Clase 7-Modelos de procesos de desarrollo

Un **modelo de proceso** especifica un proceso general, usualmente como un conjunto de etapas. Provee una estructura genérica de los procesos que puede seguirse en algunos proyectos con el fin de alcanzar sus objetivos. Para cada proyecto el modelo de proceso debe pasar por un proceso de adecuación para producir la especificación del proceso del proyecto.

Modelo de cascada

En este modelo el proyecto se divide en diferentes fases

1. Análisis de requerimientos
2. Diseño de alto nivel
3. Diseño detallado
4. Codificación
5. Testing
6. Instalación
   Una fase comienza sólo cuando la anterior finaliza. Cada una de ellas se encarga de distintas incumbencias.

Como el orden de las acciones debe ser lineal tenemos que

- El final de una fase y el comienzo de la siguiente esta claramente identificado por un mecanismo de certificación y validación del resultado de cada fase.
- Todas las fases deben producir una salida definida/tangible. (con el que la siguiente fase trabajara)

Algunos de los documentos usuales son

- Documento de requisitos/SRS
- Plan de proyecto
- Documentos de diseño
- Plan de test y reportes de test
- Código final
- Manuales del software

Ventajas

- Conceptual-mente simple: divide claramente el problema en distintas fases que pueden realizarse de manera independiente
- Enfoque natural a la solución del problema
- Fácil de administrar en un contexto contractual: existen fronteras bien definidas entre cada fase.

Desventajas

- "Todo o nada": es mas riesgoso
- Asume que los requerimientos de un sistema pueden congelarse antes de que empiece el diseño -> imposible que los requerimientos no cambien nunca
- Congelar los requerimientos suele requerir elegir hardware. En procesos que llevan mucho tiempo esto es una desventaja pues pueden surgir nuevas tecnologias
- El cliente conoce el producto recien cuando finaliza todo el proceso, no hay feedback en el medio.
- No permite cambios

Es por esto que surge el modelo de cascada con feedback, ya que permite solventar algunas de las desventajas del modelo cascada "tradicional". En este modelo cada fase da un feedback a la anterior para una mejora.

Aplicación

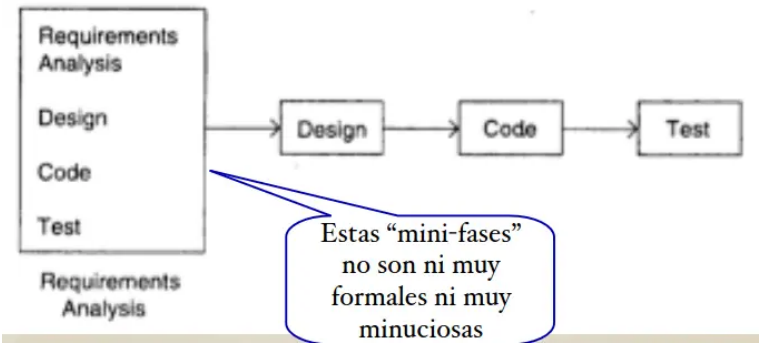
- Problemas conocidos
- Proyectos de corta duración
- Automatización de procesos manuales existentes

Prototipo

El prototipado aborda las limitaciones del modelo de cascada en la especificación de los requerimientos. Ya que en lugar de generar un documento con los mismos y "congelarlo", se produce un prototipo para poder comprenderlos. Esto permite que el cliente tenga una idea de lo que sera el SW.

- ayuda a disminuir los riesgos de requerimientos.

La etapa de análisis de requerimientos es reemplazada por una "min-cascada"



Desarrollo del prototipo

Se comienza con una versión preliminar de los requerimientos, en el prototipo se incluirán solo las características claves que necesitan mejor comprensión. El cliente usara el prototipo y proveerá feedback que mejorara la comprensión de los requerimientos. El prototipo se volverá a modificar y se repite el proceso hasta que los costos y el tiempo superen los beneficios de este proceso. Por ulitmo con todo este feedback se modifican los requerimientos iniciales y producen una **especificación final de los requerimientos**
El prototipo se **descarta**

Ventajas

- Ayuda a la recolección de requerimientos
- Mayor estabilidad en los requerimientos
- La experiencia en la construcción del prototipado ayuda al desarrollo principal.
- Reduce el riesgo
- Sistemas finales mejores y más estables.

Desventajas

- Comienzo pesado
- Potencial impacto en el costo y tiempo
- No permite cambios tardíos

Aplicación

- Cuando los requerimientos son difíciles de determinar y la confianza en ellos es baja.
- Cuando las interfaces con el usuario son muy importantes

Desarrollo iterativo

Combina beneficios del prototipado y cascada y aborda el problema de "todo o nada". La idea es desarrollar y entregar el SW incrementalmente, donde cada incremento es completo en sí mismo. De esta forma el testing se facilita.

Existe un feedback entre iteraciones que puede ser utilizado en iteraciones futuras.

Modelo con mejora iterativa

- Primero se realiza una implementación simple para el subconjunto del problema completo (solo los aspectos claves y faciles de entender)
- Se crea **lista de control del proyecto (LCP)** que contiene (en orden) las tareas que se deben realizar para lograr la implementación final.
  - Guía los pasos de iteración y lleva las tareas a realizar.
  - Cada entrada en LCP es una tarea a realizarse en un paso de iteración y debe ser lo mas suficientemente simple como para comprenderla completamente.
- Cada paso consiste en eliminar la siguiente tare de la lista haciendo **diseño, implementación y análisis** del sistema parcial y actualizar la **LCP**.
- Esto se repite hasta que se vacía la lista.

Beneficios

- Entregas regulares y rápidas
- Reduce el riesgo
- Acepta cambios naturalmente
- Permite feedback del usuario
- Prioriza requisitos

Inconvenientes

- La arquitectura y el diseño pueden no ser óptimos y verse afectados con tantos cambios.
- La revisión del trabajo hecho puede incrementarse
- El costo total puede ser mayor
- Sobrecarga de planeamiento en cada iteración.

Aplicación

- Cuando el tiempo de respuesta es importante
- Cuando no se puede tomar el riesgo de proyectos largos
- Cuando no se conocen todos los requerimientos.
- Donde no puede enfrentarse el riesgo de proyectos largos

Modelo de timeboxing

Es otro modelo iterativo, es una secuencia lineal de iteraciones. Donde cada iteración es una "mini-cascada": decidir especificación luego planear la iteración

```
Requer Constr Entrega Requer Constr Entrega Requer Constr Entrega
```

Timeboxing: primero fija la duración de las iteraciones y luego determina la especificación. Se divide la iteración en partes iguales y usa pipelining para ejecutar las iteraciones en paralelo.

```
Requer Constr Entrega
               Requer Constr Entrega
                              Requer Constr Entrega
```

El desarrollo se realiza iterativamente en "cajas temporales" (**time boxes**), donde cada una:

- Se divide en etapas fijas, que cada una tiene un equipo fijo
- Tienen igual duración -> todas las etapas tienen aprox igual duración.
- Cada etapa realiza una tarea bien definida que puede realizarse independientemente.
- Cuando un equipo de una etapa finaliza, se lo pasa al equipo de la siguiente etapa.

Ventajas

- Entregas regulares y rápidas
- Reduce el riesgo
- Acepta cambios naturalmente
- Permite feedback del usuario
- Prioriza requisitos
- Planeamiento y negociación un poco mas fácil
- Ciclo de entrega muy corto

Desventajas

- Grandes equipos de trabajo

- Administración mas compleja
- Se necesita mucha sincronización
- Posible incremento de los costos.

Aplicación

- Donde es necesario tiempos de entrega muy cortos.
- Donde hay flexibilidad en agrupar características.