**Proceso del software**

La meta de la ingeniería de software es construir productos de software, o mejorar los existentes; en ingeniería de procesos, la meta es desarrollar o mejorar procesos.

Un proceso de desarrollo de software es un conjunto de personas, estructuras de organización, reglas, políticas, actividades y sus procedimientos, componentes de software, metodologías, y herramientas utilizadas o creadas específicamente para definir, desarrollar, ofrecer un servicio, innovar y extender un producto de software.

Un proceso de software efectivo habilita a la organización a incrementar su productividad al desarrollar software:

-Permite estandarizar esfuerzos, promover re-uso, repetición y consistencia entre proyectos.

-Provee la oportunidad de introducir mejores prácticas de la industria.

-Permite entender que las herramientas deben ser utilizadas para soportar un proceso.

-Establece la base para una mayor consistencia y mejoras futuras.

-Un proceso de software mejora los esfuerzos de mantenimiento y soporte:

-Define cómo manejar los cambios y liberaciones a sistemas de software existentes.

-Define cómo lograr la transición del software a la operación, y cómo ejecutar los esfuerzos de operación y soporte.

**VIDA DEL SOFTWARE**

El propósito de este programa es definir las distintas fases intermedias que se requieren para validar el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados.   
  
Estos programas se originan en el hecho de que es muy costoso rectificar los errores que se detectan tarde dentro de la fase de implementación. El ciclo de vida permite que los errores se detecten lo antes posible y, por lo tanto, permite a los desarrolladores concentrarse en la calidad del software, en los plazos de implementación y en los costos asociados.   
  
El ciclo de vida básico de un software consta de los siguientes procedimientos:   
  
Definición de objetivos: define la finalidad del proyecto y su papel en la estrategia global.   
  
Análisis de los requisitos y su viabilidad: recopila, examina y formula los requisitos del cliente y examina cualquier restricción que se pueda aplicar.   
  
Diseño general: requisitos generales de la arquitectura de la aplicación.  
  
Diseño en detalle: definición precisa de cada subconjunto de la aplicación.

Programación (programación e implementación): implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.   
  
Prueba de unidad: prueba individual de cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones.   
  
Integración: garantiza que los diferentes módulos se integren con la aplicación. Este es el propósito de la prueba de integración que está cuidadosamente documentada.   
  
Prueba beta (o validación): garantiza que el software cumple con las especificaciones originales.   
  
Documentación: sirve para documentar información necesaria para los usuarios del software y para desarrollos futuros.   
  
Mantenimiento: comprende todos los procedimientos correctivos (mantenimiento correctivo) y las actualizaciones secundarias del software (mantenimiento continuo).

**Metodología de desarrollo de software**

Lograr la construcción de un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas.

