MÓDULO 2 MF0964_3: DESARROLLO DE ELEMENTOS DE SOFTWARE PARA GESTIÓN DE SISTEMAS

3 El ciclo de vida del software de gestión de sistemas

core networks

3 El ciclo de vida del software de gestión de sistemas

3.1 Introducción

core networks

Ciclo de vida del software

El término ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final. El propósito de este enfoque es definir las distintas fases intermedias que se requieren para validar el desarrollo de una aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo: se asegura que los métodos utilizados son apropiados.

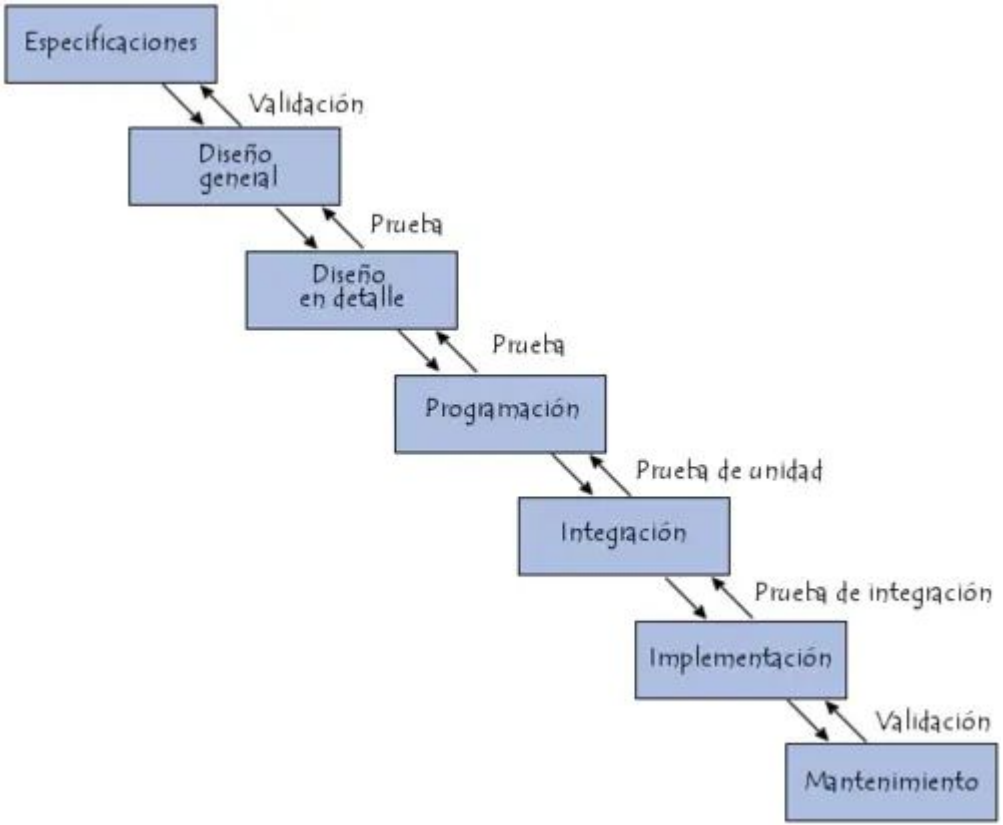
3 El ciclo de vida del software de gestión de sistemas

3.2 Modelos del ciclo de vida del software

core networks

Modelo en cascada

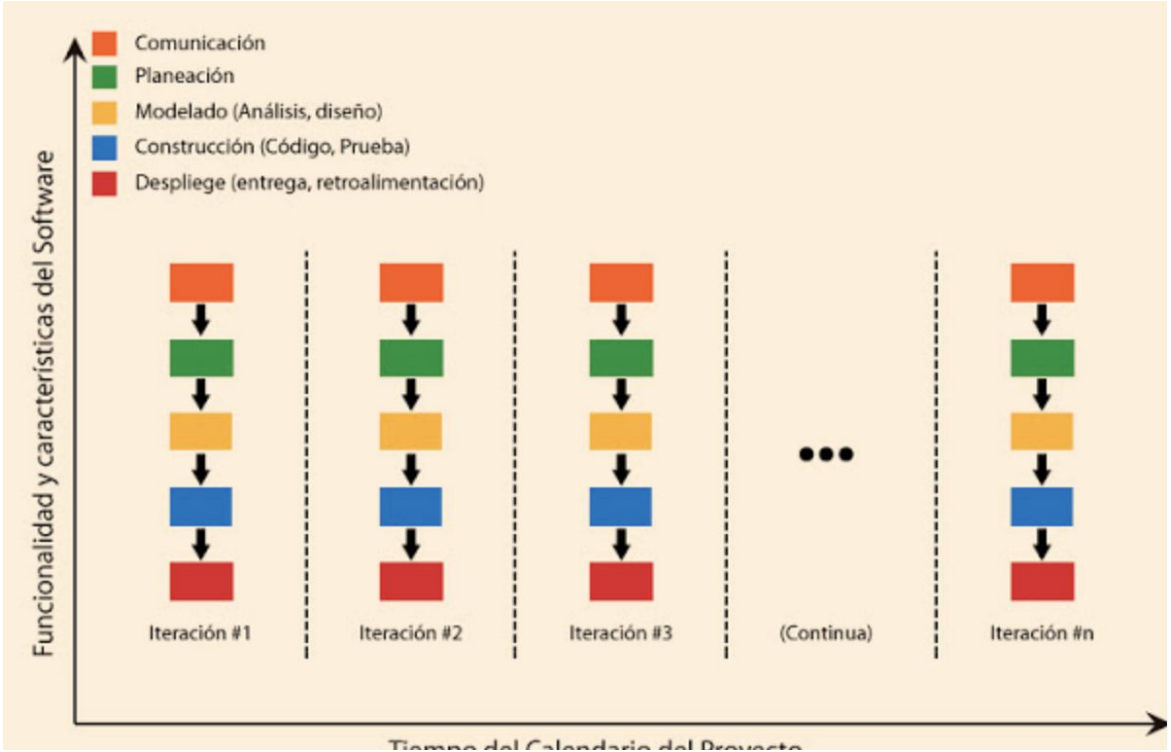
Se define como una secuencia de fases donde al final de cada una de ellas se reúne la documentación para garantizar que cumple las especificaciones y los requisitos antes de pasar a la fase siguiente:



core networks

Modelo iterativo

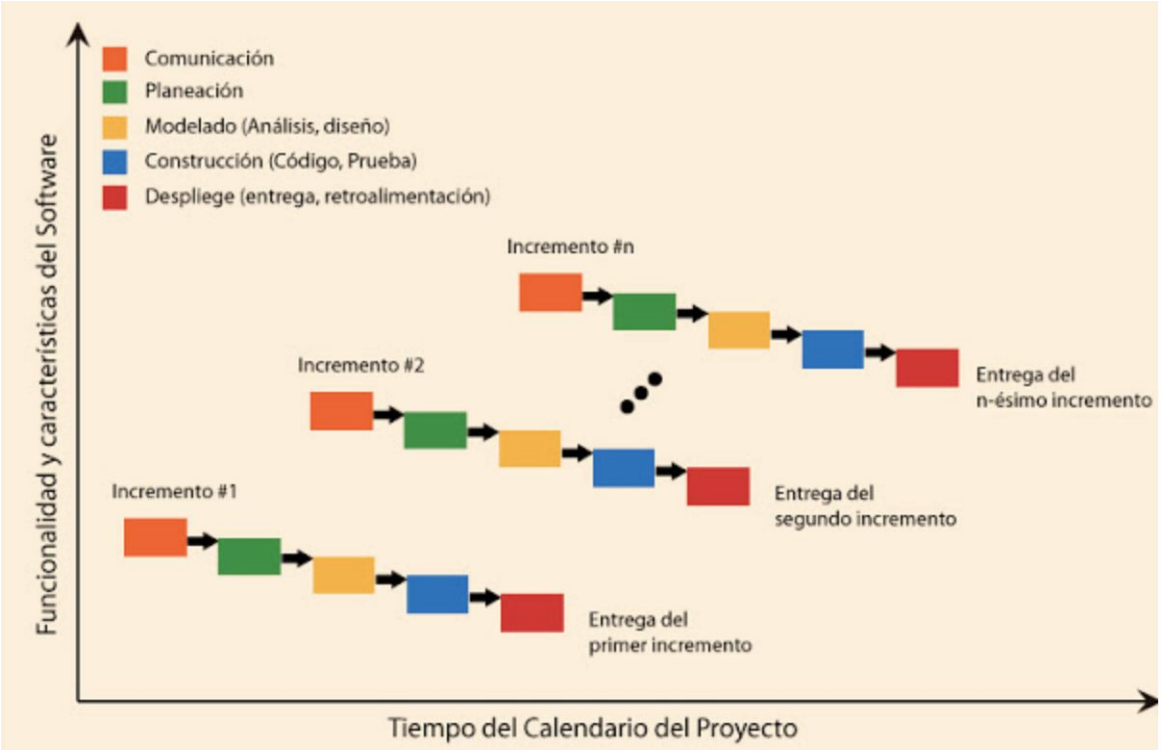
Este modelo se componen de iteraciones, en las cuales se produce la secuencia de etapas de un modelo en cascada clásico:



Modelo incremental

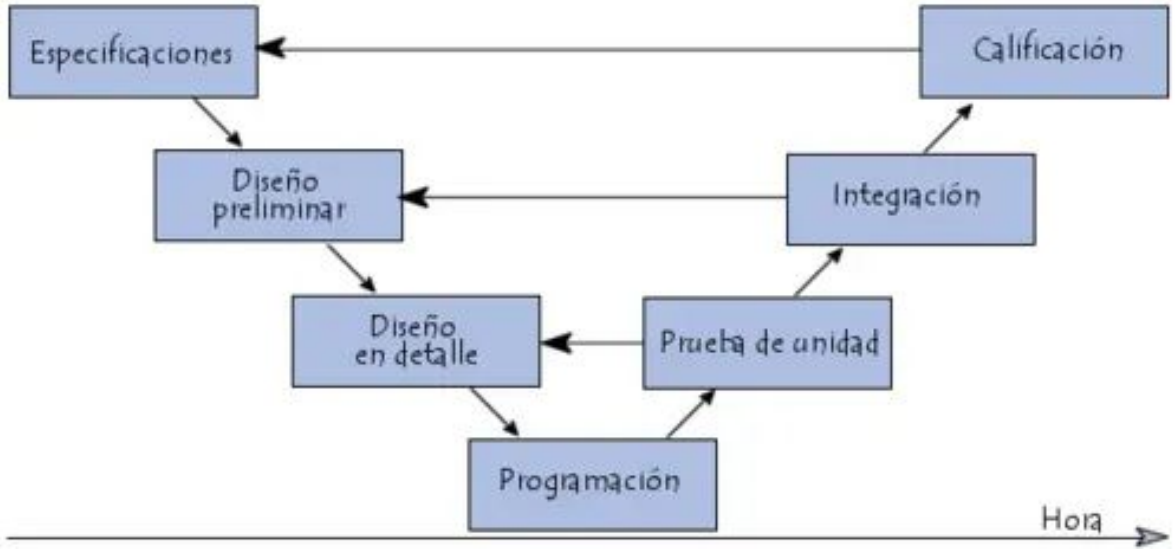
Se centra en desarrollar el sistema en partes, de forma que se van entregando a medida que se van completando.

Este modelo se caracteriza por entregar en cada iteración una parte de la funcionalidad del sistema.



Modelo en V

El modelo de ciclo de vida V proviene del principio que establece que los procedimientos utilizados para probar si la aplicación cumple las especificaciones ya deben haberse creado en la fase de diseño.

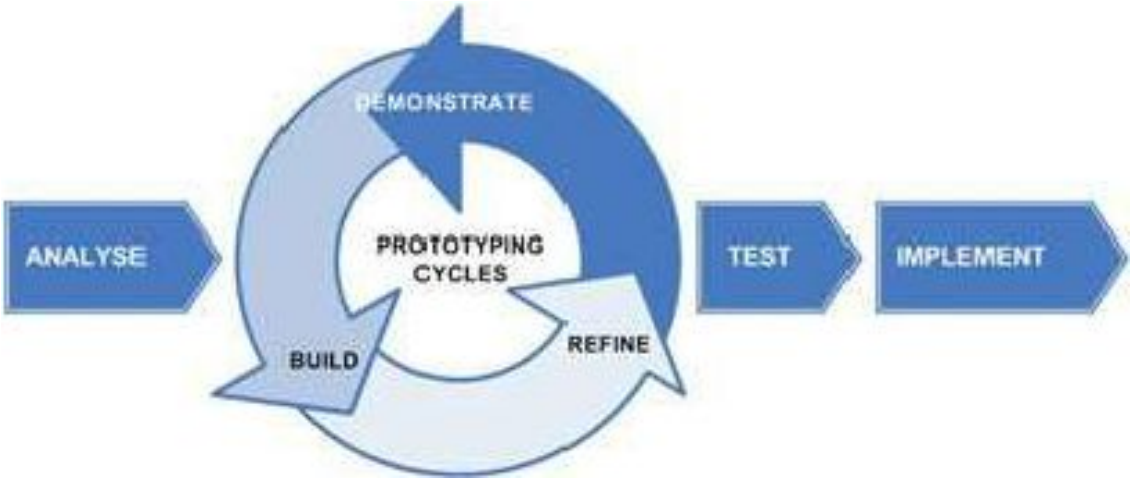


Modelo basado en componentes

El desarrollo de software basado en componentes permite reutilizar piezas de código preelaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión.

Modelo de desarrollo rápido

El modelo RAD tiene como objetivo minimizar el tiempo de desarrollo basado en la implementación de un prototipo, y a continuación, integrar la funcionalidad de forma iterativa para satisfacer las funcionalidades de negocio de la aplicación.



3 El ciclo de vida del software de gestión de sistemas

3.3 Descripción de las fases en el ciclo de vida del software

core networks

Análisis y especificaciones de requisitos

Tipos de requisitos.

- Ambiente físico (entornos o arquitectura de hardware)
- Interfaces
- Usuarios (historias de usuario)
- Funcionalidad (casos de uso)
- Datos
- Recursos
- Seguridad
- Documentación
- Rendimiento

Diseño

Tipos de diseño.

- Diseño de componentes
- Diseño de interfaz
- Diseño arquitectónico
- Diseño de datos

Implementación

En la propia definición de implementación se encuentra el significado de esta etapa, es decir, a partir de la etapa anterior se programan y despliegan los diseños definidos para completar la arquitectura de software y hardware de la aplicación.

En ocasiones esta etapa es denominada de desarrollo, denominándose en ese caso la implementación como la fase de despliegue y entrega de la característica desarrollada.

Validación, verificación y pruebas

- Validación. La validación es el conjunto de actividades que permiten asegurar que el software desarrollado cumple los requisitos de cliente.
- Verificación. Conjunto de actividades que permite asegurar si una funcionalidad está implementada correctamente y realiza lo que se espera de ella.

Tipos de pruebas o test

- Pruebas funcionales:
  - Pruebas unitarias.
  - Pruebas de aceptación.
  - Pruebas de integración.
  - Pruebas de regresión.

- Pruebas no funcionales:
  - Pruebas de carga.
  - Pruebas de estrés.
  - Pruebas de escalabilidad.
  - Pruebas de portabilidad.

3 El ciclo de vida del software de gestión de sistemas

3.4 Calidad del software

core networks

ISO9126

La Norma ISO/IEC 9126 es un estándar internacional para la evaluación del software que surge debido a la necesidad de un modelo único para expresar la calidad de un software. Fue publicado en 1992 con el nombre de “Information technology – Software product evaluation: Quality characteristics and guidelines for their use”, en el cual se establecen las características de calidad para productos de software.

Se basa en el uso de seis características básicas.

core networks

Funcionalidad

Esta característica permite calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado.

Tiene como atributos:

- Adecuación
- Exactitud
- Interoperabilidad
- Conformidad
- Seguridad

Confiabilidad

Se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.

Tiene como atributos:

- Nivel de madurez
- Tolerancia a fallos
- Recuperación

Usabilidad

Característica que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.

Tiene como atributos:

- Comprensibilidad
- Facilidad de Aprender
- Operabilidad

Eficiencia

Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados.

Tiene como atributos:

- Comportamiento con respecto al Tiempo
- Comportamiento con respecto a Recursos

Mantenibilidad

Aquí permite medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad.

Tiene como atributos:

- Capacidad de análisis
- Capacidad de modificación
- Estabilidad
- Facilidad de Prueba

Portabilidad

Se refiere a la habilidad del software de ser transferido de un ambiente a otro.

Tiene como atributos:

- Adaptabilidad
- Facilidad de Instalación
- Conformidad
- Capacidad de reemplazo

3 El ciclo de vida del software de gestión de sistemas

3.5 Resumen

core networks

Resumen

- Modelos de ciclo de vida de software
- Fases del ciclo de vida de software
- Calidad de software