



Técnico en
< DESARROLLO DE SOFTWARE >

***Metodología de Desarrollo de
Software I***

(CC BY-NC-ND 4.0)
International

Attribution-NonCommercial-NoDerivatives 4.0



Atribución

Usted debe reconocer el crédito de una obra de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace.



No Comercial

Usted no puede hacer uso del material con fines comerciales.



Sin obra derivada

Si usted mezcla, transforma o crea un nuevo material a partir de esta obra, no puede distribuir el material modificado.

No hay restricciones adicionales - Usted no puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros hacer cualquier uso permitido por la licencia.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>



Metodología de Desarrollo de Software I

Unidad I

1. Ingeniería de software

La ingeniería de software es una disciplina que integra procesos, métodos y herramientas para el desarrollo de software. Varios son los modelos de procesos que se han propuesto para la ingeniería de software, cada uno presenta ventajas y desventajas, pero todos tienen en común fases genéricas que permiten llevar a cabo el proceso de la ingeniería de software.

La ingeniería del software permite al diseñador de programas, realizar su tarea de construcción de software como un problema de ingeniería haciendo uso de guías, principios y normas que le permitirán el correcto desarrollo de su labor. Adicionalmente, dispondrá de un conjunto de herramientas que le permitirán la evaluación, validación, depuración y corrección del software desarrollado.

Ciclo de vida del software

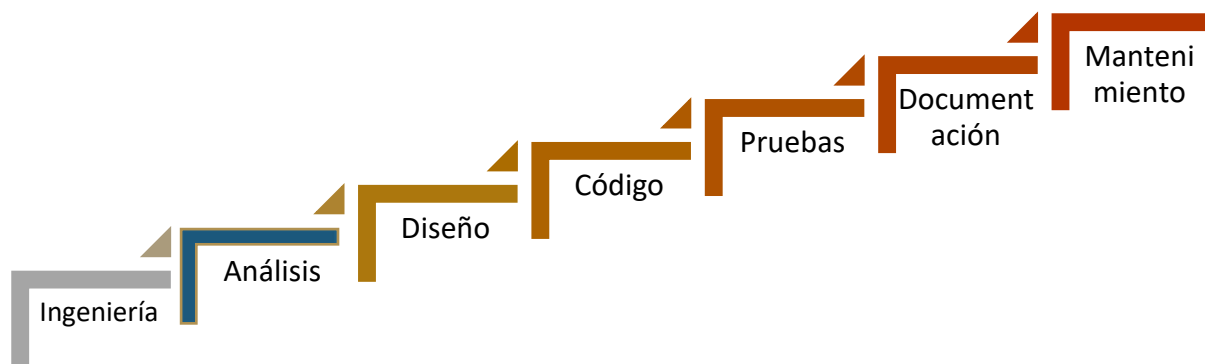
Es la forma mediante la cual se describen los diferentes pasos que se deben seguir para el desarrollo de un software, partiendo desde una necesidad hasta llegar a la puesta en marcha de una solución y su apropiado mantenimiento. El ciclo de vida para un software

comienza cuando se tiene la necesidad de resolver un problema, y termina cuando el programa que se desarrolló para cumplir con los requerimientos, deja de ser utilizado.

Existen varias versiones del ciclo de vida del software entre las cuales se destacan: el ciclo de vida clásico o en cascada, el modelo en espiral, el desarrollo de prototipos, el modelo por incrementos y el modelo extremo

Etapas del ciclo de vida del software

El ciclo de vida clásico del software siendo uno de los más utilizados tal como lo plantean diferentes autores, está conformado en su versión ampliada por siete etapas que se pueden representar mediante un modelo en cascada así:



- **INGENIERÍA DE SISTEMAS:** En esta etapa el analista luego de un minucioso y detallado estudio de los sistemas de una organización, detecta un problema o una necesidad que para su solución y/o satisfacción es necesario realizar un desarrollo de software.
- **ANÁLISIS:** En esta etapa se debe entender y comprender de forma detallada cual es la problemática a resolver, verificando el entorno en el cual se encuentra dicho

problema, de tal manera que se obtenga la información necesaria y suficiente para afrontar su respectiva solución.

- **DISEÑO:** Una vez que se tiene la suficiente información del problema a solucionar, es importante determinar la estrategia que se va a utilizar para resolver el problema. Esta etapa es conocida bajo el CÓMO se va a solucionar.
- **IMPLEMENTACIÓN:** partiendo del análisis y diseño de la solución, en esta etapa se procede a desarrollar el correspondiente programa que solucione el problema mediante el uso de una herramienta computacional determinada.
- **PRUEBAS:** Los errores humanos dentro de la programación de los computadores son muchos y aumentan considerablemente con la complejidad del problema. Cuando se termina de escribir un programa de computador, es necesario realizar las debidas pruebas que garanticen el correcto funcionamiento de dicho programa bajo el mayor número de situaciones posibles a las que se pueda enfrentar.
- **DOCUMENTACIÓN:** Es la guía o comunicación escrita en sus diferentes formas, ya sea en enunciados, procedimientos, dibujos o diagramas que se hace sobre el desarrollo de un programa. La importancia de la documentación radica en que a menudo un programa escrito por una persona, es modificado por otra. Por ello la documentación sirve para ayudar a comprender o usar un programa o para facilitar futuras modificaciones (mantenimiento).

La documentación se compone de tres partes:

a. Documentación Interna: Son los comentarios o mensajes que se añaden al código fuente para hacer más claro el entendimiento de los procesos que lo conforman, incluyendo las precondiciones y las pos condiciones de cada función.

b. Documentación Externa: Se define en un documento escrito con los siguientes puntos:

- ✓ Descripción del Problema
- ✓ Datos del Autor
- ✓ Algoritmo (diagrama de flujo o Pseudocódigo)
- ✓ Diccionario de Datos
- ✓ Código Fuente (programa)

c. Manual de Usuario: Describe paso a paso la manera cómo funciona el programa, con el fin de que el usuario lo pueda manejar para que obtenga el resultado deseado.

- ✓ **MANTENIMIENTO:** una vez instalado un programa y puesto en marcha para realizar la solución del problema previamente planteado o satisfacer una determinada necesidad, es importante mantener una estructura de actualización, verificación y validación que permitan a dicho programa ser útil y mantenerse actualizado según las necesidades o requerimientos planteados durante su vida útil. Para realizar un adecuado mantenimiento, es necesario contar con una buena documentación del mismo.

El producto y el proceso de la Ingeniería del Software

El producto

El software de computadora se ha convertido en el alma máter. Es la máquina que conduce a la toma de decisiones comerciales. Sirve para la investigación científica moderna y de resolución de problemas de ingeniería. Es el factor clave que diferencia los productos y servicios modernos. Está inmerso en sistemas de todo tipo: de

transportes, médicos, de telecomunicaciones, militares, procesos industriales, entretenimientos, productos de oficina, la lista es casi interminable. El software es casi ineludible en un mundo moderno. A medida que nos adentremos en el siglo XXI, será el que nos conduzca a nuevos avances en todo, desde la educación elemental a la ingeniería genética.

El software de computadora es el producto que diseñan y construyen los ingenieros del software. Esto abarca programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, documentos que comprenden formularios virtuales e impresos y datos que combinan números y texto y también incluyen representaciones de información de audio, vídeo e imágenes.

El proceso

El proceso es un diálogo en el que se reúne el conocimiento y se incluye en el software. El proceso proporciona una interacción entre los usuarios y los diseñadores, entre los usuarios y las herramientas de desarrollo, y entre los diseñadores y las herramientas de desarrollo [tecnología]. Es un proceso interactivo donde la herramienta de desarrollo se usa como medio de comunicación, con cada iteración del diálogo se obtiene mayor conocimiento de las personas involucradas.

Cuando se trabaja para construir un producto o un sistema, es importante seguir una serie de pasos predecibles, un mapa de carreteras que le ayude a obtener el resultado oportuno de calidad. El mapa de carreteras a seguir es llamado proceso del software.

Modelos de desarrollo de software

Existen varios modelos, paradigmas y filosofías de desarrollo, en los cuales se apoya la ingeniería de software para la construcción del software, entre ellos se puede citar:

- ✓ Modelo en cascada (modelo tradicional)
- ✓ El Modelo Incremental
- ✓ Modelo de prototipos
- ✓ Modelo Espiral o iterativo.
- ✓ Desarrollo por etapas.
- ✓ Desarrollo iterativo y creciente o Iterativo e Incremental
- ✓ Modelo de desarrollo rápido de aplicaciones (Rapid Application Development, RAD ó DRA)
- ✓ Desarrollo concurrente
- ✓ Proceso Unificado de Desarrollo RUP (Proceso Unificado de Rational)

EL MODELO INCREMENTAL

Es una combinación del Modelo de Cascada y Modelo Por Prototipos.

Reduce el rehacer trabajo durante el proceso de desarrollo y da oportunidad para retrasar las decisiones hasta tener experiencia en el sistema.

Durante el desarrollo de cada incremento se puede utilizar el modelo de cascada o prototipo, dependiendo del conocimiento que se tenga sobre los requisitos a implementar. Si se tiene un buen conocimiento, se puede optar por cascada, si es dudoso, prototipo.

El modelo incremental se centra en la entrega de un producto operacional con cada incremento.

Los primeros incrementos son versiones “incompletas” del producto final, pero proporcionan al usuario la funcionalidad necesaria para su evaluación.

EL MODELO ESPIRAL

El ciclo de desarrollo se representa como una espiral, en lugar de una serie de actividades sucesivas con retrospectiva de una actividad a otra.

Cada ciclo de desarrollo se divide en cuatro fases:

1. Definición de objetivos: Se definen los objetivos. Se definen las restricciones del proceso y del producto. Se realiza un diseño detallado del plan administrativo. Se identifican los riesgos y se elaboran estrategias alternativas dependiendo de estos.
2. Evaluación y reducción de riesgos: Se realiza un análisis detallado de cada riesgo identificado. Pueden desarrollarse prototipos para disminuir el riesgo de requisitos dudosos. Se llevan a cabo los pasos para reducir los riesgos.
3. Desarrollo y validación: Se escoge el modelo de desarrollo después de la evaluación del riesgo. El modelo que se utilizará (cascada, sistemas formales, evolutivo, etc.) depende del riesgo identificado para esa fase.
4. Planificación: Se determina si continuar con otro ciclo. Se planea la siguiente fase del proyecto.

EL MODELO LINEAL

Llamado algunas veces "ciclo de vida básico" o "modelo en cascada", el modelo lineal secuencial sugiere un enfoque sistemático, secuencial, para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

Es un ciclo de vida en sentido amplio, que incluye no sólo las etapas de ingeniería sino toda la vida del producto: las pruebas, el uso (la vida útil del software) y el mantenimiento.

Cuatro P's

El principal objetivo de las Cuatro P's es marcar la línea del éxito del gestor (líder de proyecto) y de sus proyectos, garantizando que durante toda la vida del proyecto exista un gran equilibrio entre el proyecto, los procesos, las personas (roles) y los productos.

Persona

El factor humano siempre será el más importante en el desarrollo de soluciones de software, muchos empresarios famosos, líderes de empresas tecnológicas, coinciden que el éxito que han alcanzado sus empresas no se debe a las herramientas que utilizan, son las personas y el trabajo en equipo.

Producto

Se denomina productos a todos aquellos artefactos que se creen durante la vida del proyecto, modelos, códigos, ejecutable, documentación, diagramas UML, bocetos de la interfaz de usuario, prototipos, componentes, planes de prueba, ingeniería y gestión, colección de modelos, modelos de casos de uso, análisis, diseño, despliegue,

implementación y prueba. Antes de planear un proyecto, se deben establecer los objetivos y el alcance que tendrá el proyecto, además de sus restricciones, herramientas, técnicas y planes de gestión. Con una buena planificación se puede estimar el tiempo que tomará desarrollar o construir el producto y redimensionar el valor cuantitativo del mismo.

Proceso

Se denomina proceso al conjunto de actividades que se realizan para crear el producto (Plantilla para crear el proyecto). El proceso se define en términos de flujos de trabajo (conjunto de actividades), se identifican trabajadores y artefactos, además de que se utilizan diagramas de actividad de UML para describir los flujos de trabajo. El equipo de desarrollo debe elegir el proceso adecuado y que le permita obtener una solución o producto que satisfaga las necesidades o requerimientos del cliente.

Proyecto

- ✓ Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. Es un elemento organizativo de gestión que establece una secuencia de cambios, por el cual va evolucionando diariamente. El mismo establece una serie de iteraciones e hitos dentro de los cuales se desarrollan una serie de casos de usos o requisitos. Debemos tener en cuenta muchos parámetros para realizar un proyecto con la calidad requerida, y saber que existen factores que atentan con el buen desempeño de un proyecto, estos es:
- ✓ El personal de software no entiende las necesidades de los clientes.
- ✓ El ámbito del producto está mal definido.

- ✓ Los cambios se gestionan mal.
- ✓ La tecnología elegida cambia.
- ✓ Las necesidades comerciales cambian.
- ✓ Los plazos de entrega no son realistas.
- ✓ Los usuarios se resisten a la utilización del software.
- ✓ Se pierde el patrocinio.
- ✓ El equipo del proyecto carece de personal con las habilidades apropiadas.
- ✓ Los gestores evitan las mejores prácticas y las lecciones aprendidas.

2. Capability Maturity Model Integration (CMMI)

¿Qué es CMMI?

CMMI (Capability Maturity Model Integration, según sus siglas en Inglés). Modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Desarrollado por el Instituto de Ingeniería de Software de la Universidad Carnegie Mellon (SEI), y publicado en su primera versión en enero de 2002.

Su implementación aumenta la fiabilidad del software producido, la visibilidad de los procesos de producción y soporte, la reusabilidad de componentes, y como resultado de la combinación de este tipo de mejoras, disminuye los costes de producción y mantenimiento de las aplicaciones.

Disciplinas en CMMI

CMMI se aplica a 4 disciplinas distintas y nosotros podemos elegir una de ellas para centrarnos en aspectos específicos.

- ✓ **Ingeniería de Sistema** - Cubre la construcción de un sistema con o sin software
- ✓ **Ingeniería de Software** - Cubre la construcción de soluciones software
- ✓ **Integración de productos y procesos de desarrollo** - Cubre la relación a largo plazo con el cliente.
- ✓ **Relación con proveedores** - Cubre los procesos relacionados con la subcontratación de partes del sistema

Dos representaciones: continua y escalonada

El modelo para software (CMM-SW) establece 5 niveles de madurez para clasificar a las organizaciones, en función de qué áreas de procesos consiguen sus objetivos y se gestionan con principios de ingeniería. Es lo que se denomina un modelo escalonado, o centrado en la madurez de la organización.

El modelo para Ingeniería de Sistemas (SE-CMM) establece 6 niveles posibles de capacidad para una de las 18 áreas de proceso implicadas en la ingeniería de sistemas. No agrupa los procesos en 5 tramos para definir el nivel de madurez de la organización, sino que directamente analiza la capacidad de cada proceso por separado. Es lo que se denomina un modelo continuo.

REPRESENTACIÓN ESCALONADA

Nivel 1 Iniciado: En el nivel de madurez 1, la mayoría de los procesos son "ad-hoc" y caóticos. La organización usualmente no provee un ambiente estable para soportar los procesos. Éxitos en estas organizaciones se debe a la competencia y esfuerzos heroicos de la gente dentro de la organización y no al uso de procesos probados. A pesar de este caos, organizaciones pertenecientes al nivel de madurez 1 con frecuencia producen

productos y servicios que funcionan; sin embargo, ellos frecuentemente exceden sus presupuestos y no cumplen sus planes.

Nivel 2 Repetible: En el nivel de madurez 2 se ordena el caos. En el nivel 2 las organizaciones se enfocan en tareas cotidianas referentes a la administración. Cada proyecto de la organización cuenta con una serie de procesos para llevarlo a cabo, los cuales son planeados y ejecutados de acuerdo con políticas establecidas; los proyectos utilizan gente capacitada quienes disponen de recursos para producir salidas controladas; se involucran a las partes interesadas; son monitoreados, controlados y revisados; y son evaluados según la descripción del proceso

Nivel 3 Definido: En el nivel de madurez 3, procesos son caracterizados y entendidos de buena forma, y son descritos en estándares, procedimientos, herramientas, y métodos. El conjunto de procesos estándares de la organización, los cuales son la base para el nivel de madurez 3, es establecido y mejorado continuamente. Estos procesos estándares son usados para establecer consistencia a través de la organización. Los proyectos establecen sus procesos adaptando el conjunto de procesos estándares de la organización de acuerdo a guías de adaptación.

Nivel 4 Manejado cuantitativamente: En el nivel de madurez 4, la organización y proyectos establecen objetivos cuantitativos para medir la calidad y realización de los procesos y los usa como criterios en el manejo de ellos. Los objetivos cuantitativos son definidos en base a las necesidades de clientes, usuarios finales, organización, y actores de los procesos.

Nivel 5 Optimizado: En el nivel de madurez 5, una organización mejora continuamente sus procesos basándose en el conocimiento de las causas comunes de variación inherente en los procesos. El nivel de madurez 5 se focaliza sobre la mejora continua de los procesos a través de mejoras continuas, incrementales y tecnológicas. Los objetivos de mejora cuantitativa de procesos para la organización son establecidos, continuamente revisados para reflejar cambios en los objetivos del negocio y usados como criterio en la mejora de procesos.

Componentes

Área de proceso: Conjunto de prácticas relacionadas que son ejecutadas de forma conjunta para conseguir un conjunto de objetivos.

COMPONENTES REQUERIDOS

Objetivo genérico: Los objetivos genéricos asociados a un nivel de capacidad establecen lo que una organización debe alcanzar en ese nivel de capacidad.

El logro de cada uno de esos objetivos en un área de proceso significa mejorar el control en la ejecución del área de proceso

Objetivo específico: Los objetivos específicos se aplican a una única área de proceso y localizan las particularidades que describen que se debe implementar para satisfacer el propósito del área de proceso.

Referencias:

Universidad UNAD

Enlace a recurso:

- ✓ http://datateca.unad.edu.co/contenidos/301404/301404_ContentoEnLinea/index.html
- ✓ http://datateca.unad.edu.co/contenidos/301404/301404_ContentoEnLinea/leccion_11_el_modelo_lineal_secuencial.html

EcuRed

Enlace a recurso:

- ✓ <http://www.ecured.cu/index.php?title=CMMI>
- ✓ http://www.ecured.cu/index.php?title=Modelo_en_cascada
- ✓ http://www.ecured.cu/index.php?title=Ingenier%C3%ADa_de_software
- ✓ http://www.ecured.cu/index.php/Ciclo_de_vida_del_software

Microsoft

Enlace a recurso:

- ✓ <https://msdn.microsoft.com/es-gt/library/ee461556.aspx>

Calidad y Software.com

Enlace a recurso:

- ✓ http://www.calidadyssoftware.com/otros/introduccion_cmmi.php

CMMI INSTITUTE

Enlace a recurso:

- ✓ <http://cmmiinstitute.com/resources>

Software Engineering Institute

Enlace a recurso:

- ✓ <http://www.sei.cmu.edu/library/assets/whitepapers/Spanish%20Technical%20Report%20CMMI%20V%201%203.pdf>

Universidad Nacional de Columbia

Enlace a recurso:

- ✓ <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060024/Lecciones/Capitulo%20I/problemas.htm>

Descargo de responsabilidad

La información contenida en este documento descargable en formato PDF o PPT es un reflejo del material virtual presentado en la versión online del curso. Por lo tanto, su contenido, gráficos, links de consulta, acotaciones y comentarios son responsabilidad exclusiva de su(s) respectivo(s) autor(es) por lo que su contenido no compromete al área de e-Learning del Departamento GES o al programa académico al que pertenece.

El área de e-Learning no asume ninguna responsabilidad por la actualidad, exactitud, obligaciones de derechos de autor, integridad o calidad de los contenidos proporcionados y se aclara que la utilización de este descargable se encuentra limitada de manera expresa para los propósitos educativos del curso.



Galileo
UNIVERSIDAD
La Revolución en la Educación

GES
Galileo Educational System