



Universidad
Tecmilenio®



Ingeniería de Software

Estimación del software

Semana 10



Te invito a realizar la siguiente actividad de bienestar-mindfulness antes de comenzar a revisar el tema:

<https://youtu.be/16yTNMiA5Ks>



Estimar el esfuerzo que requiere un proyecto es fundamental para calcular los recursos (económicos, humanos, equipamiento y tiempo) que se necesitarán para producir un software de calidad.



Los elementos que intervienen en la estimación de proyectos de desarrollo de software son:

Cantidad y tamaño
de los componentes
necesarios.

Esfuerzo para
desarrollo
de software.

Costo que
debe cubrirse.

Duración o plan
de ejecución.



Existen varios métodos para estimar el tamaño del software y el esfuerzo requerido para completar un sistema:

Modelado
algorítmico
de costos

Modelo
COCOMO II

Estimación del
esfuerzo orientado
a objetos



Los principios que ofrece PSP-Bok para la estimación son:

La estimación es incierta.

La estimación es un proceso de aprendizaje.

La estimación es una habilidad.

Trabaja por ser consistente.

Utiliza un método definido para generar estimaciones.

La estimación esta sujeta al error.

Realiza estimación al detalle.

Utilice datos al hacer estimaciones.



Ejercicio

Realiza una estimación con la técnica de estimación-esfuerzo en programación orientado a objetos, para una pantalla que registre a un alumno en interfaz gráfica web de un sistema de gestión escolar.



Estimar el tamaño el software es una actividad que requiere de un proceso iterativo en el que se realicen ajustes para que la estimación sea más precisa. Esto requiere de la disciplina de los programadores para documentar con precisión el trabajo que realizan.



- Chemuturi, M. (2009). Software estimation best practices, tools & techniques. EE. UU: J. Ross Publishing.
- Pomeroy-Huff, M, Cannon, R., Chick, T., Mullaney, J. y Nichols, W. (2010). The Personal Software Process (PSP) Body of Knowledge, Version 2.0. Recuperado de <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1034&context=sei>



Ingeniería de Software

Planificación
de proyectos

Semana 10



El plan de desarrollo incluye actividades y todos los costos relacionados con el mismo; por lo general suelen ser una carga más elevada para el proyecto que solo el desarrollo. Piensa en las pruebas, la documentación y capacitación. Todas estas actividades deben ser incorporadas al plan desde un inicio.



Las tres razones sobre la importancia de un plan son:

Sin la planeación, cualquier esfuerzo por desarrollar un software no podría ser gestionado de forma adecuada.

Planear es una habilidad que el ingeniero de software puede aprender y mejorar con la práctica.

Un plan permite realizar un mejor trabajo de software.



Un plan define el trabajo que debe realizarse para obtener un resultado esperado.

Incluye la definición y secuencia de tareas, estima los recursos necesarios (tiempo, presupuesto y personas), y asigna sus respectivos responsables.



Los pasos para realizar el proceso de planeación son:

Alcance del
proyecto

Determinar y
estimar actividades

Realizar estimación
inicial

Documentar la
estimación del
proyecto

Acordar cualquier
solicitud de cambio
antes de aceptarla.



Realiza un plan de actividades para el desarrollo de una pantalla que registre a un alumno en interfaz grafica web en un sistema de gestión escolar.



La construcción del software debe ser un ejercicio de planeación que exprese el trabajo que será necesario realizar para crear el sistema o la aplicación.

El proyecto se divide en una secuencia de actividades más pequeñas, lo que permite mantener un control más estrecho del tiempo, costos y recursos asignados.



- Humphrey, W. (2005). PSP(SM): A Self-Improvement Process for Software Engineers. USA: Pearson.



Ingeniería de Software

Planificación del
equipo de trabajo

Semana 10



El diseño de software tiene cuatro etapas que inician desde el análisis o ingeniería de requerimientos, se construye el diseño del sistema y pasa por una revisión técnica que permitirá realizar algunos ajustes al diseño, se procede a realizar una aprobación del mismo y finalmente se tendrá un diseño terminado.



El éxito en un equipo de trabajo reside en los siguientes factores:

Un líder genuino que motive y ayude a tomar buenas decisiones al equipo.

Un equipo que conozca que su misión es exigente pero alcanzable.

Miembros del equipo comprometidos a alcanzar la meta establecida.

Recursos adecuadamente capacitados, con herramientas y apoyo necesarios.



La estrategia Big Bang o cascada, diseña y construye todas las partes de un sistema, las ensambla y revisa si funcionan. Este enfoque es un desastre para desarrollos que son muy grandes, principalmente porque los defectos del sistema son muy difíciles de detectar y corregir a tiempo.



La estrategia enfoque por etapas o versiones incrementales, aborda la construcción del software en partes más pequeñas o en versiones, en las que se pueda aplicar todo el proceso de desarrollo. Esto permite administrar mejor el desempeño de cada parte enfocando el esfuerzo del equipo.



Define la mejor estrategia para desarrollar un software de gestión escolar.



Gran parte del éxito de muchas empresas dedicadas a la fabricación de software radica en el talento de su personal, en el trabajo en equipo y en ambientes de trabajo adecuados.



- Kendall E. y Kendall, J. (2011). *Systems analysis and design* (8ª ed.). New Jersey: Prentice Hall.
- Pohl, K. y Rupp, Chris. (2011). *Requirements Engineering Fundamentals*. EE. UU: Rocky Nook.
- Pressman, R. (2010). *Ingeniería de Software. Un enfoque práctico* (7ª ed.). México: McGraw-Hill.
- Sommerville, I. (2011). *Ingeniería de Software* (9ª ed.). México: Pearson.



Ingeniería de Software

Administración
del riesgo

Semana 10



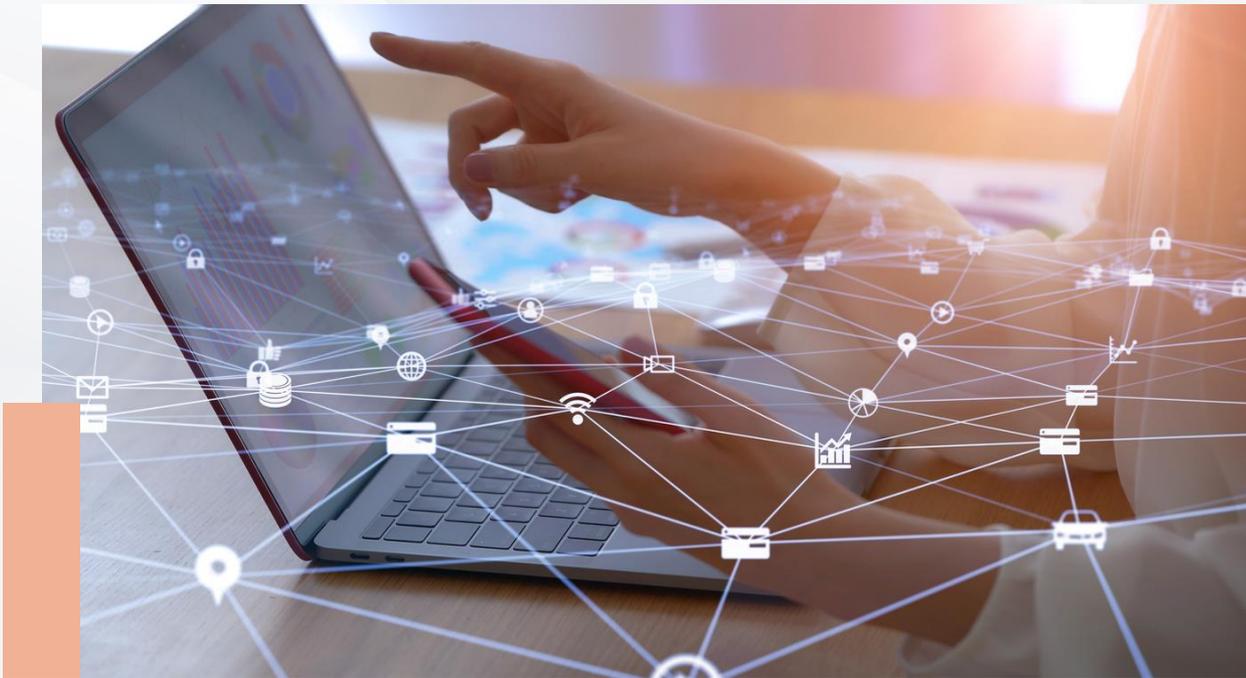
Los proyectos de desarrollo tienen la probabilidad de verse afectados por eventos que pueden poner en riesgo la confiabilidad, seguridad, calidad y efectividad del producto.



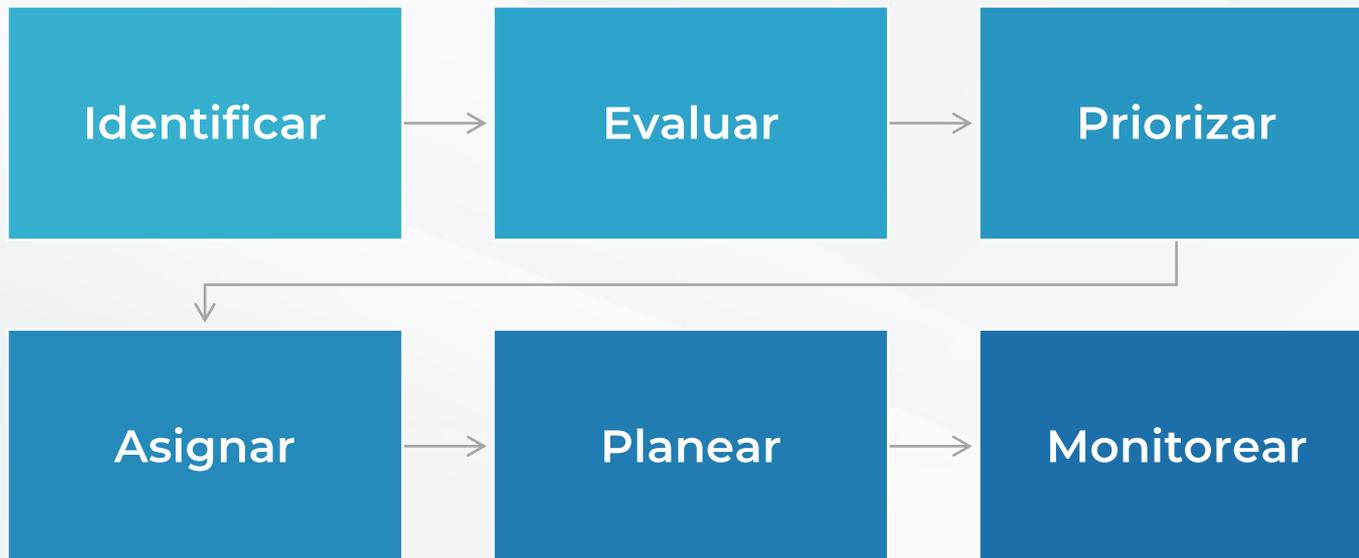
Un riesgo es la probabilidad de que suceda un evento no deseado. En todo proyecto de desarrollo de software existen riesgos que se tienen que administrar. Cualquier evento que comprometa la calidad, la fecha de entrega o el presupuesto estimado, constituye un riesgo para el proyecto.



El objetivo primordial de la gestión del riesgo es prevenir que un riesgo se materialice o encontrar las acciones para mitigar al máximo los posibles efectos en caso de presentarse.



La metodología TSP establece un proceso de administración del riesgo mediante los siguientes pasos:



Enlista los cinco principios de la gestión de riesgos.



Cuando en los proyectos de software se tienen identificados los riesgos de mayor impacto con suficiente anticipación, es posible crear planes de mitigación que puedan ser aplicados para asegurar los factores de éxito del proyecto: tiempos de entrega, calidad, y mantenerse dentro de presupuesto.



- ISACA. (2013). CISA Review Manual. USA: ISACA.
- El-Haik, B. y Shaout, A. (2010). Software Design for Six Sigma: A Roadmap for Excellence. USA: John Wiley & Sons.
- Humphrey, W. (2006). TSP(SM) Coaching Development Teams. USA: Pearson.
- Humphrey, W., Chick, T, Nichols, W, & Pomeroy-Huff, M. (2010). Team Software Process (TPS) Body of Knowledge. USA: SEI.

