

Ciclos de Vida

Definiciones

- "Aproximación lógica a la adquisición, suministro, desarrollo, explotación y mantenimiento del software"
- "Marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso"

Características

Un ciclo de vida debe determinar el **orden de las fases** del Proceso de Software y establecer **criterios de transición** para pasar de una fase a la siguiente.

Ciclo de Vida vs ciclo de Desarrollo

Ciclo de vida \neq Ciclo de vida de desarrollo

- Ciclo de vida: Toda la vida del sistema, desde la concepción hasta el fin de uso.
- Ciclo de vida de desarrollo: Desde el análisis hasta la entrega al usuario.

Codificación directa

- También llamado **Code and Fix**
- Sinónimo de **arte** en el Desarrollo de Software
- Antónimo de ingeniería en el Desarrollo de Software
- **No es recomendable** para proyectos poco más grande que "enanos"

Cascada

También conocido como **Waterfall**, nace en los años 70 como alternativa al Code and Fix, que provocó la crisis del software. Sigue un gran número de metodologías y es **el más conocido, estudiado, difundido y empleado**. Pone especial énfasis en la **realización temprana de actividades** de definición de los requisitos y de la documentación de análisis y diseño como paso previo a la Codificación. **Secuencia lineal** de las siguientes actividades generales: Analizar, Diseñar, Codificar, Probar e Implementar. Cada fase genera productos de salida que servirán a la siguiente fase para desarrollar la actividad de la misma como productos de entrada.

Ventajas

- Conforman un **marco de referencia** para asignar todas las actividades de desarrollo software.
- Es un método muy estructurado que establece pautas de trabajo muy claras.
- Facilita mucho la **coordinación** de los recursos implicados.
- Facilita la **disposición de hitos de seguimiento / control** en el desarrollo de proyectos.
- Facilita la **estimación y el seguimiento / control** del progreso de las actividades.
- Facilita la **detección de desviaciones** y la realización de acciones correctivas.

- Proporciona productos entregables intermedios que conforman el producto final.

Inconvenientes

- Comienza estableciendo **todos** los requisitos del sistema a desarrollar, aunque muchas veces esto no es viable.
- **Gran rigidez**: cada actividad es prerequisite de las que le siguen.
- Los posibles **problemas se detectan tarde**, aunque se incluyan actividades de verificación y validación.
- Los únicos productos parciales aprovechables están en forma de documentos: **nada está hecho hasta que todo está hecho**.

Otros modelos de Ciclo de Vida

Ciclo de vida en V

Es un modelo similar al de Cascada, aunque "mejorado". Tiene dos tiempos claramente marcados: **rama descendente**, actividades de desarrollo; **rama ascendente**, actividades de prueba. Ambas ramas se comunican en materia de pruebas, cada fase de la rama descendente tiene su homóloga en la rama ascendente para las actividades de prueba correspondientes.

Ventajas

- Conformar un marco de referencia para asignar todas las actividades de desarrollo software, incluyendo las actividades V&V de lo que se hace en las sucesivas etapas.
- Favorece la consideración de las pruebas lo antes posible (comunicación horizontal entre las dos ramas de la V)

Inconvenientes

- Al igual que en Cascada, se comienza estableciendo todos los requisitos del sistema a desarrollar.
- Posee una gran rigidez, aunque menos que el de Cascada por haber comunicación horizontal.
- Los únicos productos (parciales) aprovechables están en forma de documentos.

Prototipado

El proceso básico del prototipado involucra las siguientes tres fases en bucle:

- Escuchar al usuario/cliente.
- Construir y revisar el prototipo.
- El usuario/cliente prueba el prototipo Su sentido es análogo al que tienen otras ingenierías, el prototipado o construcción de un prototipo es un proceso que facilita al desarrollador la creación de un modelo del software a desarrollar.

Tipos de prototipos

Desechables: nos sirve para eliminar dudas sobre lo que realmente quiere el cliente, además de desarrollar la interfaz que más le convenga al cliente. Evolutivos: es un modelo parcialmente construido que puede

pasar de ser prototipo a ser software , pero no tiene una buena documentación y calidad, hay que conseguir que tenga los criterios mínimos de calidad.

Ventajas

El prototipo es un mecanismo ideal para extraer requisitos cuando no están claros, incluso por parte del usuario.

Inconvenientes

Existe una clara tendencia del usuario/cliente a creer que el trabajo ya está hecho y que en breve dispondrá de un sistema funcional. El desarrollador toma necesariamente decisiones de implementación simplemente porque le son conocidas y le permiten desarrollar rápidamente el prototipo.

Incremental

Es el proceso de construir una implementación parcial del sistema global y posteriormente ir aumentando la funcionabilidad del sistema. Es la aplicación reiterada de varias secuencias basadas en el modelo Cascada. Cada aplicación del ciclo constituye un incremento del software, cada incremento resulta en un producto operativo, cada incremento puede ser: el refinamiento de un incremento anterior o el desarrollo de una nueva funcionalidad no incorporada en incrementos anteriores; en el primer incremento de un sistema se debe abordar el núcleo esencial del sistema, en incrementos posteriores se abordarán funciones suplementarias. El usuario/cliente evalúa el resultado de un incremento y se elabora un plan para el incremento siguiente. El proceso se repite hasta que se elabore un producto completo. Al seguir un desarrollo incremental, el software debe construirse de tal forma que facilite la incorporación de nuevos requisitos.

Ventajas

La dotación de personal no tiene que estar disponible para una implementación completa. Permite la obtención de incrementos operativos a lo largo del proceso de desarrollo, frente a modelos tradicionales basados en el de Cascada. Reduce la posibilidad de que los requisitos del usuario/cliente cambien durante el desarrollo. Mayor flexibilidad ante más funcionalidades. La calidad y estabilidad del software progresan con las iteraciones, y hay oportunidad para la mejora. Ayuda en proyectos que utilizan tecnologías nuevas o en fase de aprendizaje por parte de la organización.

Incremental

Puede ser muy complejo definir el núcleo operativo para lograr el primer incremento: un software con capacidades suficientes como para ser operativo y que facilite la incorporación de nuevos requisitos. Puede resultar complicado establecer y priorizar las funcionalidades que se mejoran o incorporan a los sucesivos incrementos. Las soluciones de incrementos anteriores pueden no ser válidas para incrementos posteriores. No es útil en proyectos cortos en duración o ámbito ni en los que la funcionalidad a implementar esté perfectamente definida, ya que el coste extra de planificación no merece la pena.

Espiral

Propuesto por Barry W. Boehm en 1988, el modelo Espiral de proceso evolutivo permite combinar la naturaleza interactiva de construcción de propositos con los aspectos controlados y sistemáticos del modelo en Cascada, además pone especial énfasis en el análisis de los riesgos para reducir su impacto. El software

se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental será un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema. La idea básica es que cada fase (vuelta de espiral) establece los siguientes pasos:

- Determinación de objetivos, alternativas y restricciones.
- Evaluación de alternativas.
- Desarrollo del siguiente nivel de producto.
- Planificación de la siguiente fase.

Ventajas

Permite adaptar el proceso de desarrollo a las necesidades cambiantes del proyecto y al conocimiento que se va adquiriendo. Permite el manejo de prototipos , enlazándolo con el análisis de riesgos. Gestiona explícitamente los riesgos, permitirá reducirlos antes de que se conviertan en problemas.

Inconvenientes

Requiere de una considerable habilidad para la consideración del riesgo.

Ventajas

Permite adaptar el proceso de desarrollo a las necesidades cambiantes del proyecto y al conocimiento que se va adquiriendo. Permite el manejo de prototipos , enlazándolo con el análisis de riesgos. Gestiona explícitamente los riesgos, permitirá reducirlos antes de que se conviertan en problemas.

Inconvenientes

Requiere de una considerable habilidad para la consideración del riesgo.