



Ingeniería en Desarrollo de software
Semestre 5

Programa de la unidad didáctica:
Métricas de desarrollo de software (PSP)

Información general de la unidad didáctica

Clave:
15143529

Ciudad de México, febrero de 2025

Universidad Abierta y a Distancia de México





Índice

I. INFORMACIÓN GENERAL DE LA UNIDAD DIDÁCTICA.....	3
a. Ficha de identificación	3
b. Descripción	3
c. Fundamentación de la unidad didáctica.....	4
d. Logros	4
e. Competencia(s) a desarrollar	5
f. Temario.....	5
g. Metodología de trabajo.....	6
h. Evaluación.....	7
i. Fuentes de consulta	9



I. INFORMACIÓN GENERAL DE LA UNIDAD DIDÁCTICA

a. Ficha de identificación

Nombre de la Licenciatura o Ingeniería:	Desarrollo de Software
Nombre del curso o unidad didáctica	Métricas de desarrollo de software (PSP)
Clave de unidad didáctica:	15143529
Seriación:	No aplica
Semestre:	Quinto
Horas contempladas:	72

b. Descripción

El Proceso Personal de Software (por sus siglas en inglés conocido como PSP) es una metodología de reciente creación, proveniente del Instituto de Ingeniería del Software (SEI). PSP va dirigido a los ingenieros de sistemas, les permite mejorar la forma en la que construyen software; considerando aspectos como la planeación, calidad, estimación de costos y productividad. PSP es una metodología 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, esta unidad didáctica tiene como objetivo el analizar las actividades establecidas en PSP orientadas al conocimiento, administración y mejora de las habilidades al construir programas. El alumno será capaz de realizar todas las tareas y actividades básicas que el ingeniero de software debe cumplir durante el proceso de desarrollo de un producto de software.

La unidad didáctica de **Métricas de desarrollo de software (PSP)** forma parte del quinto semestre de la Ingeniería en Desarrollo de Software. Esta unidad didáctica provee al alumno de los conocimientos básicos para continuar con la materia de Desarrollo de software en equipo (TSP) entre otras.

Métricas de desarrollo de software (PSP) está conformada por las unidades:

1. Métricas de desarrollo de Software (PSP).
2. Planeación: introducción, medición y estimación.
3. Planeación: recursos y calendario.



c. Fundamentación de la unidad didáctica

El Proceso Personal de Software (*Personal Software Sroccess, PSP*), es una metodología para aumentar la calidad de los productos de software. Se implementa a nivel personal pero después es integrado a actividades de equipo de trabajo TSP (*Team Software Process*) y estas, a su vez, pueden combinarse con otros modelos de calidad como: CMMI (*Capability Maturity Model Integration*) o la norma mexicana MoProSoft (Modelo de Procesos para la Industria del Software).

PSP es un modelo de calidad que pueden aplicar los ingenieros de software para mejorar la manera en que producen software. Considerando los conceptos de planeación, medición, estimación, productividad, etc. Por lo tanto, es importante cuando un ingeniero de software está interesado en automatizar y mejorar su proceso y producir sus trabajos de desarrollo con calidad. Es una herramienta útil ya que involucra buenas prácticas de la Ingeniería de Software por ejemplo la planificación, seguimiento, medición, etc. hasta obtener un proceso bien definido.

La figura académica en línea:

- Semanalmente programa y actualiza las actividades.
- Complementa el desarrollo de los temas.
- Selecciona temas.
- Plantea preguntas disparadoras.
- Dirige la discusión.
- Retroalimenta.
- Obtiene conclusiones a partir de las participaciones de los estudiantes.
- Cierra el foro con conclusiones y trabajo colaborativo.

d. Logros

Entender la disciplina de métricas de desarrollo de software por medio de:

- Identificar las características de PSP y su relación con TSP y CMMI.
- Analizar el proceso de líneas base, formas, elementos, procesos, mediciones de PSP, defectos y resumen del plan de proyecto.
- Comprender lo que es un plan, su contenido y etapas en un proyecto de software.
- Analizar la medición del tamaño del software utilizando estándares.
- Utilizar el método PROBE.
- Estimar los recursos y tiempo que deben ser planeados.
- Analizar aspectos básicos para combinar estimaciones.
- Aplicación de los procedimientos del PSP 0, 0.1, 1.1.



e. Competencia(s) a desarrollar

Competencia (s) General(es):

- Aplicar el procedimiento Métricas de Desarrollo de Software (PSP) para establecer la mejora continua en el proceso personal de desarrollo de software a través de la identificación de métodos y herramientas de calidad.

Competencias específicas de unidad

- Identificar los conceptos básicos de PSP para analizar un programa, considerando la relación de PSP con TSP (*Team Software Process*) y CMMI (*Capability maturity model integration*).
- Aplicar la medición y estimación para planear un programa, tomando en cuenta el plan, sus etapas, criterios y estándares de medición, así como diferentes métodos de estimación.
- Aplicar la estimación de recursos y tiempos para planear el desarrollo de un programa a través de la elaboración de un calendario con el proceso PSP 1.1.

f. Temario

1. Métricas de desarrollo de software (PSP)

1.1. Antecedentes de PSP y su relación con TSP y CMMI

1.1.1. Antecedentes de PSP

1.1.2. Características de PSP

1.1.3. Relación de PSP con TSP (*Team Software Process*) y CMMI (*Capability maturity model integration*)

1.2. Generalidades de PSP

1.2.1. El proceso de línea base

1.2.2. Las formas y elementos de PSP

1.2.3. Proceso y mediciones del PSP0

1.2.4. Registro de tiempos de defectos

1.2.5. Resumen del plan del proyecto

1.2.6. Personalización del proceso inicial

2. Planeación: Introducción, medición y estimación

2.1. Introducción a la planeación

2.1.1. ¿Qué es un plan?



- 2.1.2. ¿Por qué hacer planes?
 - 2.1.3. Contenido del plan de un software
 - 2.1.4. Planeando un proyecto de software
 - 2.1.5. Producir un plan de calidad
 - 2.1.6. Etapas de la planeación
 - 2.2. Medición del tamaño del software
 - 2.2.1. Medición del tamaño
 - 2.2.2. Establecer un conteo estándar
 - 2.2.3. Contadores de LOC y tipos
 - 2.2.4. Consideraciones del re-uso
 - 2.2.5. Conteo de líneas de código
 - 2.2.6. Calcular la productividad
 - 2.2.7. PSP 0.1
 - 2.3. Estimación del tamaño del software
 - 2.3.1. Contexto
 - 2.3.2. Métodos de estimación
 - 2.3.3. Proxy
 - 2.3.4. PROBE
 - 2.3.5. PSP 1
- 3. Planeación: recursos y calendario**
- 3.1. Estimación de recursos
 - 3.1.1. Planeando el recurso
 - 3.1.2. Estimando el tiempo de desarrollo
 - 3.1.3. Combinación de estimaciones
 - 3.2. Estimación de calendario
 - 3.2.1. Estimación de calendarios
 - 3.2.2. Seguimiento del concepto de valor ganado
 - 3.2.3. PSP 1.1

g. Metodología de trabajo

La unidad didáctica de Métricas de Desarrollo de Software (PSP) está organizada con el enfoque de Aprendizaje basado en problemas. Cada problema que tiene relación con el desarrollo de programas, pues en él se dará una descripción detallada del requerimiento, la teoría del problema con un ejemplo y valores de prueba para validar su funcionalidad. Los problemas que no son programas se basarán en un contexto del área de Desarrollo de software y PSP.

Al aplicar este tipo de metodología en la unidad didáctica, también se toman en cuenta:



- El uso de las siguientes *herramientas tecnológicas*: a) un foro general al inicio de la unidad didáctica cuyo propósito presentarse y conocer a tus compañeros con la idea de favorecer la comunicación y el conocimiento entre los estudiantes, el cual estará abierto durante el desarrollo de la materia para que intercambies experiencias y conocimientos de esta.
- La realización de *actividades formativas*, entre las que destacan: tareas, investigaciones y diseñar diagramas como parte final para la aplicación del conocimiento adquirido.
- La construcción *de evidencias* será a través de planteamiento de problemas a lo largo de las tres unidades; así aplicarás el conocimiento adquirido en cada una de ellas en la resolución de dichos problemas.

Además, se cuenta con otras herramientas didácticas como: el Foro, en donde podrás interactuar con la figura académica y tus compañeros. La evidencia de aprendizaje ayuda a concretar la competencia que se busca lograr por unidad.

h. Evaluación

La evaluación del aprendizaje es un proceso, a través del cual se observa, recoge y analiza información relevante del proceso de aprendizaje de los estudiantes, con la finalidad de reflexionar, emitir juicios de valor, así como tomar decisiones pertinentes y oportunas para optimizarlo (Díaz Barriga A.F. & Hernández R.G., 2005). Orienta la toma de decisiones, da pauta a determinar acciones en términos de valoración de conocimientos, nivel del desempeño, reorientaciones de aprendizaje, mejora del proceso educativo y adecuación de actividades, entre otras acciones.

De acuerdo con lo anterior, mediante la evaluación te brindaremos apoyo y seguimiento para identificar las dificultades en el desarrollo de conocimientos, habilidades y actitudes del proceso integral de aprendizaje.

En el marco del Modelo educativo de la UnADM, la evaluación de la unidad didáctica se realiza en los siguientes momentos: 1) formativa y 2) sumativa.

EVALUACIÓN FORMATIVA

Se realiza en paralelo al desarrollo del proceso de enseñanza-aprendizaje de cada unidad, y sirve para localizar dificultades cuando aún estás en posibilidad de remediarlas.

En este primer momento de evaluación, se aplican estrategias asociadas a las:

- **Actividades individuales** (tareas). Se trata de un primer momento de aprendizaje, en el cual se consideran tus perspectivas, experiencias, intereses, capacidades y necesidades.



- **Actividades colaborativas** (foros). El trabajo colaborativo fomenta y promueve el aprendizaje en contribución con otros compañeros, ya que eres responsable no sólo de tu aprendizaje, sino de contribuir a que los demás aprendan en equipo y se fomente un ambiente de confianza; por ende, que se logren las metas de aprendizaje.

EVALUACIÓN SUMATIVA

Se aplica al final del proceso de tu experiencia de aprendizaje, su propósito es verificar los resultados alcanzados y el grado de aprendizaje o nivel de conocimientos, habilidades y actitudes que hayas adquirido.

Este segundo y último momento de evaluación, se mide y valora a través de las siguientes actividades:

- **Evidencias de aprendizaje.** Son actividades que tienen como objetivo integrar el proceso de construcción de tu aprendizaje, la evaluación, la retroalimentación y la planeación de la nueva ruta de aprendizaje que seguirás de acuerdo con los resultados individuales obtenidos.
- **Actividad complementaria.** Esta actividad es planeada por la figura académica considerando las competencias y logros de la unidad didáctica, toda vez que identifica los conocimientos, habilidades y actitudes que te hizo falta desarrollar o potenciar (se realiza en una ocasión al finalizar la última unidad).
- **Actividad de reflexión.** Es un ejercicio de metacognición que permite que tomes conciencia de tu proceso de aprendizaje, el punto de partida son las experiencias del contexto académico y la reflexión sobre tu desempeño. Se trata de una acción formativa que parte de tu persona y no del saber teórico, que considera tu experiencia de aprendizaje (se realiza en una ocasión al finalizar la última unidad).

A continuación, se presenta el esquema general de evaluación correspondiente a esta unidad didáctica:



Esquema general de evaluación		
Tipo de evaluación	Actividades	Puntaje
Formativa	Actividades individuales	30%
	Actividades colaborativas	10%
Sumativa	Evidencias de aprendizaje	40%
	Actividad complementaria	10%
	Actividad de reflexión	10%
Total		100%

Recuerda que la calificación final que te permitirá acreditar se asigna de acuerdo con los criterios e instrumentos de evaluación establecidos para cada actividad, los cuales son diseñados con base en las competencias y logros de esta unidad didáctica.

i. Fuentes de consulta

Bibliografía básica

Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A software architect's perspective*. Addison-Wesley Professional.

Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change* (2nd ed.). Addison-Wesley Professional.

Cohn, M. (2005). *Agile estimating and planning*. Prentice Hall.

Humphrey, W. (1995). *A discipline for software engineering* (The complete PSP book). Addison-Wesley.

Humphrey, W. (2005). *PSP: A self-improvement process for software engineers*. Addison-Wesley.



- Kerzner, H. (2017). *Project management metrics, KPIs, and dashboards: A guide to measuring and monitoring project performance* (3rd ed.). Wiley.
- Kim, G., Debois, P., Willis, J., Humble, J., & Allspaw, J. (2016). *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations*. IT Revolution Press.
- Humphrey, W. (2006). *TSP(SM): Leading a development team*. Addison-Wesley.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Scrum.org.
- Zapata, J. (2001). *Introducción al proceso software personal, Watts Humphrey*. Addison-Wesley.

Bibliografía complementaria

- Chrissis, M. B., Konrad, M., & Shrum, S. (2009). *CMMI: Guía para la integración de procesos y la mejora de productos* (2ª ed.). Madrid, España.
- <http://www.sei.cmu.edu/library/assets/cmmi-dev-v12-spanish.pdf>.
- García, Y. (2010). *Proceso de software personal*. EcuRed (Enciclopedia colaborativa en la red cubana). Cuba.