

Métodos Agiles

Los primeros sistemas eran más tradicionales, más estables, tenían los requerimientos definidos, pero ahora hoy en día se modificó este concepto, porque no se tiene bien claro las tecnologías de lenguaje de programación porque puede cambiar.

La metodología ágil es una forma de intentar paliar el problema de tiempo rápido de respuesta requerimientos vagos, o inexistente, falta de conocimientos. *porque en un lenguaje de programación [...]*.

kanban

¿Cómo es en términos generales kanban? características:

Pueden poner las tablas que quieras, background conjunto de tareas por hacer.

Kanban propone un cambio en el concepto de jefe. Cambia la concepción de trabajo de sistema.

En kanban todas las tareas están ahí, y nosotros somos responsable de trabajar.

Sacando una calidad de trabajo mucho mejor.

Scrum

Las principales características de scrum, lo que hace en concepto Es

¿Qué es un sprint?

Es una carrera, es generar una pequeña interacción, en las empresas es muy común que utilicen kanban para cada una de esas tareas usan scrum. Y es muy común combinarlas.

Un sprint no puede llevar más de 2 semanas,

Las secciones de scrum se hacen diariamente, en un equipo de trabajo, equipo se reúne informalmente, agregando una conversación establecido, haciendo tres preguntas:

¿Pudiste hacer lo que te propusiste el día de ayer? ¿Lo solucionaste?

¿Qué vas hacer el día de mañana?

Una suma de falta de esta técnica puede llevar consecuencias de estancamiento de tareas.

Nota:

- *Las Habilidades branda* no está escrito por ningún lado.
- *Las Habilidades duras* son concepto que estudiaremos a lo largo de la carrera.

Nunca tener un delay, tener un problema por más de 2 horas, es un problema y el objetivo es estar siempre trabajando.

Partir un proceso grande en pequeños procesos

Las consecuencias que se ven cuando ven un proceso grande y los partimos en pequeños procesos son:

- La resolución a corto plazo.
- Implica un conocimiento mayor al proyecto que realizará, tardando más tiempo en el análisis. En consecuencia, se dedica mucho tiempo en programar que en analizar.

Se tiene que dedicar más tiempo al análisis y al diseño, y hacer iteraciones aquí, porque mejor análisis echo mejor diseño se logra realizar y un buen diseño en teoría no deja a lazar ningún problema serio de programación.

Nota:

- *Las metodologías ágiles es para la administración del proyecto de desarrollo de software, las principales son kanban, scrum.*

Continuo prime

¿Qué es continuo prime?

Es una consecuencia de trabajar con metodología ágil. Porque cuando hacemos unos sprinter, estoy generando nuevo código funcional a diario, en consecuencia, genera una desviación de las metodologías ágiles de diseño. Es lo que se conoce como una infraestructura necesaria para las metodologías ágiles.

Un continuo prime es, construir continuamente. Quiere decir que todas las noches se compila el código para actualizar el sistema, y se ejecuta un set de pruebas que se ven resultados.

Consecuencias grandes, realiza una carga de trabajo muy grande en el proyecto, porque necesitamos una herramienta más, que realice baje, compile, evalúe y diga esta. pero lo que no tenemos es el lugar, la implementación, las pruebas, usar algún lenguaje que me genere una automatización de las pruebas. Sirve para programar sin parar.

Para empezar a trabajar con metodologías ágiles y que sea un proceso completo coherente y que sirva, tengo que dedicar mucho tiempo a la infraestructura.

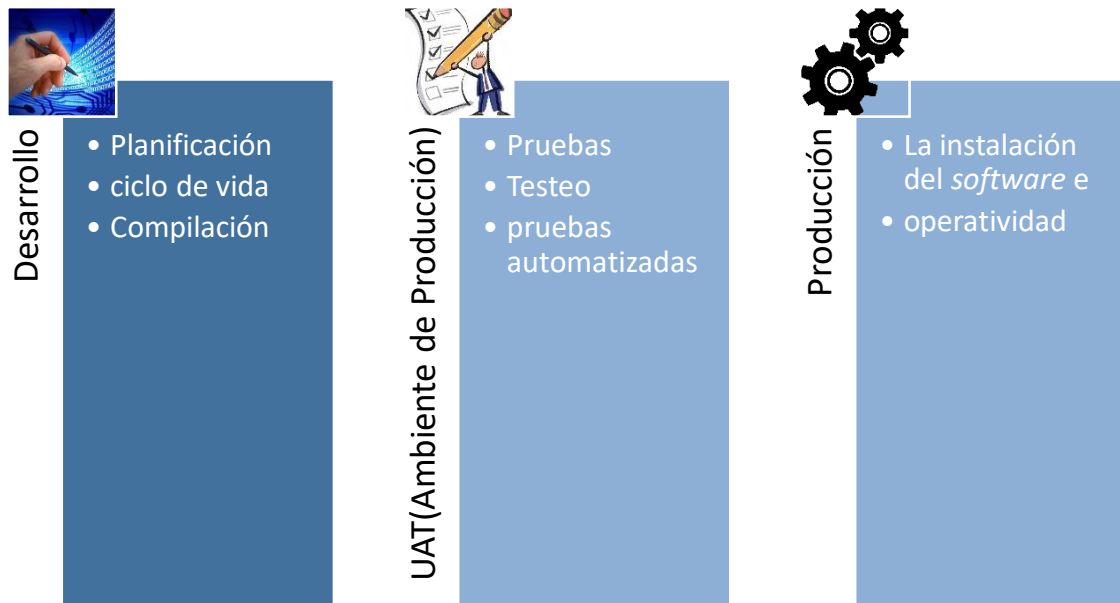
Los equipos que trabajan con metodologías ágiles están divididos en:

- los programadores que utilizan las metodologías ágiles
- un pequeño equipo que se encarga de mantener todas las estructuras, y ver cómo funciona eso.

¿Cuáles son las principales herramientas de continuo prime? Jackson, Jenki.

Menciones dos herramientas que sirven para el continuo prime son el Ant y Maven

- Maven sirve para hacer la validación de pruebas de librerías, versionados,
- Ant permite de tener un script medianamente automatizado donde puedo tener: copiar, bajar, mover, probar. Siendo una herramienta simple

Infraestructura:**Entorno de Trabajo:**

Jackson y jenki pasa de un ambiente a otro, haciendo las pruebas correspondientes El control de versión que usamos es el GIT. Que se integra perfectamente al Ant, jenki y a Jackson.

Pruebas de Software

¿Cuáles se conocen? Automatizadas, pruebas de unidad, pruebas de código, pruebas de caja negra, pruebas de caja blanca, pruebas de Integración.

1. Pruebas de unidad

Buscar librería de unix test, a nivel de métodos, estas librerías hacen un test por unidad. Darle valores definidos completo de entrada y salida, porque cuando uno trabaja voy interactuando con distintos métodos, si un método está mal puede afectar hasta la punta del sistema. Siendo muy difícil sacar los bugs, es el spot. En donde todos los lugares que llama ese método en todo el sistema.

Para saber constantemente que todos los métodos están bien implementados.

Teniendo nueva responsabilidad para el dueño del código. Se tiene que dedicar tiempo para hacer estas pruebas.

2. Pruebas de integración de módulos

Pruebas de integración con performance

3. Pruebas de Caja Blanca

Son pruebas metidos dentro del código, como trabaja el código internamente. Que hace y que no se hace

4. Pruebas de Caja Negra

Qué es lo que hace el código, no teniendo acceso al código,

¿Cómo automatizaría una prueba de caja negra?

Con un programa que simule un usuario, programa de estrés.

5. Prueba funcional (distinto de prueba caja negra)

Mozilla selenium, tiene una aplicación que graba un scan, y empieza a trabajar en HTML, y graba todos los pasos que hicimos, que se puede repetir todas las veces que se desee. Haciendo automatizar las pruebas funcionales.

El pro de selenium, es tal cual cómo lo ve el usuario, por lo cual podemos donde se ve la imagen, por donde está, lo que queramos,

La contra tenemos que levantar un browser, tener un usuario con permisos y accesos.

6. Prueba de Estrés

Es simular cuantos usuarios pueden usar concurrentemente el sistema dando un tiempo de respuesta medianamente digna. Como se hace, se escribe un programa, que va a simular como si fuese un browser, que se va a conectar con el servidor WEB (ejemplo). *Logrando un browser que simule usuarios que usan el sistema*

Otra herramienta, la mejor librería que sirve para hacer pruebas de estrés es `loadrunner`

Virtualizadores

Sirve Para crear entornos, en un solo equipo pongo una máquina virtual y tengo entornos (Diseño, UAT, Producción).

Comentar

Es una convención de buena práctica, dejar un comentario encabezado por archivo. Con las siguientes características.

- El nombre del programa,
- La fecha de creado,
- Una descripción breve,
- Autor

Comentarios de distintas cosas. Cada comentario tiene que estar equilibrado para la comprensión. “Que no falte pero que no sea excesivo”.

Diseño de Sistema

Primero se empieza con los requisitos, los “casos de uso” es lo más apropiado para empezar el proyecto. ¿Qué voy hacer? Es una tarea que me voy a mover. Se van poniendo las distintas partes que va teniendo el sistema.

Puedo tener historias de usuario distintos, por sistema, y partes de que sistema y que se puede trabajar el compilador.

¿Porque caso de uso?

Para cada actor se muestra la relación entre un caso de uso,