Ciclos de Vida

• Modelos de ciclo de vida:

– abstracción de las **fases** o **estados** por los que pasa un producto software a lo largo de su vida

- Sucesión de etapas por las que atraviesa un producto software a lo largo de su existencia

– representa una **posible aproximación** a la producción de software

– Debe:

• Determinar el orden de las fases

• Establecer los criterios de transición entre fases

• No hay un modelo de ciclo de vida que sirva para todos los proyectos, depende de:

– cultura de la organización

– deseo de asumir riesgos

– área de aplicación

– volatilidad de los requisitos

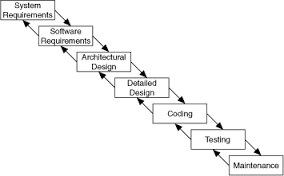
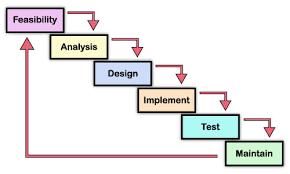
– comprensión de los requisitos

– experiencia previa

|  |  |
| --- | --- |
| causa | Efecto |
| 1 Un proyecto raramente sigue un flujo secuencial (sin regresar a etapas posteriores) antinatural | 1a no se definen todos los requerimientos en su etapa (no hay retroalimentación) |
| 2 el usuario ve resultados hasta el final | 2a es tardado, retrasos innecesarios al no hacer actividades paralelas (esperan)  2b no retroalimenta al desarrollador |
| 3 los errores de cada etapa (requerimiento, análisis y diseño) no se detectan | 3a los errores se detectan hasta la fase de pruebas, lleva más tiempo corregirlo en pruebas que en su propia etapa |
| 4 el modelo se deforma | 4a las actividades de pruebas y mantenimiento crecen |

**Funciona mejor si:**

* Se fortalecen las etapas iniciales
* Los requisitos se conocen completamente y sin ambigüedad y que no cambien



|  |  |
| --- | --- |
| Situación | Solución |
| -El cliente no sabe bien lo que quiere  -No se comprende bien lo que quiere  -Se desconoce la viabilidad tecnológica del proyecto | Crear una versión con funcionalidad reducida para  a) ser evaluada por el usuario  b) recibir retroalimentación para refinar la especificación |

Tipos de Prototipos

• Desechable:

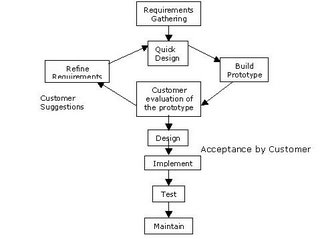
– sólo se incorporan los aspectos no entendidos y se elimina

• Maqueta:

– prueba algún tipo de interacción de datos, de usuario y entre componentes

• Prototipo evolutivo (versión Beta)

– el prototipo se desarrolla apegado a la especificación de las partes mejor entendidas para ampliarlas en un futuro y sea la versión definitiva.



El prototipo puede ser usado con otros prototipo-cascada prototipo-RAD

**Se usa cuando**

* Requerimientos deben ser bien conocidos (90%) sin sorpresas
* Usuarios involucrados
* Proyectos factibles en el corto plazo
* Cuando la funcionalidad a entregar sea incremental (adelantos y logros visibles)
* No se requiere que el sistema tenga alto rendimiento
* Riesgos técnicos bajos
* El sistema pueda ser modularizado (no muchas interrelaciones)

**Beneficios**

* Reducción de tiempo y aumento de productividad, menor costo

**Debilidades**

* Como el cascada los avances se tardan en visualizarse
* Aumento de desincronización y riesgo de no “cerrar” las funcionalidades relacionadas
* Es muy difícil para sistemas heredados
* Depende de que las dudas se solucionen en el momento sino hay retrasos (bomberazos)

**Modelo V**

**El mismo modelo V con otras palabras**

**Modelo V**

Es una variante del modelo de cascada con énfasis en la verificación y validación del producto, las pruebas son diseñadas y ejecutadas paralelamente con su correspondiente fase de desarrollo.

**Cuando usar modelo V**

* Cuando los sistemas requieran uso pesado
* Cuando los requerimientos son conocidos
* Cuando la tecnología es conocida

**Fortalezas**

* Énfasis en verificación y validación del producto en etapas tempranas del desarrollo del producto
* Cada producto es estable (90%)
* La retroalimentación al proyecto es más fácil con entregables
* Es sencillo y fácil de usar,

**Debilidades**

* No maneja iteraciones por fase lo cual es riesgoso
* No es fácil de manejar con cambios en requerimientos.
* No contiene actividades de análisis de riesgo

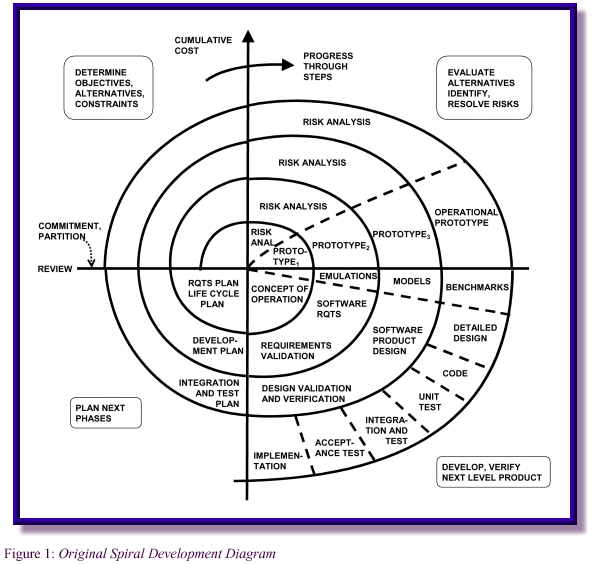
Filosofía V begin with the end in MInd

Arquitectura Put First Things First

Think Win/Win

Satisfy a maximum number of business needs with a minimum of effort

**Modelo Espiral**



Spiral Model

Agrega análisis del riesgo, integra el RAD prototipo y cascada

Cada ciclo involucra la misma secuencia de pasos como el modelo de cascada

Spiral Quadrant 1  
Determina objetivos, alternativas y restricciones

* Objetivos: funcionalidad, rendimiento, interface, determina factores críticos de éxito.
* Alternativas: construir, reusar, comprar, sub-contratar.
* Restricciones: costo, tiempo, interface, etc.

Spiral Quadrant 2  
Evaluar alternativas, identificar y resolver riesgos

* Estudiar alternativas respecto a objetivos y restricciones
* Identificar riesgos (falta de experiencia, nueva tecnología, tiempos cortos, procesos pobres o no hay).
* Resolver riesgos (evaluar si pierdes dinero de continuar desarrollando el sistema)

Spiral Quadrant 3  
Desarrollo del producto (al siguiente nivel)

* actividades: waterfall
  + Create a design
  + Review design
  + Develop code
  + Inspect code
  + Test product

Spiral Quadrant 4  
Planear siguiente fase

* actividades
  + Desarrollo plan de proyecto
  + Desarrollo plan de administración de la configuración
  + Desarrollo plan de pruebas
  + Desarrollo plan de instalación

**Cuando usar modelo Espiral**

* Cuando los costos y evaluación de riesgos es importante
* Compromiso con el proyecto a largo plazo (por los cambios)
* Usuarios no saben bien sus necesidades
* Requerimientos son complejos
* Se desarrolla una nueva línea de productos
* Cambios importantes son esperados (investigación y/o exploración)

**Fortalezas**

* Ayuda a detectar errores
* El usuario ve el sistema en etapas tempranas (prototipos) y retroalimenta
* Las funciones de alto riesgo son desarrolladas primero
* El diseño no necesariamente tiene que ser perfecto o completo
* Usuarios pueden estar más involucrados debido a todos los pasos del ciclo de vida
* Los costos acumulados son revisados frecuentemente

**Debilidades**

* Tiempo utilizado en evaluar riesgos es alto (mas personal)
* Tiempo gastado en planear, refijar objetivos, análisis de riesgo y prototipado puede ser excesivo
* el modelo es complejo
* se requiere experiencia en aseguramiento de riesgos es necesario
* espiral puede continuar indefinidamente por demasiada planeación y análisis
* no es fácil cerrar una fase y comenzar la otra fase