1

El ciclo de vida de un software es la duración de la creación de un software desde el análisis hasta finalizarlo con el mantenimiento

2

Análisis, diseño, codificación, pruebas, documentación y mantenimiento

* Análisis. Se conocen las necesidades del cliente y se especifican los requisitos del software. Los requisitos para ellos son:
  + Ser completas
  + Ser concisas
  + Evitar ambigüedades
  + Ser entendible por el cliente
  + Evitar detalles del diseño
  + Separar requisitos funcionales y no funcionales
  + Dividir y jerarquizar el modelo
  + Fijar criterios de validación
* Diseño. Se descompone y organiza el sistema en elementos para volverlos independientes y desarrollarlos por separado. Se especifica la interrelación y la funcionalidad de los elementos. Algunos tipos de diseños son:
  + Diseños arquitectonicos
  + Diseños detallados
  + Diseños de datos
  + Diseños de interfaz
* Codificación. Se emplean dos tipos de lenguajes:
  + Lenguaje de programación (Java)
  + Lenguaje de otro tipo (HTML)
* Pruebas. Estas se emplean para evitar que el sistema funcione de manera errónea y para descubrir errores del sistema
* Documentación. En este se explica como funciona el código y para que otra persona pueda utilizarlo
* Mantenimiento. Durante la creación del sistema es necesario realizar cambios. Para ello muchas veces usamos versiones anteriores para evitar errores. Algunos son los siguientes:
  + correctivo
  + perfectivo
  + evolutivo
  + adaptativo

3

Modelo de cascada. Consiste en ir realizando cada tarea para el desarrollo del software y hasta que cada fase no esté terminada no se pasa a la siguiente

4

Las ventajas del modelo en cascada son:

* El modelo tiene una gran antigüedad por lo que es conocido y eficaz.
* Se asegura que cada fase este bien creada y elaborada

Desventajas del modelo en cascada son:

* De haber un error en cualquiera de las fases es preciso volver a empezar desde el principio por lo que se pierde tiempo
* Hay fases que tienen más o menos variantes

5

6

Se suele crear un modelo del prototipo durante la fase del analisis para que el cliente señale los cambios correspondientes que desea

8.

Los cuatro principios del manifiesto ágil son:

* Individuos e interacciones sobre procesos y herramientas
* Software funcionando sobre documentación extensiva
* Respuesta ante el cambio sobre seguir un plan
* Colaboración con el cliente sobre negociación contractual

9

La historia de usuario es una representación de una actividad escrita utilizando el lenguaje común.

Esta es una manera rápida de administrar los requisitos de los usuarios sin que sean demasiado extensos sus documentos.

Sus principales características son:

* Deben ser independientes unas de otras
* Negociables
* valoradas por los clientes o usuarios
* Estimables
* Pequeñas
* Verificables

Sus ventajas son:

* Al ser cortas, se pueden realizar de manera rápida.
* Se necesita poco mantenimiento
* Permite ser cercano al cliente
* Divide el proyecto en pequeñas entregas
* Estimula fácilmente el esfuerzo de desarrollo
* Ideal para proyectos no muy claros

Sus desventajas son:

* Sin pruebas de validación pueden quedar abiertas interpretaciones haciendo difícil utilizarlas como base para un contrato
* Se necesita un contacto permanente con el cliente mientras el proyecto dure
* Las pruebas de usuario pueden resultar difíciles
* Requieren desarrolladores muy competentes

10

https://es.wikipedia.org/wiki/Lean\_software\_development

El desarrollo de software lean fue inicialmente, originado en el Sistema de Producción de Toyota y ahora, apoyado por una corriente que está surgiendo desde la comunidad Ágil. Este método ofrece todo un marco teórico sólido y basado en la experiencia, para las prácticas ágiles de gestión.

El software lean tiene siete principios, similares a los principios de manufactura esbelta.

* Eliminar los desperdicios
* Amplificar el aprendizaje
* Decidir lo más tarde posible
* Entregar tan rápido como sea posible
* Capacitar al equipo
* Construir integridad intrínseca

11

Ventajas de KANBAN

Organización del flujo de trabajo. Las tareas se organizan entre los trabajadores

Distribuye las tareas

Cumplen los tiempos de entrega

Se mejora la calidad del producto

Evita la acumulación de trabajo

Desventajas

Costes. En caso de trabajar con unidades muy grandes este sistema puede salir muy caro

Limitado

No es óptimo para todo tipo de proyectos. En caso de haber una variación o evento inesperado puede afectar de manera negativa al proyecto.

No permite anticiparse a grandes aumentos de la demanda

1

Kanban se basa en una transparencia total del trabajo y una comunicación en tiempo real de la capacidad. Para ello se usa el tablero de kanban que se puede considerar una fuente importante sobre el trabajo del equipo.

Este tablero permite que el trabajo del equipo se visualice, y se identifiquen y resuelvan inmediatamente todos los problemas . El tablero de kanban se basa en los siguientes trabajos: Por hacer, En curso y Terminado.

Las diferencias entre Kanban y Scrum son:

* Mientras que Scrum el cliente ve el trabajo final después de cada sprint (cada sprint dura unas 2 o 4 semanas), en kanban el cliente ve continuamente el proyecto.
* En el equipo de scrum siempre hay un propietario del producto, experto de scrum y equipo de desarrollo. por el contrario en kanban no hay funciones existentes (como mucho un orientador ágil)
* Durante el kanban se puede pedir cualquier tipo de cambio. Por el contrario scrum no permite hacer cambios durante sus sprints porque pondría en peligro las elecciones de estimación

2

Scrum es un proyecto que funciona por ciclos temporales cortos y de duración fija que empieza cuando el cliente explica los detalles de su proyecto o producto resolviendo las dudas de los programadores sobre este. Después de la reunión el equipo se reúne y se reparte las tareas, se reorganizan para colaborar entre ellos, compartir información, resolver problemas…

Cada día se realiza una reunión donde se plantean las siguientes cuestiones

* ¿Qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo?
* ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?
* ¿Qué impedimentos tengo o voy a tener que nos impiden conseguir nuestro objetivo?

El último día se realizará una revisión donde se muestran el producto de una manera muy sencilla y el cliente dirá si está satisfecho con el trabajo.

El equipo analizará cómo han trabajado y cuáles han sido los principales problemas

3

Product backlog. Es la única fuente de autoridad que permite mostrar los resultados del equipo.

Sprint backlog. Son los objetivos prioritarios del equipo.

4

Jefe del proyecto--------------> Coach del equipo

Cliente-------------------------------> Product owner

Equipo de desarrollo--------->Development team

5

Cultura de la empresa sobre el trabajo en equipo

Compromiso con el cliente para colaborar y poder ver los avances y proyectos

Compromiso de la Dirección para resolver problemas, realizar cambios …

compromiso y colaboración entre los diferentes equipos y sus miembros

Relación entre el cliente y los equipos

Facilidad de hacer cambios

Estabilidad en los grupos

6

Simplicidad. este tipo de proyectos se basa en diseños simples, pruebas en continua mejoras por lo que no es necesario una gran complejidad.

Comunicación. Suele haber equipos de dos personas, además el cliente es parte del equipo de desarrollo por lo tanto hay que decirle y mostrarle las novedades.

Retroalimentación. Tanto las las parejas como el equipo de desarrollo

Respeto y humildad. En los equipos debe haber respeto para evitar discusiones y enfados que podrían afectar al proyecto.

Valentía y coraje

7

Metáforas, ayudan a que las personas no especializadas entiendan a lo que nos referimos

Pequeñas entregas, el cliente irá viendo los avances del proyecto y cómo va evolucionando

Juego de planificación es la comunicación entre la parte técnica y empresarial

Diseño sencillo permite que sea más fácil su uso e implantar nuevos cambios en el Programación por parejas las parejas trabajan con un solo ordenador y se ayudarán, revisarán el trabajo que ambos realizan.

Pruebas se realizarán pruebas para comprobar el buen funcionamiento del programa