Universidad Politécnica de Durango

Ingeniería en software

Unidad II

Modelos y metodologías de desarrollo de software

(Caso práctico)

Docente: Elva Liliana Limón Dávila

Alumno(a): Daniela Beatriz Martínez Montes
Matrícula: 1603150184

13 de Octubre de 2017

Índice

Introducción 3
Contenido 4
Tablas 4
Tabla 1-buenas y malas prácticas 4
Cuadro comparativo – modelos de desarrollo 5
Cuadro comparativo-Metodologías 7
Modelo desarrollo apropiado 9
Conclusión 12
Bibliografía 13

Introducción

En la actualidad la mayor parte del software no puede desarrollarse de manera individual y exige la participación de equipos de desarrollo, que en algunos casos ni siquiera comparten el mismo lugar o país de trabajo. Así, para un ingeniero informático es importante contar con los conocimientos y habilidades para participar eficazmente en un equipo de trabajo, todo esto para que se adapte fácilmente al ambiente de trabajo ya que el desarrollo de software, es uno de los sectores tecnológicos más competitivos, ha tenido una evolución constante en lo que se refiere a las metodologías y modelos, es decir las formas en las cuales se realiza la planeación y organización para el diseño del software, básicamente con el objetivo de mejorar, optimizar procesos y ofrecer una mejor calidad del producto, en el presente documento se expandirán los conocimientos sobre las metodologías y modelos ya existentes, mediante cuadros comparativos, así mismo se verá qué modelo de desarrollo es el que mejor se adapta al proyecto y se justificara el porqué de esta elección, el proyecto del que estaremos hablando es iKeeper, este es un rastreador geosatelital, el cual obtiene las ubicaciones y son enviadas a la aplicación, ikeeper almacena todas las ubicaciones, una de las funciones especiales es que si se desea ver cual fue una ubicación particular a una determinada hora, solo se necesita darle doble clic y aparecerá nuestro marcador verde en esa ubicación.

Contenido

En esta sección del documento se analizan los distintos modelos y metodologías del desarrollo de software y se realiza la elección de estas, de acuerdo a las características del proyecto desarrollado en el cuatrimestre Mayo-Agosto del 2017, así mismo se explicaran las buenas y malas prácticas que ocurrieron durante el desarrollo de dicho proyecto

Tablas

A continuación se mostraran algunas tablas para poder tener un mejor entendimiento de los temas que se mencionaron arriba

En la siguiente tabla se muestran las buenas y malas prácticas que fueron realizadas de forma individual y en equipo en el desarrollo del proyecto

Tabla 1-buenas y malas prácticas

Buenas Prácticas		**Malas Prácticas**	
Individual	**equipo**	**individual**	**equipo**
Dedicación al proyecto	Todos saben cómo funciona el código	No comentar líneas código	Demasiada multitarea
Previsión del presupuesto	Todos saben en que fue gastado el presupuesto	No documentar	Enfocarnos todos en un mismo trabajo
Aceptar criticas	Comunicación con los demás miembros	No contaba con horarios fijos para trabajar	En momentos no sabíamos que estaba haciendo cada uno
Cooperación en el proyecto	División de gastos	Detección tardía de errores	No establecer un objetivo claro desde un principio

En el siguiente cuadro comparativo se muestran algunos de los diferentes modelos de desarrollo, donde se explican sus ventajas y desventajas, así como su característica más importante

Cuadro comparativo – modelos de desarrollo

Modelo	Características	Ventajas	Desventajas
Cascada	No se puede pasar a una siguiente etapa hasta que se culmine en la que esta	• Permite tener bajo control el proyecto • Es muy sencilla • facilita la gestión de proyectos • Sus fases no se mezclan	• Es muy difícil incorporar nuevas cosas • Si se piden cambios al final, ya no se pueden modificar • Realizar pruebas en etapas avanzadas • Tarda mucho tiempo en pasar todo el ciclo
Espiral	En cada giro se construye un nuevo modelo del sistema separa la parte del usuario (la parte del usuario) y el procesamiento de los datos	• Es un Modelo de proceso realmente adaptable • puede aplicarse a lo largo de vida del software • El análisis del riesgo se hace de forma explícita y clara	• Requiere mucho tiempo • es más costoso por que se requiere más tiempo • su complejidad es un poco más elevada

Basado en componentes	conduce a la reutilización del software, y la reutilización proporciona beneficios a los ingenieros de software	• reutilización del software (volver a utilizar lo que ya tenemos) • se ahorra tiempo en el desarrollo de software • Reduce los errores • Reduce los costes • Sus ciclos de desarrollo son más cortos	• Requiere experiencia en la identificación de posibles riesgos • Tener definido como se va a trabajar desde un principio

En el siguiente cuadro comparativo se muestran algunas de las diferentes metodologías que existen para el desarrollo de software, donde se explican sus ventajas y desventajas, así como su característica más importante, cada metodología combina una filosofía y un conjunto de directrices de desarrollo

Cuadro comparativo-Metodologías

Metodología	Característica	Ventajas	Desventajas
PSP	• más enfocado al proceso administrativo es decir documentación	• Resultados predecibles • Eficiencia de procesos • control de calidad • mejora productividad de las personas • se implementa a nivel personal	• cada uno de los miembros tiene que tener el compromiso y la disciplina de seguir el plan • si algún miembro se va, es necesario entrenar a los nuevos miembros
TSP	Toma de base los principios del PSP para realizar los procesos y principios de ingeniería de software en un ambiente de trabajo ene quipo	• Ayuda a los líderes de proyecto a dirigir y motivar a los grupos • Reduce el tiempo en solucionar problemas • Mejora la productividad • Detección temprana de riesgos y defectos	• cada uno de los miembros tiene que tener el compromiso y la disciplina de seguir el plan • Cada miembro debe de estar entrenado en el PSP
Programación extrema	tiene como base la simplicidad y como objetivo principal la satisfacción del cliente	• Fomenta la comunicación entre los clientes y los desarrolladores • Puede ser aplicada a cualquier lenguaje de programación.	• se necesita de la presencia constante del usuario, lo cual en la realidad es muy difícil de lograr. • Altas comisiones en caso de fallar

SCRUM	Impulsa la comunicación verbal con todos los miembros del equipo y disciplinas involucradas en el proyecto	• el ScrumMaster puede ir controlando el proyecto y delegando roles. • Cada persona sabe que es lo que tiene que hacer y no es necesario estar reorganizando una y otra vez	• reunirse demasiadas veces por el mismo tema, puede provocar perdida de interés. • Si una persona renuncia o hay algún cambio es complicado remplazar ese rol.
ASD	Esta metodología se adapta al cambio en lugar de luchar contra el	• Promulga colaboración y interacción de personas. • Es muy tolerante a cambios.	• Los errores y cambios que no son detectados con anterioridad afectan la calidad del producto y costo total. • La prolongación de ciclos también afecta la calidad del producto y su costo total.
Desarrollo ágil	se caracteriza por ser rápida, dinámica, con contenido específico y corresponder de manera apropiada a los cambios	• Esta preparados para cambios durante el proyecto • Menos énfasis en la arquitectura del software	• Es muy propenso a errores

En esta parte del documento se especifica qué modelo de desarrollo ha sido el apropiado para aplicarlo al proyecto, así mismo se da la justificación de esta elección y se explica que tareas tienen que ser realizadas en cada una de sus fases

Modelo desarrollo apropiado

Modelo basado en componentes

Este Sería el modelo más apropiado para el proyecto ya que este conduce a la reutilización del software, se reutiliza código que puede ser adaptable para cualquier situación, y en este caso, para poder desarrollar el proyecto, se reutilizó mucho código, por ejemplo:

- Código para conexión a base de datos, para poder utilizar sentencias dentro del proyecto, obtenida de la clase de programación orientada a eventos
- Código para poder recibir las coordenadas (arduino), obtenido de la página oficial de tinyGPS
- Código para que sea posible poner en funcionamiento la referencia de Gmap, obtenido del repositorio de github gmaps

Además de esto, nos permite realizar cambios y por esto mismo nos favorecería para una construcción más rápida del software y que cuente con mejor calidad

Lo que se desarrollaría en cada fase es lo siguiente

I. *Análisis y comparación de procesos*

La implementación del proyecto será en ambientes computacionales, se basa en una aplicación de rastreo GPS para computadoras o portátiles. Sera lo más sencillo posible para que sea fácil su utilización para usuarios comunes

- Conexión satelital
- Integración de mapa
- Mostrar ubicación en tiempo real del lugar en el que este el dispositivo.
- Utilizará exclusivamente sistema operativo Windows

- Almacenamiento de ubicaciones para que el usuario tenga la opción de visualizar la ubicación en la hora que seleccione
- Creación de cuentas para nuevos usuarios

- Contar con una cuenta de acceso al sistema.
- Un recibimiento de coordenadas al sistema.

El sistema se desarrollará en licencias abiertas, por lo tanto, no se pagará por su uso

II. Identificación de componentes

Los componentes serán seleccionados por los requerimientos funcionales que se mencionaron anteriormente, se evaluara cuáles de ellos necesitaran ser modificados

- Cuenta de acceso al sistema: Los usuarios del sistema tendrán acceso a el, mediante su propia sesión y contraseña, esto por cuestiones de seguridad y para delimitar roles ya existentes
- Sistema operativo Windows: el ambiente en donde trabajara exclusivamente esta aplicación es en Windows
- Creación de cuentas: Sera necesario crear un usuario y contraseña en caso de que sea la primera vez que se utiliza la aplicación
- Recibimiento de coordenadas: Para que la aplicación pueda brindar su uso esperado, esta recibirá coordenadas cada intervalo de tiempo que el usuario decida

III. Integración de sistema:

Una vez que ya todos los componentes fueron evaluados, se comenzó a codificar y ensamblar los componentes:

- Se crea la interfaz de registro, con todos los campos que debe de tener(nombre,apellido,correo electrónico,sexo)
- Se crea la interfaz de login
- Se crea la interfaz principal, donde se pondrá el mapa y en donde las coordenadas van a llegar

- Las respectivas bases de datos de los tres puntos anteriores son creadas para que todos los registros e información sean almacenadas ahí

IV. *Pruebas:*

Los elementos ya previamente integrados y programados comenzaran a testearse, es decir se verificara que el funcionamiento de la aplicación este correctamente y que cumplan con todos los requisitos antes mencionados, los puntos más importantes, son los siguientes:

- Comprobar que el recibimiento de coordenadas sea correcto (no existan valores nulos)
- Comprobar que los valores recibidos si estén almacenados en la base de datos
- Que el servicio de mapa (google maps) tome de manera correcta las coordenadas
- Que los usuarios si puedan eliminar registros almacenados
- Las nuevas cuentas se creen correctamente

En caso de que uno de los requerimientos no estén funcionando de forma adecuada, este tendrá que ser modificado, para poder re-integrarlo

V. *Mantenimiento:*

Una vez que el sistema ya esté listo, cada seis meses se va a realizar un mantenimiento preventivo, los encargados de hacerlo serán los desarrolladores, es decir se va a realizar actualizaciones de las referencias de extensiones que fueron usadas, tales como Gmap

Conclusión

La administración de proyectos de desarrollo de software es de gran utilidad ya que nos permite darnos cuenta que gestionar el desarrollo de un producto no es cosa sencilla o de una sola vez, se tienen que tomar en cuenta diversos puntos, tales como entregar dentro del plazo previsto y con los fondos establecidos. Y si bien no estamos muy acostumbrados a realiza documentación, esto nos permite darnos cuenta que la administración de un proyecto no es una actividad insignificante, puede ser tan transcendente.

De igual forma para determinar qué modelo era el más factible para nuestro proyecto, se estudiaron diversos modelos, y fue posible ver sus ventajas y desventajas y cuál de estos resultaba más costoso. Para nuestro proyecto el modelo que fue elegido fue el basado en componentes, que representa una gran opción, ya que brinda un paradigma de desarrollo donde el software es desarrollado mediante la reutilización de componentes de software pre-existentes, además fue fácil darnos cuenta que cuando se está llevando a cabo la elaboración de algún sistema de software siempre se va a presentar algún problema durante el desarrollo, quizá no un problema funcional, la mayoría de las veces son porque las especificaciones del cliente no fueron claras, el desarrollo se llevó a cabo sin volver a preguntar y al final esto tuvo como consecuencia que el sistema fuera rechazado por el cliente, Implementar un modelo de desarrollo sería útil para estos casos, ya que la mayoría de estos nos permiten a nosotros los desarrolladores hacer citas continuas con el cliente y llegar a saber si lo que se haciendo va por buen camino

Si bien en el proyecto que fue desarrollado no fue utilizada ninguna metodología, estas llegan a ser altamente necesarias, ya que permiten tener una mejor producción y una mejor relación con el equipo

Bibliografía

https://www.fing.edu.uy/inco/cursos/ingsoft/pis/proceso/MUM/dat/intro/intro.htm

http://www.tdx.cat/bitstream/handle/10803/6837/14Jcb14de16.pdf?sequence=14

https://www.ctr.unican.es/asignaturas/Ingenieria_Software_4_F/Doc/M2_08_Administracion-2011.pdf

https://issuu.com/ingildardo/docs/proyecto-sgo