

**DIVISIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIÓN**

CARRERA

• DESARROLLO DE SOFTWARE MULTIPLATAFORMA •

ACT 3.1 PSP / ESTÁNDARES Y MÉTRICAS

PROFESORA: RUTH BETSAIDA MARTINEZ DOMINGUEZ

FECHA: 21 DE NOVIEMBRE DEL 2024

CUAT. - GRUPO: 4B-DSM

3ER PARCIAL



EDUARDO ALBERTO FLORES HERRERA



Índice

Introducción.....	3
Desarrollo.....	4
Conclusión	8
Referencias	10

Introducción

El Personal Software Process (PSP) es una metodología de desarrollo personal orientada a mejorar la calidad del software y la productividad del desarrollador individual. Fue desarrollada por Watts Humphrey con el objetivo de proporcionar a los ingenieros de software las herramientas y técnicas necesarias para gestionar su propio trabajo de manera más efectiva. Al aplicar el PSP, los desarrolladores pueden seguir un enfoque estructurado que les permite medir y analizar su desempeño, identificando áreas de mejora a través de métricas cuantificables. Este enfoque les ayuda no solo a producir software de mejor calidad, sino también a cumplir con los plazos establecidos y a minimizar los defectos en el producto final.

El PSP se divide en varios niveles que van desde la planificación y el diseño hasta la codificación, compilación, pruebas y análisis retrospectivo. A través de estos niveles, se busca introducir mejoras continuas en el proceso de desarrollo, promoviendo una mentalidad orientada hacia la calidad y la eficiencia. Además, el PSP enfatiza la importancia de la documentación detallada, el registro de tiempos y defectos, y la revisión constante, para que los desarrolladores puedan aprender de sus errores y mejorar iterativamente.

Este documento detalla la aplicación del PSP a un proyecto tutorizado enfocado en la gestión del personal de un hotel. A lo largo de este proyecto, se implementaron prácticas del PSP para mejorar la planificación, la calidad del código y la gestión del tiempo. Se describirán los procedimientos utilizados y cómo estos contribuyeron al desarrollo eficiente del sistema, además de reflexionar sobre los beneficios obtenidos y los desafíos encontrados durante la implementación del PSP en un entorno de desarrollo real.

Desarrollo

1. Entendiendo el PSP

El PSP es una metodología que divide el proceso de desarrollo de software en diferentes fases, cada una con un enfoque específico para mejorar la calidad del producto y la eficiencia del desarrollador. Las fases principales del PSP incluyen:

- **Planificación:** Estimar el tiempo necesario para completar el proyecto y definir los objetivos.
- **Diseño:** Crear una arquitectura que soporte los requerimientos.
- **Codificación:** Implementar el diseño utilizando buenas prácticas de programación.
- **Compilación:** Identificar errores sintácticos y de integración.
- **Prueba:** Detectar defectos en el software para garantizar su correcto funcionamiento.
- **Análisis retrospectivo:** Reflexionar sobre las fases anteriores y realizar ajustes futuros.

Durante este proyecto, tomé notas sobre los conceptos clave del PSP y me familiaricé con las mejores prácticas de cada fase, las cuales apliqué en el desarrollo de la aplicación de gestión de personal del hotel.

2. Identificación de procedimientos del PSP

Para aplicar los elementos del PSP a mi flujo de trabajo, identifiqué los siguientes procedimientos clave:

- **Registro del Tiempo:** Medí el tiempo invertido en cada fase del desarrollo, utilizando un cronómetro y anotando cada periodo de trabajo en una hoja de cálculo. Esto me ayudó a tener una visión más clara sobre mi productividad y eficiencia. Por ejemplo, dediqué 15 horas a la planificación y

diseño del módulo de gestión de empleados, mientras que la fase de codificación tomó 54 horas.

- **Registro de Defectos:** Documenté cada defecto encontrado durante las fases de compilación y prueba, especificando la fase en la que ocurrió, la naturaleza del error, el tiempo invertido en solucionarlo y el impacto que tuvo. Por ejemplo, durante la implementación del módulo de autenticación, encontré un defecto relacionado con la verificación de roles, que me tomó 5 horas corregir.
- **Medición del Tamaño del Proyecto:** Utilicé métricas como el número de líneas de código (LOC) y el número de funcionalidades para evaluar el tamaño del proyecto. El módulo de gestión de empleados, por ejemplo, consistió en aproximadamente 1577 líneas de código y cubrió funcionalidades como el CRUD de empleados y la asignación de roles.
- **Medición de la Productividad:** Calculé la productividad utilizando el número de líneas de código escritas por hora y el tiempo requerido para solucionar defectos. Durante la fase de desarrollo, logré una productividad promedio de 30 líneas de código por hora, y dediqué un promedio de 2 horas a la corrección de cada defecto identificado.

3. Aplicación de procedimientos al proyecto tutorizado

Para aplicar los procedimientos del PSP, seleccioné un proyecto tutorizado relacionado con el desarrollo de un sistema de gestión para el personal de un hotel. Este sistema permite gestionar los datos de los empleados, sus roles, registros de check-in/check-out, y otras funcionalidades internas. A continuación, detallo cómo se aplicaron los procedimientos del PSP:

- **Planificación:** En la fase de planificación, estimé el tiempo necesario para el desarrollo de cada módulo del sistema, como la gestión de empleados, autenticación y roles. Definí las funcionalidades principales y establecí un calendario para completarlas. Utilicé hojas de cálculo para realizar estimaciones y asegurarme de que cada fase estuviera bien definida. La

planificación inicial estimó 89 horas de desarrollo, distribuidas en 15 horas para el diseño, 54 horas para la codificación y 20 horas para las pruebas.

- **Registro del Tiempo:** Utilicé una hoja de cálculo para registrar el tiempo dedicado a cada fase del proyecto. Esto me permitió identificar cuánto tiempo invertí en cada actividad y comparar las estimaciones iniciales con los tiempos reales. Por ejemplo, la implementación del módulo de autenticación me tomó 12 horas, mientras que la fase de pruebas de este módulo se extendió a 8 horas debido a la complejidad de la lógica de permisos.
- **Registro de Defectos:** Durante la fase de codificación y prueba, encontré varios defectos relacionados con la lógica de autenticación y la gestión de estados de los empleados (activo/baja). Documenté cada defecto en un formato que incluyó: descripción del problema, fase en la que se encontró, tiempo para corregirlo y la causa del defecto. Un ejemplo notable fue un defecto en la lógica de validación de los estados de los empleados, que me tomó 4 horas identificar y corregir. Este proceso de registro me ayudó a identificar patrones comunes y a implementar mejores controles para prevenir errores similares en el futuro.
- **Medición del Tamaño:** Calculé el tamaño del proyecto basándome en la cantidad de líneas de código y en el número de módulos funcionales desarrollados. Por ejemplo, el módulo de gestión de empleados incluyó 1577 líneas de código dividido en empleados y autenticación (empleados 706 líneas de código, mientras que el módulo de autenticación consistió en 871 líneas adicionales. Este enfoque me dio una visión clara del progreso durante el desarrollo y me permitió evaluar la complejidad del proyecto. Además, el módulo de check-in/check-out del personal abarcó 700 líneas de código y tomó aproximadamente 15 horas para ser completado.
- **Medición de Productividad:** Con base en el registro de tiempos y el tamaño del proyecto, pude calcular mi productividad. Durante la fase de codificación del módulo de gestión de empleados, logré una tasa de 30

líneas de código por hora, lo cual estuvo cerca de mi estimación inicial. En la fase de pruebas, la productividad disminuyó debido a la necesidad de corregir varios defectos que no fueron anticipados en la planificación original. Este análisis me permitió ajustar mis expectativas y mejorar la asignación de tiempo para futuros proyectos.

Conclusión

Aplicar los procedimientos del PSP me permitió tener una mejor comprensión de cómo administro mi tiempo y la calidad del código que desarrollo. Durante el proceso, pude identificar los siguientes beneficios:

- **Mayor visibilidad sobre el tiempo invertido en cada fase del desarrollo:**
El registro detallado del tiempo me permitió ajustar mis expectativas y planificación en función de los resultados reales, lo que resultó en una mayor precisión en mis estimaciones para futuras tareas.
- **Reducción de defectos a través de la aplicación sistemática del registro y análisis de errores:** Al documentar cada defecto, pude identificar patrones comunes, como problemas recurrentes en la lógica de validación de roles y estados, y prevenir errores similares en futuras fases del desarrollo.
- **Mejora continua basada en el análisis de mi productividad y eficiencia:**
El PSP me ayudó a identificar áreas donde necesitaba mejorar, como la gestión del tiempo durante la fase de prueba y la eficiencia en la integración de módulos complejos. Por ejemplo, después de identificar que la fase de pruebas me tomaba más tiempo de lo esperado, decidí implementar pruebas más automatizadas para mejorar mi eficiencia.

No obstante, el uso del PSP también presentó algunos desafíos, como la necesidad de dedicar tiempo adicional para llevar registros detallados y hacer ajustes constantes. Sin embargo, considero que estos desafíos fueron compensados por la mejora en la calidad del software y la eficiencia del desarrollo.

En general, el PSP me ayudó a entender mejor mi flujo de trabajo y a realizar mejoras para futuros proyectos, haciendo mi trabajo más organizado y enfocado en la calidad. La implementación del sistema de gestión del personal del hotel fue

un proceso enriquecedor que me permitió aplicar los principios del PSP de manera efectiva, logrando un software más robusto y eficiente.

Referencias

Humphrey, W. S. (2005). PSP: A Self-Improvement Process for Software Engineers. Addison-Wesley Professional.

Pressman, R. S., & Maxim, B. R. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.://blog.hubspot.es/sales/que-es-pmbok