través de la reducción de la incidencia de errores
- 7. Mayor precisión en las estimaciones de tamaño del software y tiempo de desarrollo

4 PRINCIPIOS EN QUE SE BASA EL PSP

El diseño del PSP se basó en los siguientes principios de planificación y mejora de la calidad:

- Cada ingeniero es diferente; para ser más eficiente, debe planificar su trabajo basándose en datos tomados de su propia trayectoria profesional.
- Para mejorar auténticamente su trabajo, los ingenieros deben usar procesos personales bien definidos y cuantificados.
- Para obtener productos de calidad, el ingeniero debe asumir la responsabilidad personal de la calidad de sus productos. Los buenos productos no se obtienen por azar, sino como consecuencia de un esfuerzo positivo para hacer un trabajo de calidad.
- Cuanto antes se detecten y corrijan los defectos menos esfuerzo será necesario.
- Es mas efectivo evitar los defectos que detectarlos y corregirlos.
- Trabajar bien es siempre la forma más rápida y económica de trabajar

5 OBJETIVOS DEL PSP

- Planificar el trabajo
- Esforzarse para cumplir la planificación
- Esforzarse para obtener productos de la mejor calidad

6 CARACTERÍSTICAS:

- Es una metodología de la Ingeniería de Software con fundamentos de CMMI.
- Tiene un enfoque hacia la producción de software de calidad
- Favorece los procesos de estimación, planeación y desarrollo de software.
- Como todo proceso de calidad, está orientada a mantener la mejora continua.
- Se puede establecer junto con los modelos de calidad TSP y CMMI.
- Es un proceso definido y ayuda a medir la mejora.
- Involucra actividades de revisión e inspección.
- Está diseñado para uso individual
- Se combinan actividades de: administración de proyectos, Ingeniería de software y calidad. Cuando se logra un proceso de PSP, estable y maduro tiene las siguientes ventajas:
 - Estimación más precisa de tiempos, costos y recursos.
 - Cumplir con los compromisos.
 - Productividad en aumento.
 - Localización de los defectos desde fases iniciales.
 - Mejora los tiempos del ciclo de vida.
 - Reduce costos.
 - Facilita el seguimiento a procesos.
- Los desarrolladores comprenden su condición actual y obtienen un ambiente y disciplina que propicia la mejora de su capacidad. Cuando la implementación del PSP no tiene una administración adecuada, ocurren las siguientes desventajas:
 - Puede considerarse como un proceso burocrático porque genera documentación.
 - La metodología es muy precisa, puede propiciar la exageración en su aplicación.
 - Implementar esta metodología puede consumir mucho tiempo extra
 - La renuencia por parte de los desarrolladores de aplicar la documentación y por sentirse expuestos al evidenciar sus tiempos de desarrollo y defectos.
 - Si no se considera como una inversión, generará la idea de ser un proceso muy costoso a un corto y mediano plazo.

7 PSP tiene unas actividades que se definen en su marco de trabajo

son las siguientes:

- planeacion
- desarrollo
- post mortem

En la Planeación se analizan los requisitos a desarrollar despejamos dudas y aclaramos ambigüedades y se realiza la estimación. El Desarrollo se divide a su vez en: Diseño: En esta etapa se diseña el programa de acuerdo con los requerimientos, se diseñan las pruebas a ejecutar en la etapa de pruebas de desarrollo y se Ingresan en el log de defectos cualquier defecto encontrado con respecto a los requerimientos, mientras se hace el diseño. Ingresar en el log de tiempos, el tiempo total gastado en el diseño. Código, Aquí se realiza la Implementación del diseño. Se registra cualquier defecto encontrado con respecto al diseño. Se Ingresan en el log de tiempos, el tiempo total gastado en la codificación. Compilación: Compilar el programa hasta que no haya errores de compilación. En esta actividad se reparan todos los defectos encontrados. Se registran cualquier defecto encontrado con respecto al código. Se ingresar en el log de tiempos, el tiempo total gastado en compilación. Pruebas: Probar hasta que no haya errores. Reparar todos los defectos encontrados. Registrar cualquier defecto encontrado con respecto al código. Ingresar en el log de tiempos, el tiempo total gastado en pruebas. Post Mortem: En esta actividad se analizan los resultados y descubrir en que proceso se insertaron más bugs y analizar las lecciones aprendidas, además se revisa el resumen de las anteriores actividades para:

- recordar y guardar en el log cualquier defecto que fuese omitido
- corregir la informacion guardada sobre los defectos y corregir
- corregir cualquier error en la recopilacion de tiempos

8 ESTRUCTURA

Está formado por siete niveles incrementales. Cada nivel contiene todas las actividades del nivel anterior más una o dos actividades nuevas y está enfocado hacia un problema en particular. Cada actividad está acompañada de una o más formas que apoyan el proceso.

9 NIVELES DEL MODELO PSP

PSP tiene un marco de proceso de evolución similar al que tiene CMM. PSP trata parcialmente 12 de las 18 KPA's definidas en el CMM. Las KPA's son las áreas de procesos clave o Key Process Areas por su significado en inglés, estas áreas ayudan a guiar a los programadores a que exista un mejoramiento notable en el proceso de software. En CMM un nivel de madurez sólo se alcanza si se logran cumplir todas las KPA's que exige cada nivel. Sin embargo PSP sólomente cubre de manera parcial estas KPA's debido a que es un complemento de CMM y no depende uno del otro en ningún sentido por lo que es considerado como material de apoyo. Como se ha visto anteriormente el Instituto de la Ingeniería del Software (SEI) ha desarrollado el proceso personal del software para definir y reparar la holgura que existe entre el modelo de la madurez de la capacidad y el individuo. Por lo tanto, es ideal utilizarlo junto con CMM pero no es obligatorio ya que es un proceso y no un modelo como lo es CMM. Para desarrollar software de alta calidad, cada componente individual también debe de contar con la más alta calidad posible. La estrategia total de PSP es cerciorarse de que todos los componentes individuales se desarrollen con la más alta calidad. PSP logra esto proporcionando un marco de proceso personal ya definido que el programador puede utilizar. Este marco es:

- Desarrollar un plan para cada proyecto y/o componente
- Registrar su tiempo de desarrollo.
- Registrar sus defectos
- Conservar sus datos en informes del proyecto
- Utilizar sus datos para planear los proyectos y/o los componentes futuros.
- Analizar sus datos para desarrollar sus procesos con mas calidad para mejorar su funcionamiento.

El proceso personal de software fue diseñado para ayudar y guiar a los ingenieros de software a realizar bien su trabajo y haciendo esto pueden llegar a cubrir las KPA's requeridas. PSP también muestra como aplicar métodos avanzados de ingeniería a sus proyectos y/o deberes diarios. Asimismo provee métodos de estimación y de planeación muy bien detallados que son necesarios para dar un seguimiento a su trabajo. La disciplina del PSP provee un marco estructurado para desarrollar habilidades personales y métodos que se necesitarán más adelante para ir forjando al ingeniero de software. Es importante que la calidad del software desarrollado abarque hasta el más mínimo detalle, por muy pequeño que éste sea, ya que si no se hace así, puede dañar el sistema entero. La figura 2 muestra un diagrama que contiene todos los niveles de PSP. Asimismo se muestra que cada nivel cuenta con sus propios requerimientos o KPA's pertenecientes únicamente a PSP pero que podrían compartir intereses con las KPA's de CMM. Es importante para las personas o empresas que

quieran implementar PSP saber que deben de cumplir con todas las KPA's para que avancen de la mejor manera posible al siguiente nivel. Cabe recalcar que se puede "personalizar" el proceso agregando o removiendo tareas conforme a las exigencias de cada persona o empresa. Esto quiere decir que por lo mismo de que PSP es un proceso y no un modelo, se puede amoldar a las necesidades del programador.

10 VENTAJAS:

En el método PSP se puede encontrar una serie de ventajas en la productividad del personal desarrollador.

- Mejora la productividad de las personas implicadas al desarrollo del software
- Se reducen los errores en la codificación.
- Se lleva un mejor control del trabajo individual.
- Se mejora la calidad.
- Facilita la identificación de las fortalezas y las falencias para entrar a mejorarlas.

11 DESVENTAJAS:

Todo lo anterior nos muestra lo práctico que es usar este método, pero puede presentar una serie de desventajas

- La tensión emocional por sentirse controlado.
- El manejo del tiempo al hacer el registro de los tiempos

12 PROPÓSITO

- Ayudar al ingeniero de software a realizar mejor su trabajo.
- Proporcionar datos y técnicas de análisis que se pueden utilizar para determinar que tecnología y métodos aplicar.
- Establecer la estructura que permita comprender por qué se cometen los errores y cómo se pueden detectar.

13 PSP, se centra en la administración del tiempo y en la administración de la calidad a través de la eliminación temprana de defectos

En mi experiencia trabajando con PSP en el proceso de desarrollo y después de varias iteraciones utilizándolo me ayudo a ser más acertada en la estimación de los requerimientos ya que tenía datos para comparar desarrollos de funcionalidades semejantes, con las revisiones e inspecciones que propone el proceso se logra identificar (mediante un checklist), el mayor número de defectos posibles en cada componente, no solamente a nivel de lógica, allí también se revisa estándares de codificación, el modelo propone una revisión personal del código y del diseño y una revisión de pares que la elabora un tercero (desarrollador que no esté involucrado en la construcción del componente que se revise), los bugs que de esta revisión salgan se registran en la herramienta. Esto permite que el producto contenga la menor cantidad de defectos cuando se encuentre en la etapa de pruebas de integración de componentes, con lo cual se pretende reducir los tiempos de las fases de pruebas de integración y de sistema. Una de las herramientas más reconocidas que nos ayudan a llevar el proceso personal del Software y que en mi opinión es la más sencilla de utilizar es el Process Dashboard una herramienta OpenSource en Java. Esta herramienta incluye gráficos que muestran el porcentaje de avance real del equipo con respecto al plan. La herramienta oficial que propone el SEI es el Student Workbook. Entrenarse en PSP es necesario para proporcionar a los ingenieros conocimiento y habilidades para usar TSP (Team Software Process) un método de establecimiento y mejora del trabajo en equipo para procesos software.

14 MOTIVACIONES

- Demostrar los principios del proceso personal
- Apoyar el desarrollo de planes más precisos.
- Determinar los pasos para lograr la mejora de la calidad de los productos.
- Establecer bases de comparación para medir la mejora del proceso personal
- Determinar el impacto en el desempeño de los cambios del proceso.

15 BENEFICIOS DE PSP

- Los datos y su análisis permitirán determinar las fortalezas y debilidades.
- Los datos y su análisis posterior conducirán hacia nuevas ideas para la mejora del proceso.

- Se tendrá control total sobre el calendario, aceptando sólo aquellos compromisos que se puedan cumplir. Si se enfrenta con una presión no razonable, puede recurrir a la base de datos histórica de desempeño y demostrar que no es posible establecer el compromiso.
- Se gana un sentido de satisfacción personal.
- La parte de calidad ayudará a producir mejores productos de trabajo.
- El equipo de trabajo tendrá mayor confianza porque existe una disciplina para el desarrollo de los productos.

16 BENEFICIOS DE LA METODOLOGÍA EN LA CONDUCTA DEL EDUCANDO EL PSP

muestra cómo producir de forma regular software de alta calidad. Utilizando el PSP se obtienen datos que muestran la efectividad del trabajo y se identifican los puntos fuertes y las debilidades, además se practican habilidades y métodos que ingenieros del software van a desarrollar durante muchos años de pruebas y errores. El PSP enseña a ingenieros y futuros ingenieros, cómo administrar la calidad de sus productos y cómo hacer compromisos que ellos puedan cumplir. Puede ser empleado en muchas fases en el ciclo de desarrollo de programas pequeños, definición de requerimientos, documentación, pruebas y mantenimiento. (Velasco, 2003) El diseño de PSP se basa en los siguientes principios de planeación y de calidad (Humphrey).

- Cada ingeniero es esencialmente diferente; es decir, los ingenieros deben planear su trabajo y basar sus planes en sus propios datos personales.
- Para mejorar constantemente su funcionamiento, los ingenieros deben utilizar personalmente procesos bien definidos y medidos.
- Para desarrollar productos de calidad, los ingenieros deben sentirse personalmente comprometidos con la calidad de sus productos.
- Para hacer un trabajo de ingeniería de software de la manera correcta, los ingenieros deben planear de la mejor manera su trabajo antes de comenzar y deben utilizar un proceso bien definido para realizar de la mejor manera la planeación del trabajo.
- Para que los desarrolladores lleguen a entender su funcionamiento de manera personal, deben medir el tiempo que pasan en cada proceso, los defectos que inyectan y remueven de cada proyecto y finalmente medir los diferentes tamaños de los productos que llegan a producir.

Si bien el trabajo disciplinado en cualquier esfera siempre ha demandado el uso de buenos estándares y un soporte sólido, la industria de software no tiene

tradición de un desempeño disciplinado y tampoco incluye este aspecto en su enseñanza de manera profunda. La industria colombiana necesita profesionales con una visión total del proceso de desarrollo de software y con la disciplina para enfrentarlo. La dimensión cognitiva y procedimental en la construcción de software son factores fundamentales en la formación de profesionales en el área de informática, pero los estudiantes de nuestras universidades no cubren en su totalidad el proceso de desarrollo de software, y otros temas como planificación, asignación de recursos, estimación de costos, tamaños y tiempos, definición y trabajo con estándares, calidad, entre otros no son abordados de forma profunda llevando esto a generar una barrera en la inserción en el mundo laboral en casas productoras de software.

17 HERRAMIENTAS AUTOMATIZADAS

PSP requiere de herramientas que permitan:

- Simplificar el proceso de recolección de los datos de tiempo y defectos.
- Automatizar los cálculos requeridos
- Facilitar el acceso a cada una de las formas.
- Llenar automáticamente cada una de las formas con los datos registrados, eliminando la necesidad de copiar la información a mano.
- El control de tiempo puede hacerse más preciso en segundos en lugar de minutos.
- Implementar una guía de tareas automatizada
- Cada usuario pueda adecuar la herramienta a sus prácticas.
- Salvar automáticamente los datos históricos y dejarlos listos para su análisis.

18 PSP EN EL AULA

La implementación de la metodología PSP, en el proceso de enseñanza permite establecer criterios para valorar el rendimiento en los procesos de desarrollo utilizados para la construcción de productos de software. Teniendo en cuenta que PSP es una metodología que se deriva del Modelo de capacidad y madurez, ambos desarrollados por Watts S. Humphrey en el Instituto de Ingeniería del Software SEI, adscrito a la Universidad de Carnegie Mellon. Los métodos de calidad lleva tiempo aprenderlos y practicarlos, pero ayudaran al educando durante su carrera a mejorar consistentemente la calidad del desarrollo. La estrategia es implementar esta metodología en forma paralela con los cursos relacionados con el área de conocimiento de herramientas de programación autoevaluando contantemente su desempeño para evidenciar su aprendizaje y la madurez en

los procesos asociados a la construcción de productos de software. El objetivo primordial es la ejercitación y desarrollo, por parte de los estudiantes, de los roles, habilidades y funciones que deberán desplegar en su vida como profesional de la informática. Es por ello que el acercamiento a las técnicas de PSP se realiza de forma vivencial en las asignaturas que le permiten la construcción de productos de software, que se incrementan en complejidad en forma paralela a su experiencia y conocimiento. En los cursos iniciales se busca solo que los estudiantes se familiaricen con el control del tiempo, no solo de trabajos de programación sino de la vida en general, que interioricen la necesidad de gestionar el tiempo, los compromisos y las programaciones como una vía para mejorar la calidad del software. Con esta iniciación se pretende que los estudiantes reconozcan el valor de registrar los datos de esfuerzo y tamaño y que puedan usar estos datos en la planificación de sus proyectos y analizar su efectividad personal. La estimación de trabajos futuros estaría basada precisamente en la referencia de trabajos anteriores. En las asignaturas de ingeniería de software se implementarían metodologías a nivel grupal y corporativo.

19 COSTO DE PSP

PSP toma tiempo para aprenderlo y aplicarlo. La mejor manera de aprender PSP es tomando el curso que le toma a un programador un total de 130 horas para completarlo. Después de un poco de práctica se acostumbra a usar el método como hábito, sin embargo, la recolección y análisis de los datos requiere de un esfuerzo mayor. Hay que permitir de 30 segundos a un minuto para registrar los tiempos o defectos. Puede tomar hasta una hora completar el reporte de cierre, donde los datos son recolectados y analizados.

Adoptar el PSP puede ser como adoptar una nueva forma de vida para el programador. Demasiadas expectativas sobre una mejora inmediata pueden terminar en frustración cuando no se obtengan los resultados que se esperaban. PSP puede causar un conflicto interno, no debe nunca pensar mucho en una debilidad sino crecerse con su fortaleza.

20 CONCLUSIÓN

Se puede considerar como la guía de trabajo personal para ingenieros de software en organizaciones que emplean un modelo CMMI con nivel de madurez o de capacidad de procesos que implica la medición cualitativa y mejora de procesos. Uno de los mayores problemas que tiene es la gran cantidad de datos que hay que tomar. El PSP tiene obsesión por la toma de datos y elaboración de tablas. El PSP se orienta el conjunto de áreas clave del proceso que debe manejar un desarrollador cuando trabaja de forma individual. El PSP amplía el proceso de mejora a la gente que realiza el trabajo de desarrollo de software, concentrándose en las prácticas de trabajo de los ingenieros en una forma individual, enseñando como manejar la calidad desde el principio de un producto. PSP son nuestras

propias métricas, que permiten estructurar y ordenar nuestro trabajo del día a día (no solo de desarrollo de software, esto lo voy a explicar mas adelante). El resultado de nuestro trabajo, además puede ser llevado a un trabajo en equipo TSP (Team Process Software), el cual es “comandado” por un sistema de gestión de la configuración y por supuesto, un Jefe de Proyecto quien evalúa los resultados y avances de los miembros del equipo.

21 BIBLIOGRAFIA

- 21.1** <https://asprotech.blogspot.com/2010/06/personal-software-process-bsp-elementos.html>
- 21.2** <https://www.unipamplona.edu.co/unipamplona/portallIG/home40/>
- 21.3** <https://lsi2.ugr.es/~mvega/docis/bsp.html>
- 21.4** <https://desarrollossoftware.webnode.es/unidad-iii-proceso-personal-de-desarrollo-de-software-bsp-/>
- 21.5** <https://es.linkedin.com/pulse/bsp-personal-software-process-diana-marcela-software-developer>
- 21.6** <https://www.unipamplona.edu.co/unipamplona/portallIG/hom>
- 21.7** <https://eveconde.neocities.org/unidad4/evidencias/pspytsp.pdf>