#29/08/2022

========================

---- ENFOQUE EN EL DESAROLLO DE SOFTWARE ----

#Rod Stephens# propone clasificar los enfoques de desarrollo

* Modelos predictivos: Proporciona todas las funciones terminadas al mismo tiempo.
* Modelo incremental: En la primera versión entrega una versión básica. Después, cada interacción proporciona una mejor versión de las funciones.
* Modelo interactivo: Va liberando funciones con cada versión conforme avanzan los incrementos.
* Agile: Proporciona una versión básica de una función en la primera versión. Después trabaja en la siguiente función y mejora la primera.

--- Modelos Predictivos

- \*Cascada\* El paso previo necesita ser completado antes de continuar al siguiente.

Análisis/Requerimientos -> Diseño -> Implementación -> Verificación -> Despliegue -> Mantenimiento

- \*Cascada con retroalimentación\* Esta variación permite retroceder al paso anterior.

El verdadero reto está en completar efectivamente cada etapa antes de continuar a la siguiente.

! No se arrastran errores, pero tardamos más tiempo.

Análisis/Requerimientos <- -> Diseño <- -> Implementación <- -> verificación <- -> Despliegue <- -> Mantenimiento

- \*Sashimi\* Es un modelo en cascada con pasos que se superponen.

! Se requiere esfuerzo extra para llevar el control del proyecto

- \*Cascada Incremental\* Al final de cada incremento se genera una nueva versión del producto.

! Se requiero mucho tiempo ya que pasa múltiples veces por todas las etapas

- \*Modelo en V\* Se divide el producto desde su nivel conceptual más alto en tareas cada vez más detalladas.

Se trabaja en el proceso de volver hasta la parte superior conceptual del producto/aplicación.

! Se trabaja en paralelo por lo que el tiempo se reduce, pero aumenta la cantidad de recursos necesarios.

-------------------------------------------

--- Actividad

Un cliente espera que tú y tu equipo de 5 desarrolladores escriban una aplicación de procesamiento de texto tan poderosa como Microsoft Word en los próximos 3 meses.

-------------------------------------------

- Que modelo predictivo usarías? \*Cascada\*

- Cual sería la razón para usar tal modelo?

\* Al tener 3 meses, tenemos que cuidar el tiempo que nos va a tomar el modelo por esto descartamos incremental y retroalimentación.

\* Por el número de personas en el equipo descartamos el modelo en V.

\* Nos quedamos con Sashimi o Casada, pero el modelo

--- Modelo incremental

- \*Prototipos\* Visibilidad del producto desde el inicio del ciclo de vida con el primer prototipo

Permite introducir cambios en las iteraciones siguientes del ciclo

Permite la retroalimentación continua del cliente.

El resultado inicial de un prototipo nunca es el producto listo para el mercado

! Los cambios constantes pueden dar lugar a virios diseños y alteraciones en el código. Esto ralentiza el proceso.

Requerimientos -> #Ciclo#(Diseño -> Prototipado -> Refinamiento -> Revisión) -> Desarrollo -> Pruebas -> Mantenimiento

- \*Espiral\* Se usa para aplicaciones complejas y de alto riesgo. Es un modelo de cascada con prototipo.

1. Determinar objetivos, alternativas y restricciones.
2. Análisis de riesgo -> Evaluar alternativas -> Identificar y resolver riesgos -> Construir un prototipo.
3. Desarrollar y probar el prototipo
4. Planear la siguiente iteración

- \*Unified Process\* No es un proceso, es un \*framework\* de desarrollo iterativo incremental.

El trabajo se divide entre equipo/áreas que trabajan en paralelo en 4 fases:

1. Ideas, calendario inicial, objetivos generales
2. Requerimientos del proyecto. Casos de uso, diagramas, identificación de riesgos.
3. Se implementa y prueba el código.
4. Transfieres el proyecto a los clientes. Documentación, entrenamiento.

! No funciona bien para pequeños proyectos. En su lugar se usa para proyectos grandes y de alto riesgo.

- \*Rational Unified Process\* Es la version IBM del Unified Process

Usa las mismas 4 fases y disciplinas.

! La principal ventaja son las herramientas que provee IBM: plantillas, documentos, modelos

- \*Cleanroom\* Este modelo enfatiza la prevención y no la eliminación de defectos.

Usado en desarrollo de aplicaciones con pasos cuidadosamente monitoreados y probados.

1. Métodos formales: El código se produce utilizando #metodos matemáticos# formales que ayudan a garantizar que el código satisfaga los modelos de diseño
2. Control de calidad estadístico: El código se produce de forma incremental. La calidad de cada incremento se mide para garantizar que el proyecto esté haciendo un progreso aceptable
3. Pruebas estadísticas: las pruebas utilizan experimentos estadísticos para estimar la calidad de la aplicación

-------------------------------------------

--- Actividad

Se planea implementar una aplicación médica (control de medicamentos) con 10 funcionalidades principales.

Las especificaciones no están del todo claras. Sin embargo, el cliente desea comenzar con el proyecto lo antes posible e ir trabajando mientras se finaliza su desarrollo.

-------------------------------------------

- Que tipo de enfoque para el desarrollo de software usarías? \*Espiral\*

\* Se nos indica que es una aplicación medica por lo que se de alto riesgo.

\* Cuenta con un número elevado de funcionalidades principales lo que la hace compleja.

\* Por esas razones elijo el enfoque de espiral, aparte, este enfoque nos permite comenzar a trabajar con objetivos no tan claros y en las siguientes iteraciones los vamos refinando.

- Conviene utilizar un modelo predictivo?

No creo ya que al no tener los objetivos/especificaciones claras es muy probable que se termine el proceso y el cliente se dé cuenta que quería algunos cambios. Si tuviera que elegir uno usar el de cascada incremental ya que podemos generar diferentes versiones, pero considero que sería arriesgado y consumiría más tiempo.