Guía de Ciclos de Vida

# Hoja de revisiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Fecha** | **Resumen de cambios** | **Autor** | **Aprobación** |
| 1.0 | 08/10/2014 |  | José Luis Medellín Fuentes | Comité |

**Contenido**

Hoja de revisiones 2

Introducción 4

1. Propósito 4

2. Alcance 4

Criterios por Tipo de Proyecto 4

Descripciones de ciclos de vida 5

1. Cascada 5

2. Desarrollo Iterativo 7

Ciclo de Vida Iterativo Incremental 7

Ciclo de Vida Iterativo Evolutivo 8

Criterios de Uso 10

# 

# Introducción

Este documento es una guía para seleccionar el *Modelo de Ciclo de Vida* que se adoptará para el desarrollo de un Proyecto de Software.

La elección del ciclo de vida es una de las actividades más importantes que se efectúan al iniciar el proyecto, debido a que se debe adoptar desde las fases iniciales del mismo.

La ingeniería de software centra su atención en los *Procesos* que intervienen en el desarrollo de un producto de software, pues son estos quienes al incorporar las necesidades y expectativas del usuario, definen un conjunto de actividades, métodos y herramientas dentro de un marco de trabajo para generar un producto o servicio, que deberá satisfacer las expectativas y requerimientos del usuario, así como las restricciones de tiempo, costo y calidad negociadas con el cliente.

La selección de un ciclo de vida debe considerar los siguientes aspectos:

* Ser el adecuado para los recursos tanto humanos como materiales con los que se cuenta.
* Permitir cumplir con los compromisos de tiempo, costo y calidad establecidos entre **LA ORGANIZACIÓN** y el **Cliente**.
* Garantizar la calidad de los componentes que se produzcan.
* Que los participantes en el proyecto entiendan ampliamente el ciclo de vida a emplear.

La construcción del ciclo de vida debe hacerse a partir de entender las necesidades del cliente y el diseño conceptual, durante la fase de planeación y considerar las restricciones del proyecto.

# Propósito

Facilitar la selección del Modelo de Ciclo de Vida a emplear de acuerdo a las características del proyecto, para poder optimizar los recursos y cumplir con las restricciones de tiempo, costo y calidad de los proyectos de desarrollo de software.

# Alcance

Describir los criterios necesarios para la selección del Modelo de Ciclo de Vida de Software a emplear por todos los Proyectos Requisitos del Proceso.

# Criterios por Tipo de Proyecto

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Categoría | Tipo de proyecto | Subtipo | Criterios | | | | Comentarios |
| Estrategia | Tiempo | | Plataformas |
| Límite inferior | Límite superior |
| **Diseño y Desarrollo de soluciones** | ***NUEVO DESARROLLO*** *El entregable representa un* ***nuevo proyecto*** *o un* ***nuevo conjunto*** *de requisitos de un proyecto existente* | *De Gobierno* | *Los entregables pueden ser adaptados tanto de la metodología de ND como de MAAGTICSI* | *N/A* | *N/A* | *J2EE, .NEt* | *Para que un proyecto caiga en alguna de estas categorías, es necesario que ND tenga el control de su alcance, de sus riesgos y administración en general.* |
| *General* | *El proyecto se ajusta a la metodología de ND y/o a la metodología del cliente, en caso de así requerirlo* |
| ***MANTENIMIENTO*** *El entregable representa un* ***cambio o mejora*** *a un conjunto de componentes que ya están en un entorno productivo* | *N/A* | *N/A* | *N/A* | *N/A* | *J2EE, .NET* |
| ***INCIDENCIA Errores o fallas*** *de un componente o conjunto de componentes que ya están en entorno productivo y que deben ser corregidos a la brevedad.También pueden considerarse cambios pequeños que no impactan la funcionalidad* | *N/A* | *La corrección al componente es prioritaria para no afectar el entonrno productivo.* | *N/A* | *N/A* | *J2EE, .NET* |

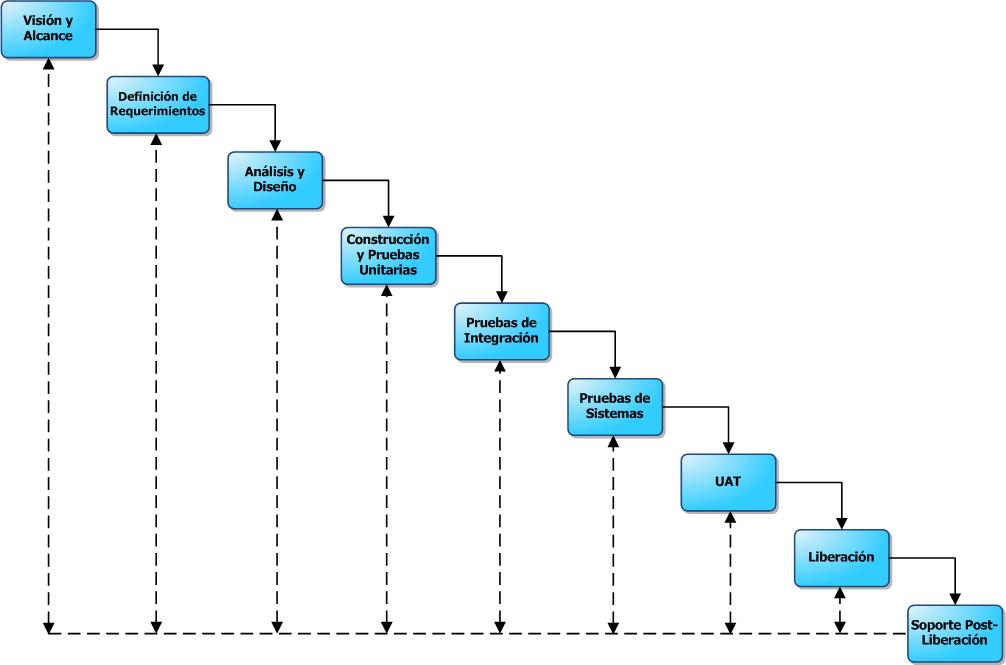
# Descripciones de ciclos de vida

Los *Modelos de Ciclo de Vida* autorizados para su uso en los *Proyectos*  son:

* Modelo de Cascada.
* Modelo Iterativo.
  + Iterativo Incremental.
  + Iterativo Evolutivo.

# Cascada

Este modelo de ciclo de vida fue propuesto por Winston Royce en el año 1970. Es un ciclo de vida que admite iteraciones. Después de cada etapa se realiza una o varias revisiones para comprobar si se puede pasar a la siguiente. Es un modelo rígido, poco flexible, y con muchas restricciones. Aunque fue uno de los primeros, y sirvió de base para el resto de los modelos de ciclo de vida.



Una de las **ventajas**, además de su planificación sencilla, es la de proveer un producto con un elevado grado de calidad sin necesidad de un personal altamente calificado. Se pueden considerar como inconvenientes: la necesidad de contar con todos los requerimientos (o la mayoría) al comienzo del proyecto y si se han cometido errores y no se detectan en la etapa inmediata siguiente, es costoso y difícil volver atrás para realizar la corrección posterior.

Además, los resultados los veremos hasta que no estemos en etapas finales del ciclo, por lo que, cualquier error detectado nos trae retraso y aumenta el costo del desarrollo en función del tiempo que insume la corrección de éstos.

Es un ciclo de vida adecuadopara los proyectos en los que se dispone de todos los requerimientos al comienzo, para el desarrollo de un producto con funcionalidades conocidas o para proyectos, que aun siendo muy complejos, se entienden perfectamente desde el principio.

Problemas del Modelo de Cascada.

* No es aplicable a productos de software altamente iterativos, aunque el modelo puede acoplar iteración y lo hace indirectamente.
* Los requerimientos obtenidos no pueden ser cambiados durante el proceso. Es difícil tener todos los requerimientos bien definidos al principio, como lo requiere el modelo y además presenta dificultades para acomodar posibles incertidumbres existentes al comienzo de los proyectos.
* Los productos de software raramente siguen el flujo secuencial que propone el modelo. Siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
* Un error importante no detectado al principio puede ser desastroso, no sólo al final del proyecto sino durante todo el proceso.
* Se requiere mucha paciencia por parte del cliente, porque solo hasta las etapas finales del desarrollo podrá tener una versión operativa del producto.
* Los responsables del desarrollo del software siempre se retrasan innecesariamente. La naturaleza lineal lleva a estados de bloqueo, en el que algunos miembros del equipo deben esperar a otros miembros para completar tareas pendientes, lo que aumenta el tiempo invertido en el proyecto.
* El tiempo invertido en pasar a través de todo el ciclo es demasiado.
* Las revisiones de proyectos de gran complejidad son muy difíciles.
* Es inflexible y se dificulta el proceso de estimación del proyecto completo.

Consideraciones del Ciclo de Vida de Cascada

El ciclo de vida de cascada debe ser usado tomando en cuenta las siguientes consideraciones:

* Es muy fácil de implementar.
* Se recomienda utilizar para sistemas donde los requerimientos tienen un alto grado de estabilidad.
* Se recomienda usar en proyectos de software que no son de mantenimiento o versiones subsecuentes de proyectos anteriores.
* Se puede utilizar si el sistema no requiere iteraciones para su desarrollo.

# Desarrollo Iterativo

El Modelo de Ciclo de Vida Iterativo organiza un conjunto de fases en iteraciones, al término de cada iteración se cuenta con una versión mejorada o con mayores funcionalidades del producto. El usuario es quien luego de cada iteración, evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga al usuario. El modelo Iterativo considera dos: *Incremental y el Evolutivo*.

### Ciclo de Vida Iterativo Incremental

El ciclo de vida Iterativo Incremental se centra en primer lugar en determinar las necesidades del usuario y definir los requerimientos del producto, para continuar entonces con el resto del desarrollo a través de una secuencia de iteraciones sucesivas, incrementando la funcionalidad del producto de iteración a iteración hasta obtener la funcionalidad completa. La aplicación de esta estrategia requiere que las necesidades del cliente se encuentren *muy bien comprendidas desde el inicio del ciclo del proyecto*, en consecuencia los requerimientos del producto pueden ser definidos claramente.

El desarrollo incremental es el proceso de construcción de implementaciones parciales que van agregando funcionalidad a través de un conjunto de iteraciones sucesivas. Con cada iteración se entrega un producto totalmente operacional.

Los requerimientos son agrupados en bloques y son desarrollados como paquetes independientes a través de todas las fases del ciclo de vida. Para cada bloque de requerimientos se realiza una iteración del proceso completo. Los bloques deben ser funcionalmente útiles para el cliente y son entregados en etapas refinadas sucesivamente.

Se establecen hitos al final de cada fase como metas intermedias para cada iteración. Se realizan revisiones en cada hito para determinar si está preparado para pasar a la siguiente fase, si la revisión determina que no se está listo se permanece en la fase hasta que se esté preparado. Existen entregas parciales pero funcionales en cada iteración al finalizar cada bloque de requerimientos.

La filosofía del modelo de desarrollo incremental considera los siguientes puntos:

* Construir un sistema pequeño es siempre menos riesgoso que construir un sistema grande. Al ir desarrollando parte de las funcionalidades, es más fácil determinar si los requerimientos planeados para los niveles siguientes son correctos.
* Se evitan proyectos largos y se entrega “*Algo de valor*” a los usuarios con cierta frecuencia.
* Se involucra el usuario.
* Si un error importante es identificado, sólo la última iteración necesita ser descartada y el incremento previo puede ser usado.
* Los errores de desarrollo realizados en un incremento, pueden ser arreglados antes del comienzo del próximo incremento.

El desarrollo incremental es en esencia la repetición en serie del modelo en cascada, por lo que es compatible con éste modelo. Así, el modelo cascada puede ser usado para administrar cada esfuerzo de desarrollo.

En la siguiente figura se puede apreciar la estructura en secuencia de las fases del Ciclo de Vida *Iterativo Incrementa*l.



**Consideraciones del Ciclo de Vida Iterativo Incremental**

El Iterativo Incremental es apropiado cuando:

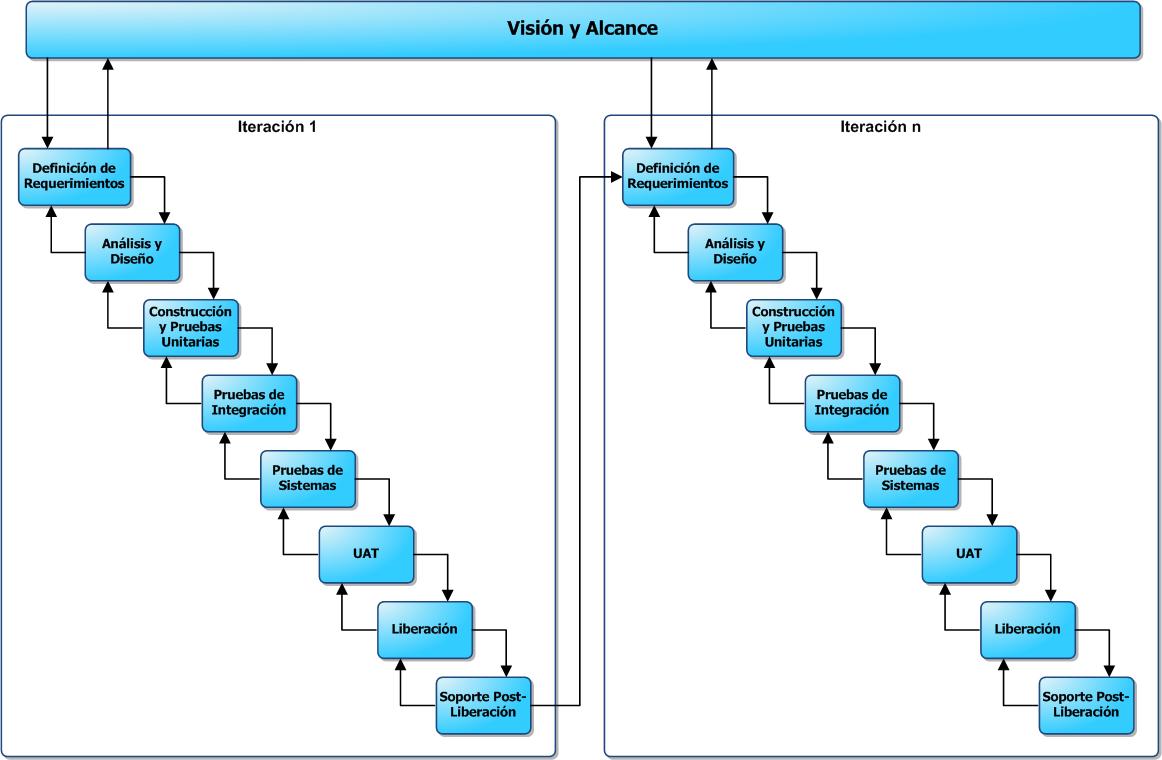
* Todos los requerimientos del sistema son conocidos al inicio del ciclo de desarrollo.
* Los requerimientos y la arquitectura puede estabilizarse en las fases tempranas del ciclo de desarrollo.
* El equipo del proyecto cuenta con experiencia en la temática del problema a solucionar.
* Las liberaciones incrementales de funcionalidad tienen gran valor para el cliente.
* El desarrollo del sistema es muy complejo y se pueden tomar segmentos de requerimientos o clasificarlos por módulos para construir un incremento.
* Existe un alto grado de derivación de los requerimientos (dependencia).

### Ciclo de Vida Iterativo Evolutivo

El Ciclo de Vida Iterativo Evolutivo se diferencia de la incremental bajo el hecho de que las necesidades del usuario no se encuentran completamente comprendidas al inicio del proyecto o son variables, en consecuencia los requerimientos del producto no se encuentran definidos claramente, por lo cual es necesario definirlos en iteraciones sucesivas.

Los requerimientos son cuidadosamente estudiados y evaluados, sólo aquellos que se encuentran bien comprendidos se seleccionan para la primera iteración, desarrollando una implementación parcial del producto basada en los requerimientos seleccionados. Se construye la primera versión del producto, el cual se entrega a los usuarios, quienes a su vez lo utilizan y proveen la retroalimentación correspondiente. Una vez recibida la retroalimentación se actualiza la especificación de los requerimientos, y una segunda versión del producto es desarrollada a través de una segunda iteración. El proceso se repite de sucesivamente para mejorar lo ya realizado.

En la siguiente figura se puede apreciar la estructura en secuencia de las fases del Ciclo de Vida *Iterativo Evolutivo*.



**Desarrollo Iterativo Evolutivo**

Mientras que la aproximación incremental presupone que el conjunto completo de requerimientos es conocido al inicio del proyecto, el modelo evolutivo asume que los requerimientos no son completamente conocidos al inicio o son variables.

El desarrollo Evolutivo es muy afín al modelo de Cascada en cuanto a su desarrollo y funcionalidad. De este modo el modelo cascada puede ser usado para administrar cada esfuerzo de desarrollo realizado.

**Consideraciones para el Ciclo de Vida Iterativo Evolutivo**

Tomando como base un *conjunto* de requerimientos claros y bien definidos (y comprendidos), es posible la construcción de incrementos de desarrollo a través de versiones funcionales del producto, se recomienda usar este modelo cuando:

* Se requieren versiones sucesivas del producto.
* Los requerimientos no son ampliamente conocidos al inicio del proyecto o son variables.
* El conjunto de requerimientos con los que se inicia en la primera iteración deben ser muy claros y ampliamente comprendidos.
* Se cuenta con disponibilidad de los usuarios para verificar las versiones sucesivas del producto.
* Es necesario crear una arquitectura de software muy flexible y de fácil mantenimiento, ya que al desconocer gran parte de los requerimientos al inicio, puede involucrar cambios significativos en la arquitectura por requerimientos futuros.
* El equipo del proyecto no cuenta con experiencia en la temática del problema a solucionar.
* Por último, considera que en el contexto de la organización, en la mayoría de los proyectos es necesario estimar y cotizar el total del proyecto al inicio. Al utilizar el modelo evolutivo será necesario hacer los acuerdos necesarios con el cliente para permitir este esquema. Adicionalmente será necesario determinar claramente el alcance y hasta que incremento (iteración) se terminará el proyecto.

# Criterios de Uso

Los criterios por tipo de proyecto están establecidos en la sección *“*Criterios por Tipo de Proyecto”.

Es importante mencionar que en los modelos de ciclo de vida se pueden utilizar prototipos en las diferentes etapas del ciclo. Un prototipo es una aplicación preliminar que simula o enfatiza cierta funcionalidad de la aplicación. Por ejemplo, un prototipo puede facilitar la definición de requerimientos y su validación con el usuario.

|  |  |  |
| --- | --- | --- |
| **Ciclo de Vida** | **Tipo de Proyecto** | **Condición** |
| Cascada | Incidencias | Los requerimientos estén claros desde el principio, no se tienen que realizar entregas parciales, la tecnología a utilizar ya está predeterminada y es bien conocida por el equipo de desarrollo. |
| Mantenimiento |
| Nuevo Desarrollo | Cuando el Proyecto está condicionado a iniciar la construcción del producto hasta la aceptación de los documento de análisis y diseño. |
| Iterativo Incremental | Mantenimiento | Cuando existen fechas compromiso de entrega de módulos o parcialidades del producto que obliguen el inicio prematuro de la etapa de construcción. |
| Nuevo Desarrollo |
| Iterativo Evolutivo | Mantenimiento | Cuando los requerimientos tienen una volatilidad alta, se tiene poco conocimiento del dominio sobre el cual se está trabajando y la complejidad del Proyecto es alta. |
| Nuevo Desarrollo |
|  | | |
| **Nota:**   * Es factible la combinación de los diferentes tipos de ciclo de vida del Proyecto, sí estas son debidamente justificadas a través de un procedimiento de Toma de Decisiones. * Cuando el cliente exprese necesidades particulares en la selección del ciclo de vida, éste será automáticamente asignado. | | |