

Actividad [#2] - [Plan de trabajo]

[Ingeniería de Software 1]

Ingeniería en Desarrollo de Software

Tutor: Eduardo Israel Castillo García

Alumno: Pilar Barajas Cervantes

Fecha: 21/05/2023

Índice

Introducción	3
Descripción	4
Justificación	5
Plan de proyecto	6
Roles y responsabilidades	7
Calendario	9
Control de versiones de software	10
Conclusión	11
Referencias	12

Introducción

En esta segunda actividad hablaremos de cómo realizar un plan de trabajo sobre el proyecto solicitado para la empresa Coppel en donde ahora se pretende simular el proceso de gestión del proyecto de control de versiones de software. La fase de planificación es un punto clave en cualquier proyecto, en esta fase se estima el trabajo, el tiempo y el dinero que va a suponer la realización del proyecto. Este análisis, además de permitir conocer los recursos necesarios, es de gran ayuda en fases posteriores del desarrollo.

La fase de planificación se puede dividir a su vez en:

- Planificación temporal
- Estudio de viabilidad

En la primera parte, se elabora un calendario o un programa de tiempos. En estos se estima el tiempo necesario para la realización de cada una de las partes del proyecto y una fecha de finalización estimada.

La segunda parte se centra en la viabilidad del proyecto. El estudio de la viabilidad se puede dividir a su vez en dos apartados:

- Viabilidad económica
- Viabilidad legal.

Descripción

Después de ver analizado el proyecto solicitado por la empresa Coppel y elegir el modelo que estaríamos utilizando para el desarrollo de sistema de información, ahora simularemos el proceso de gestión del proyecto y el control de versiones de software. Después de identificar el método o técnica que estaríamos utilizando, estaríamos utilizando la metodología cascada ya que creemos que este método sería el mejor. La metodología cascada se utiliza cuando hay una visión clara de lo que deberíamos de realizar y lo que debería ser el producto final.

El modelo, cascada es simple y fácil de utilizar y consta de cinco fases secuenciales

- **Análisis de requisitos**
- **Diseño**
- **Implementación y pruebas unitarias**
- **Integración y pruebas de sistema**
- **Operación y mantenimiento**

Justificación

Las ventajas de utilizar la metodología cascada, es porque el modelo es simple y fácil de usar.

Como la metodología es bastante rígida, es fácil de administrar porque cada fase consta de entregables específicos. El proceso es bastante predecible, todos tienen una idea con anterioridad cómo se evolucionará el proyecto. El punto clave de la metodología es que no hay posibilidad de cambios o errores por lo que la planificación es fundamental la calidad de trabajo inicial define en mayor medida el resultado final.

Yo pienso y creo que el modelo, cascada es el mejor ya que en esta ocasión para el desarrollo de sistemas el cliente está especificando los módulos que conformaran su desarrollo de sistemas, como cualquier otro método el modelo, cascada se diseño para producir resultados positivos. Se creo específicamente para producir resultados positivos garantizando que se cumplan todos los requisitos de las partes interesadas y los consumidores. Aparte de esto también hay otros grandes beneficios al aplicar este método, como el proporcionar una visión concreta de la estructura del proyecto, también el permitir centrarse en los objetivos, así como una difusión adecuada de la información.

Plan de proyecto

Datos generales	
Visión general	<p>El cliente necesita un sistema de información bien definido en donde al entregar el proyecto software este sea un diseño muy bien implementado y cubra sus necesidades. Que los empleados al momento de realizar un cobro la interfaz sea entendible sin ninguna dificultad.</p>
Alcance	<p>Desarrollar un software de manera ordenada y diciplinada tiene ventajas, siendo así que es como se puede cumplir con los requerimientos del cliente, el objetivo es lograr que el cliente este contento con el resultado.</p>
objetivos	<p>Objetivo 1: terminar el desarrollo en tiempo y hora acordada con el cliente.</p> <p>Objetivo 2: que el cliente este seguro de lo que necesita ya que cualquier cambio aumentaría costos y retrasos en su entrega.</p>

Roles y responsabilidades

Miembros del equipo	Rol	Responsabilidades
Rafael Venegas	Analista	Determinar cuáles son las necesidades y los objetivos, para reunir los requisitos que se deben cumplir en el desarrollo de software.
Daniel Hernández	diseñador	<ol style="list-style-type: none">1. Definir la organización de la estructura que se necesita para el desarrollo del software.2. Definir como se relacionan cada uno de los elementos
Guadalupe Pacheco Susana Jacobo	Programador y validadores	Realizar una traducción de todos los elementos del diseño que se prepararon en la etapa previa al lenguaje de programación.

Juan Manuel Corona	Consulta de prueba de sistema	Probar y ejecutar el código final y verificar su funcionamiento.
Luis Manuel López	Validador	Su objetivo principal será que los sistemas informáticos funcionen de manera más eficiente. Y realizar los cambios pertinentes, si es que son necesarios, y así poder dar por concluido el proyecto.

Calendario

NOMBRE DEL PROYECTO

Recursos empresariales

FECHA DE INICIO

2023-05-21

GERENTE DE PROYECTO

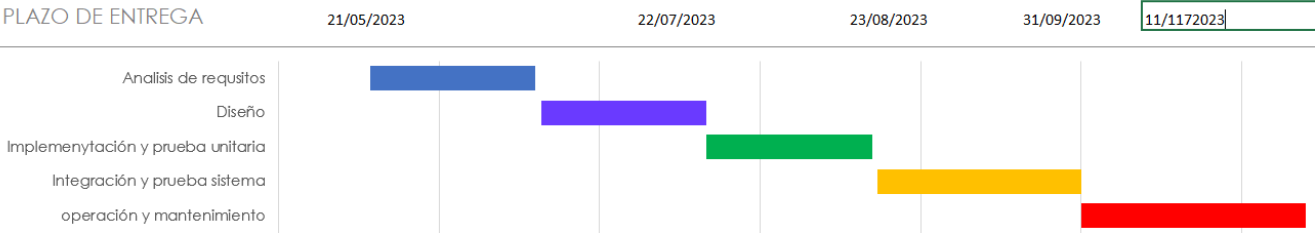
Pilar Barajas

FECHA FINAL

2023-11-11

TÍTULO DE LA FASE	LÍNEA DE TIEMPO			NOTAS DEL PROYECTO			
	EMPEZAR FECHA	FIN FECHA	DURACIÓN en días	HORARIO	PRESUPUESTO	RECURSOS	RIESGOS
Analisis de requisitos	2023-05-21	2023-06-21	31	8 horas por día	9,000	conocimiento de de lo que nesecita el cliente.	Requisitos no informados por el cliente
Diseño	2023-06-22	2023-07-22	31	7 horas por día	10,000	inspiración y conocimiento de diseño de interfaz	Escasez de recursos, factor de tiempo
Implemenytación y prueba unitaria	2023-07-23	2023-08-23	31	8 horas por día	10,000	Arquitectura y diseño teminada	Calculos incorrectos, tiempos de respuesta incorrectos.
Integración y prueba sistema	2023-08-24	2023-09-31	38	8 horas por día	8,000	Componentes implementados	Desconocimiento de la tecnologia base de proyecto
operación y mantenimiento	2023-10-01	2023-11-11	42	8 horas por día	10,000	Sistema integrado	Errorres no descubiertos.

Línea de tiempo



Control de versiones de software

El control de versiones es un control de sistemas que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, para poder recuperar versiones anteriores o versiones específicas más adelante, es un concepto clave para cualquier programador. El control de versiones es importante para dar seguimiento de los cambios en el código y mantener a cada miembro del equipo trabajando en la versión correcta.

Un método de software de control de versiones para todo el código y más utilizado es normalmente Git ya que es el más común y más utilizado.

Ventajas

- Sistema distribuido
- Rápido y ligero
- La integridad de la información está asegurada
- Permite flujos de trabajo muy flexibles
- Gratis.

Conclusión

La gestión de proyectos en campo laboral, reduce los costos del proyecto porque aumenta la eficiencia, mitiga los riesgos y optimiza los recursos. Incluso con el costo añadido de invertir en un gerente de proyecto, el objetivo que se persigue cuando se realiza una gestión de proyecto, es que se realice un producto original y que se cumpla una necesidad específica del cliente. Esta puede ser utilizada para generar nuevos servicios y productos.

Ya sean proyectos de desarrollo, de reorganización de cualquier otro tipo, hoy en día se puede afirmar que el software de gestión de proyecto es un elemento clave durante el control integral, con eficiencia y comodidad. Ya que por otro lado se facilita el registro de diferentes tareas y trabajadores implicados en el mismo dependiendo de las funcionalidades y necesidades del usuario. no en vano cumplir con todos los objetivos establecidos y la gestión eficiente de los proyectos son cosas esenciales.

Referencias

GanttPRO Project Management Blog. (2023, 18 mayo). *GanttPRO Project Management Blog*. Gantt

Chart GanttPRO Blog. <https://blog.ganttpro.com/>

ISOTools. (2023, 17 marzo). *ISO Software*. Software ISO. <https://www.isotools.us/>