Unidad 1

Definiciones

Ingeniería de software

Es la aplicación práctica del método científico en el diseño y creación de programas informáticos, así como en la documentación necesaria para su desarrollo, operación y mantenimiento.

Método

El método para la ingeniera de software indica la forma en la que se debe construir software. Incluye el análisis de requisitos, el diseño, la elaboración del programa, las pruebas y el mantenimiento.

Herramientas

Se conoce como herramientas al soporte necesario para el proceso y los métodos. Permiten realizar una tarea con mayor exactitud o calidad. También mejorar la productividad, la eficiencia y la calidad del producto.

Proceso

Se define como proceso al marco de trabajo que abarca completamente un conjunto de actividades del desarrollo de software.

Paradigma

El paradigma representa un enfoque particular para la construcción del software específicamente.

Calidad

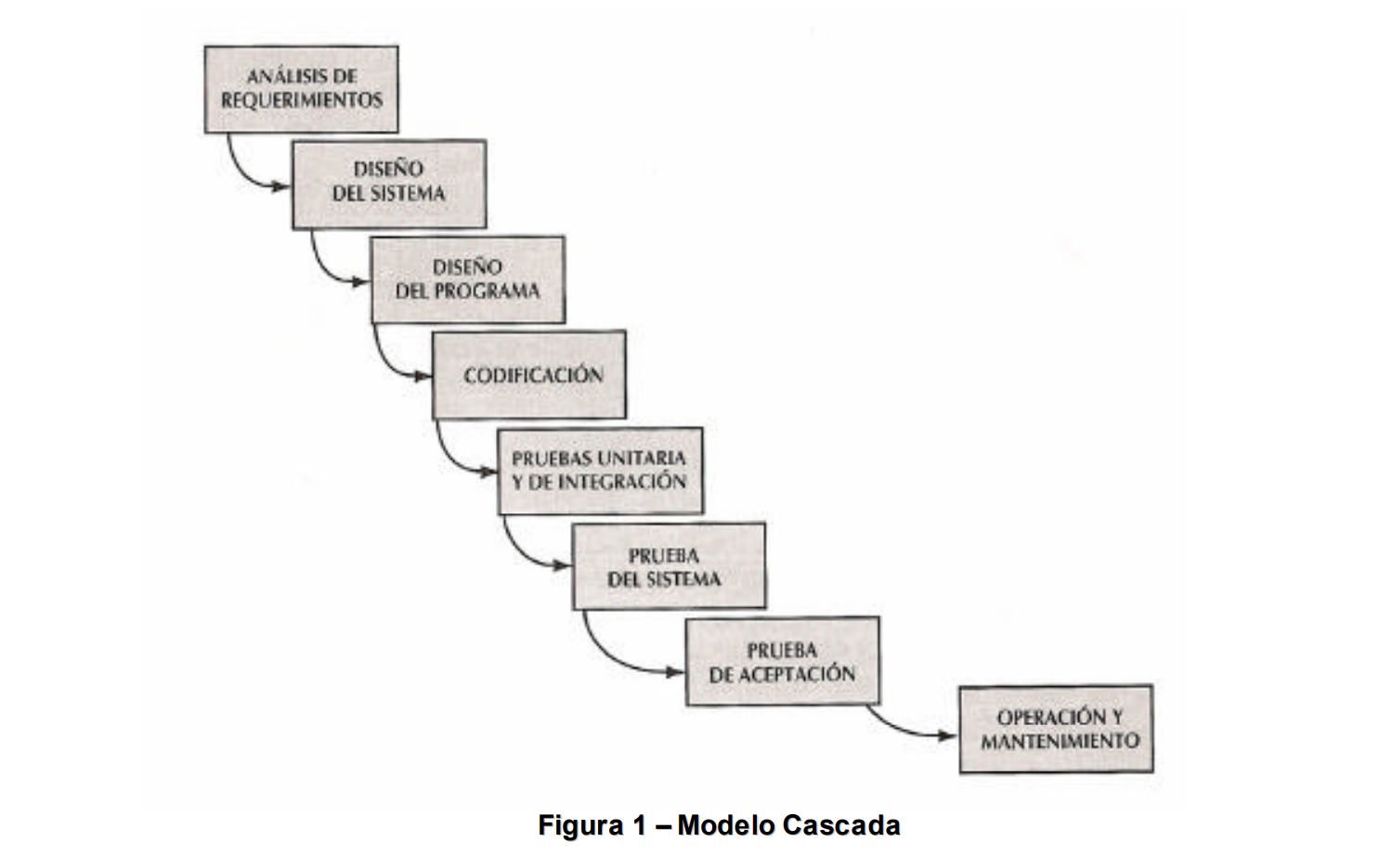
Modelos de ciclo de vida

Modelo en cascada

El modelo de ciclo de vida en cascada divide el desarrollo en etapas que deben completarse una tras otra. De esta manera, el equipo de desarrollo solo podrá comenzar a diseñar el sistema después de completarse la toma de requerimientos. Este modelo propone la forma en que deberían ocurrir las etapas del proyecto.

El modelo en cascada **con iteraciones** es una variante del modelo en cascada. El mismo se basa en iteraciones que ocurren durante una misma etapa o de una etapa hacia otra anterior. Se puede comenzar de nuevo después de finalizar el proyecto, dependiendo de la evaluación del sistema producido.

Las etapas sugeridas del modelo en cascada son las siguientes:



**Ingeniería y modelado de Sistemas**

Establece los requisitos que debe cumplir el software teniendo en cuenta la interconexión con otros sistemas.

**Análisis de los requisitos del software**

Se procesan los requisitos centrándose en el software a crear. Se crea la documentación en función de los requisitos. Para crear el software necesario para satisfacer sus necesidades, es necesario comprender el dominio del problema.

**Diseño**

El diseño se subdivide en cuatro pasos: estructurado de datos, arquitectura de software, representaciones de interfaz y proceso algorítmico. El diseño traduce requisitos en un sistema que debe ser evaluado por calidad antes de comenzar con la codificación.

**Codificación**

Al diseñarse el sistema en la etapa previa, la creación de código se realiza mecánicamente.

**Pruebas de integración**

Una vez creado el código, empieza las pruebas de este. Se debe verificar que el sistema cumple con los requisitos planteados, y corroborar la funcionalidad de este.

**Mantenimiento**

El sistema puede cambiar por corrección de errores, por entrega de nuevos requisitos o por cambios en el entorno externo. El mantenimiento implica volver a aplicar las etapas previas sobre el sistema existente.

Las limitaciones de este modelo son las siguientes:

1. El modelo es demasiado rígido para contemplar caminos alternativos cuando se cumple una etapa, dejando la posibilidad a que una o mas etapas no lleguen a completarse.
2. El modelo asume que todos los requisitos se definen antes de empezar el diseño y no tienen modificaciones posteriormente.
3. Cuando se definen los requisitos se selecciona el hardware que hará de soporte del sistema. Como consecuencia, el producto a su salida contara con hardware obsoleto.
4. El software no puede probarse hasta que el proyecto este avanzado, habiéndose consumido buena parte de los recursos. Debido a que el feedback no es constante un error grave puede complicar seriamente el proyecto.
5. La naturaleza lineal del modelo produce bloqueos ya que ciertos miembros requieren que otros miembros completen sus tareas para poder seguir con las suyas.
6. El modelo no especifica cómo el resultado de una etapa se transforma en uno nuevo. Por lo tanto, el modelo no proporciona instrucciones sobre cómo manejar los cambios que puede experimentar el producto en una etapa y cómo estos cambios afectan las etapas anteriores.
7. El modelo no tiene en cuenta el proceso evolutivo del software por lo que no comprende los procesos de avance y retroceso del mismo. Tampoco comprende el desarrollo a prueba y error de prototipo de hasta llegar a una solución.

Los beneficios del modelo en cascada son:

1. Etapas se organizan lógicamente. De esta manera, no se puede avanzar una etapa hasta que la anterior no se haya completado.
2. Cada etapa incluye un proceso de revisión de errores y la aceptación del resultado final. Como resultado, los errores que pasan de una etapa a otra deben ser mínimos.
3. El ciclo tiene un componente iterativo debido a que problemas encontrados en etapas inferiores afectan a las decisiones de las etapas superiores.

Modelo en V

Diagrama

Descripción generada automáticamente

Este modelo es una variación del modelo en cascada que se centra en que cada etapa tenga otra asociada referida a pruebas que, después de la etapa de codificación, verifiquen si el análisis de requerimientos, el diseño del sistema o el diseño del programa son correctos. Si las pruebas fallan, se redirige el ciclo hacia la etapa fallida para solucionar el problema antes de continuar con el proceso.

En este modelo, a diferencia del modelo en cascada, los errores y sus soluciones son representadas como parte del flujo de creación de software.

Las pruebas que sugiere este modelo son:

1. Prueba unitaria/integración: Valida el diseño del programa. Dirigida por el desarrollador.
2. Prueba del sistema. Valida el diseño del sistema. Dirigida por el desarrollador.
3. Prueba de aceptación: Valida que la toma de requerimientos sea correcta. Es dirigida por el cliente.

Modelo incremental

Diagrama

Descripción generada automáticamente

El modelo de ciclo de vida incremental se basa en ir construyendo el sistema gradualmente, agregando nuevas funcionalidades a la versión salida del incremento anterior. De esta manera, el sistema se construye hasta cumplir con las funcionalidades requeridas. Se aplican 4 etapas: análisis, diseño, codificación y pruebas.

El producto esencial o núcleo, que contiene los requisitos fundamentales que debe cumplir el sistema, es el primer paso de este modelo. Las versiones subsiguientes se desarrollarán en base a las mejoras y características solicitadas por el cliente a partir de la evaluación del núcleo.

Las ventajas de este modelo son:

1. Los primeros incrementos requieren de menos personas para llevarse a cabo ya que es un producto chico.
2. Mayor feedback del lado del cliente cada vez que se completa un incremento.

Modelo iterativo

El objetivo del modelo iterativo es entregar mejoras continuas del sistema a través del refinamiento. A diferencia del modelo incremental, el sistema creado por el modelo iterativo es funcional desde el principio y puede incorporar mejoras a funcionalidades ya desarrolladas.

El modelo iterativo se separa del modelo incremental en cuanto a que el primero incluye todas las funciones en una primera versión, pero dejando espacio a que estas sean mejoradas en la próxima iteración.

Modelo espiral