

Ciclo de vida de desarrollo de software

DIIT

Departamento de Ingeniería e
Investigaciones Tecnológicas

DISEÑO DE APLICACIONES WEB/MÓVILES

Índice

Introducción

Diapositiva 03

Modelo en Cascada

Diapositiva 04

Modelo Iterativo o Incremental

Diapositiva 12

Modelo en Espiral

Diapositiva 16

Modelo Ágil

Diapositiva 22

Introducción

El ciclo de vida del desarrollo de software es un modelo aplicado al desarrollo de un PRODUCTO software.

Tiene como objetivo encontrar procesos sistemáticos, reproducibles y predecibles que mejoren la productividad y la calidad.

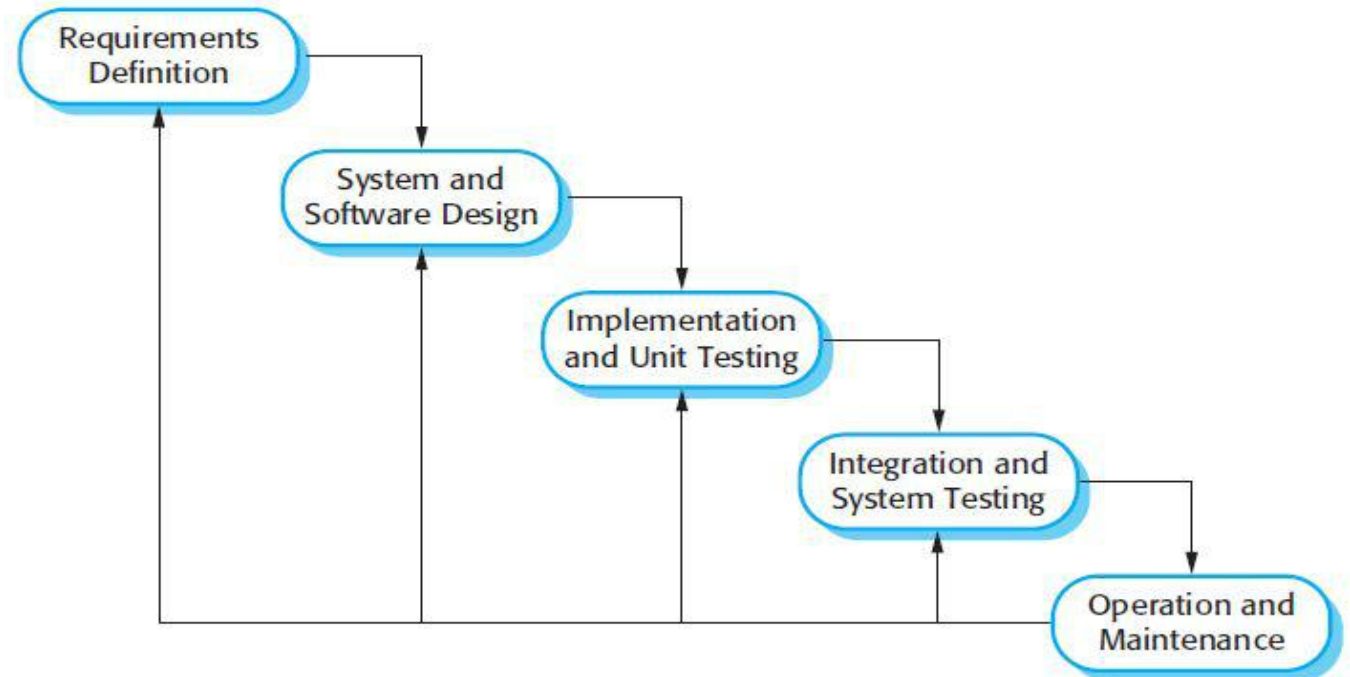
Hay varios modelos, cada uno describe un enfoque diferente para distintas actividades que tienen lugar durante todo el proceso.

Modelo en Cascada

Winston W. Royce en 1970.

Fue uno de los primeros modelos propuestos.

Una etapa de desarrollo debe completarse antes de dar comienzo a la siguiente.



Modelo en Cascada

Primera etapa: Definición de Requerimientos

Se realizan consultas a los usuarios y se confecciona un documento técnico con especificaciones del sistema.

Se definen Requerimientos Funcionales y No Funcionales.

Modelo en Cascada

Segunda etapa: Diseño del Sistema y del Software

Se establece una arquitectura completa del sistema.

En la etapa de diseño del software se identifican y describen las abstracciones del sistema de software y sus relaciones.

Modelo en Cascada

Tercera etapa: Implementación

Durante esta etapa se desarrolla el software y se realizan pruebas unitarias.

Modelo en Cascada

Cuarta etapa: Integración y prueba del sistema

Se buscan sistemáticamente y se corrigen todos los errores antes de ser entregado al usuario final.

Modelo en Cascada

Quinta etapa: Mantenimiento del sistema

El sistema ya se encuentra en funcionamiento.
Implica un sostenimiento adecuado para que el software funcione adecuadamente.

Modelo en Cascada

Ventajas del modelo

- Es un modelo fácil de implementar y entender.
- Está orientado a documentos.
- Promueve una metodología de trabajo sistemática:
Definir antes que diseñar, diseñar antes que codificar

Modelo en Cascada

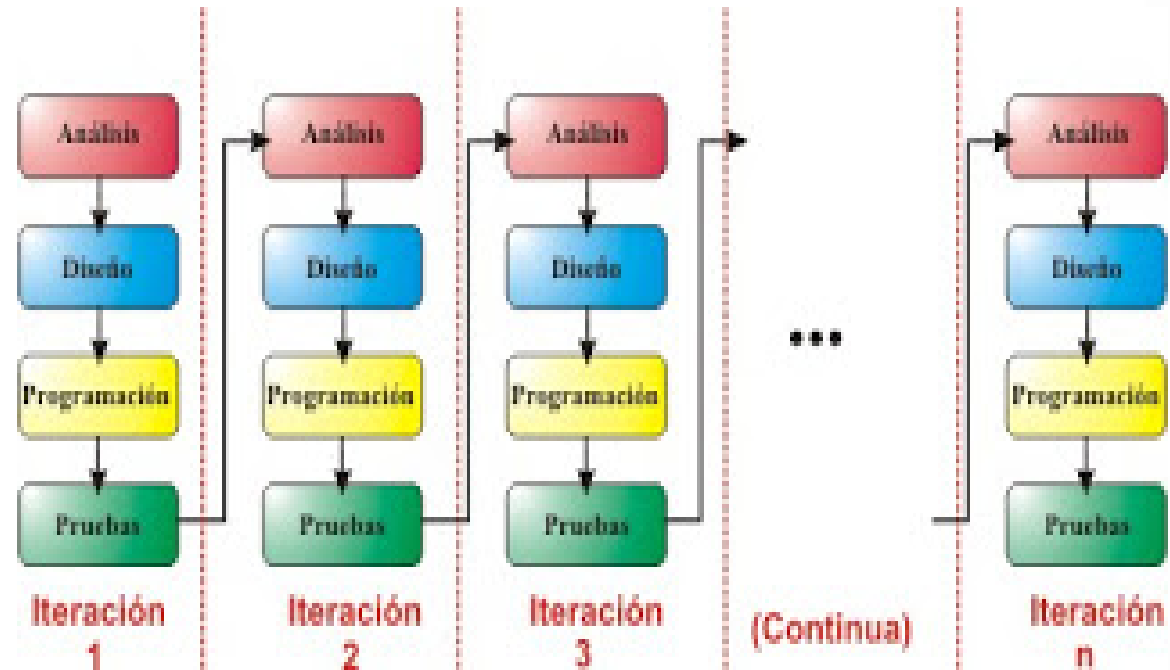
Desventajas del modelo

- Es rígido.
- Poco Flexible.
- Es costoso: cualquier error eleva el costo.
- Difícil de detectar errores.
- Se necesita tener en claro los requerimientos al comienzo del proyecto.

Modelo Iterativo o Incremental

Se desarrolla una implementación inicial reducida, y luego se irán agregando funcionalidades en cada iteración.

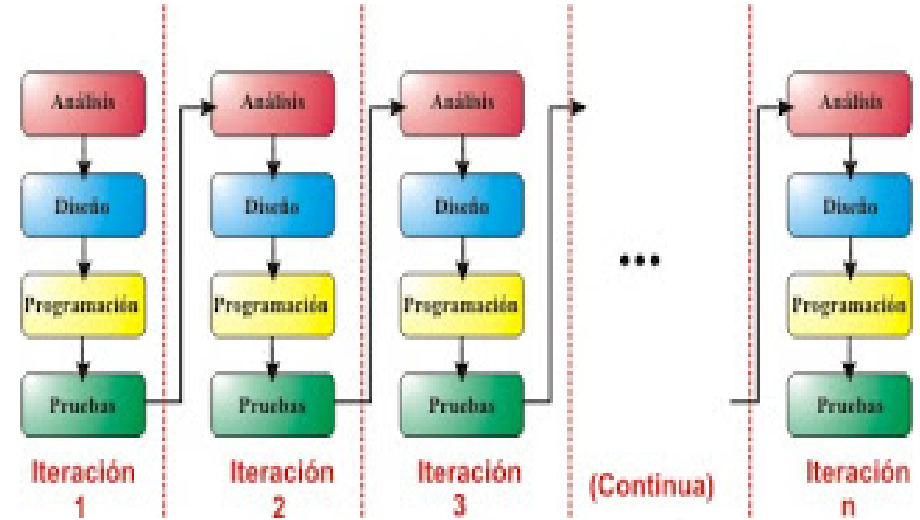
En una iteración se repite un determinado proceso de trabajo que brinda un resultado más completo para un producto final, de forma que quien lo utilice reciba beneficios de este proyecto de manera creciente.



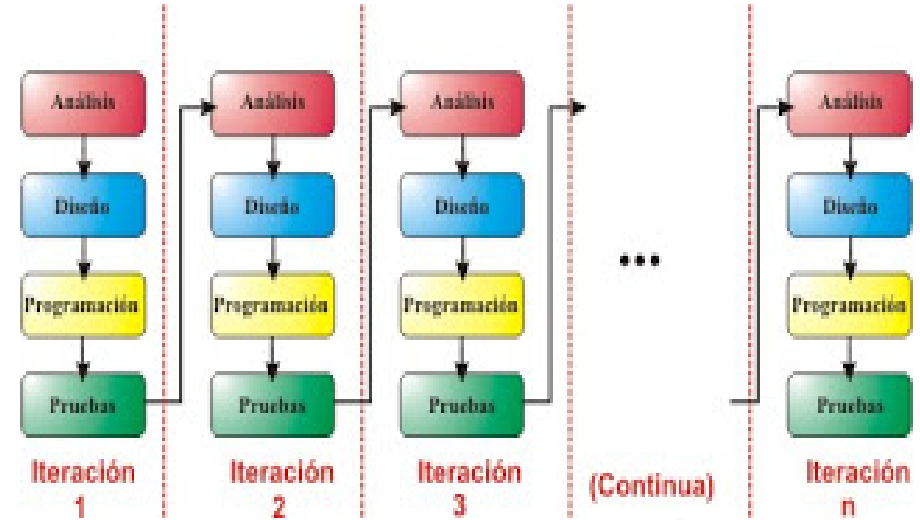
Modelo Iterativo o Incremental

Ventajas del Modelo Iterativo Incremental:

- Se reduce el tiempo de desarrollo inicial ya que se implementa una funcionalidad parcial.
- Si se detecta un error, se desecha la última iteración.
- No es necesario tener en claro TODOS los requerimientos al comienzo del proyecto.



Modelo Iterativo o Incremental



Desventajas del Modelo Iterativo Incremental:

- La entrega temprana de los proyectos produce la creación de sistemas demasiados simples a primera impresión del cliente.
- Requiere de un cliente involucrado durante todo el curso del proyecto.
- La entrega de un programa que es parcial pero funcional puede hacer vulnerable al programa debido a la falta de robustez en su sistema.

Modelo En espiral

Propuesto originalmente por BOEHM en 1986.

Proporciona el potencial para el desarrollo rápido de versiones incrementales.

El software se desarrolla en una serie de versiones incrementales.

Se divide en regiones de tareas, las cuales están compuestas por un conjunto de tareas.

Modelo en Espiral



Modelo en Espiral

Determinar de objetivos

Se especifica el alcance que el producto software alcanzará al final de la iteración.

Análisis de riesgos

Se analizan todos los riesgos potenciales que puedan llegar a afectar el desarrollo del proyecto en el tiempo establecido.

Modelo en Espiral

Desarrollar y probar

Comprende las tareas de: análisis, diseño, construcción e implantación del incremento del producto software.

Planificación

Se realiza un estudio de la situación actual del producto software analizando si es necesario o no realizar un nuevo incremento. En caso de ser necesario, se planifica la próxima iteración.

Modelo en Espiral

El gráfico de la espiral, el cual aumenta su amplitud con cada incremento, indica que se construyen sucesivas versiones del producto software.

El modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del producto software.

Una visión alternativa del modelo en espiral puede ser considerada examinando el eje de punto de entrada en el proyecto.

Modelo en Espiral

Ventajas del modelo en espiral

- Para el desarrollo de grandes proyectos, es el mejor modelo.
- Es combinable con otros modelos de desarrollo.
- Se puede implementar en un proyecto con altos grados de incertidumbre.
- El riesgo de detección de errores es bajo.
- Cada error se soluciona en el área correspondiente.

Modelo en Espiral

Desventajas del modelo en espiral

- Se suma un costo temporal por cada vuelta.
- La comunicación con el usuario debe ser muy fluida.
- En algunos casos es costoso realizar el análisis de riesgos.
- Es complicado planificar el proyecto de manera global, ya que, esto dependerá de la evolución o incrementos del producto software.
- Existe una cierta incertidumbre en cuanto hasta dónde se podrá llegar con el presupuesto inicial.

Modelo Ágil

El desarrollo ágil de software utiliza un desarrollo iterativo como base.

Se basa en un punto de vista más ligero y centrado en las personas que las soluciones tradicionales.

Los modelos ágiles utilizan retroalimentación en lugar de planificación, como principal mecanismo de control.

Metodología Ágil SCRUM

SCRUM es una de las metodologías ágiles más populares y utilizadas actualmente en el desarrollo de software.

Permite trabajar colaborativamente en equipos altamente productivos.

Se realizan entregas de software de manera incremental, en tiempos fijos, mediante sprints.

En cada sprint se entrega un incremento de Producto (versión mejorada de la entrega anterior).

Metodología Ágil SCRUM

Características Principales de Scrum

Flexibilidad y adaptación, productividad y calidad, alineamiento entre cliente y equipo, Equipo motivado.

Equipos auto-dirigidos y auto-organizados.

Se compone principalmente por:

- roles
- artefactos
- eventos

Metodología Ágil SCRUM

Roles

Product Owner:

Es la representación del cliente dentro del equipo de trabajo, su principal responsabilidad es la de expresar claramente la necesidad del cliente dentro del Product Backlog

Metodología Ágil SCRUM

Roles

Dev Team:

Se compone de las personas responsables de desarrollar el producto.
Es un equipo multidisciplinario, autogestionado y organizado.

Metodología Ágil SCRUM

Roles

Scrum Master:

Es el responsable de asegurar que el SCRUM es entendido y realizado por el equipo ajustandose a la teoría, prácticas y reglas de scrum

Metodología Ágil SCRUM

Artefactos

Product Backlog:

*Es una lista de requerimientos priorizadas, que es administrada por el Product Owner.
Evoluciona a lo largo del proyecto.*

Sprint backlog:

*Lista de requerimientos para un Sprint. Es administrada por Dev. Team.
Solo el team puede modificarlo.*

Metodología Ágil SCRUM

Eventos

Sprint Planning:

- *Se define el objetivo del Sprint.*
- *Product Owner presenta requerimientos.*
- *Team estima esfuerzo.*
- *Se genera Sprint Backlog.*
- *Asisten: Scrum Master – Dev Team - Product Owner.*

Metodología Ágil SCRUM

Eventos

Daily Meeting:

- *Duración: 15 minutos.*
- *Mismo lugar y hora.*
- *¿Qué hice ayer? ¿Qué voy a hacer hoy? ¿Qué impedimentos tengo?*
- *Team actualiza Sprint Backlog.*
- *Asisten: Dev Team, Scrum Master. El resto es opcional.*

Metodología Ágil SCRUM

Eventos

Sprint Review:

- *Duración: De 2 a 4 horas.*
- *Se presenta incremento producto.*
- *Informal, informativa, se obtiene feedback.*
- *Asisten: Todos.*

Metodología Ágil SCRUM

Eventos

Sprint Retrospective:

- *Duración: De 1 a 2 horas.*
- *Revisión de proceso de trabajo, identificar oportunidades de mejora para el próximo sprint.*
- *Asisten: Dev Team, Scrum Master y Product Owner.*

Metodología Ágil SCRUM

