

# TRABAJO PRÁCTICO N°1

## INGENIERÍA DE SOFTWARE III



### INTEGRANTES:

Luciano Moliterno - 40238958

Romano Sebastián Félix - 38636371

Ruina Oscar Cesar - 35639606

Hermida Pablo - 39287353

Imagine ser el gerente general / dueño de una organización de desarrollo de software que se encuentra en el nivel 3 de madurez del modelo CMM:

## 1) Describa los estándares de procesos de desarrollo que posee su organización.

### ❖ Requisitos:

Las actividades se definen con los clientes para determinar los requisitos de aprobación, en las cuales se determinarán primeras instancias para hacer revisiones y cambios de estas mismas, cómo definir y desarrollar los requisitos del software, definir los requisitos de interfaz y priorizar e integrar los requisitos del software.

En las cuales en esta primera etapa se definirán los siguientes documentos para su validación y aprobación al cliente:

- Especificación de requisitos del software.
- Requisitos del interfaz con el usuario.
- Requisitos del interfaz con el hardware.
- Requisitos del interfaz con el sistema

En los requisitos de interfaz se tendrán distintos diagramas, en las cuales serán técnicas orientadas a los procesos, técnicas orientadas a los datos, técnicas orientadas a los objetos, técnicas de prototipación en las cuales se tendrá en cuenta posibles cambios de acuerdo a las necesidades del cliente, en lo cual también se tendrá en cuenta estimaciones para los entregables, tanto en personal, esfuerzo y costos que se requerirán.

### ❖ Diseño

#### Procesos de Diseño.

Tienen que satisfacer los requisitos mediante una representación del sistema software, el diseño de sistema empieza desde un diseño de alto nivel que refleja los componentes y las interfaces que tiene el sistema, posteriormente se realiza un diseño de bajo nivel que detalla en profundidad lo que hace cada componente, las relaciones, los datos y restricciones y se definen las interfaces externas e internas.

#### **Procesos Diseño de alto nivel:**

En base a la especificación de requisitos se procede a modelizar el sistema, se establecen cuales van a ser los componentes y cómo se relacionan entre ellos, si se modeliza en base al paradigma de objetos en esta etapa se definen las clases y las relaciones entre ellas y sus atributos, también si se aplica se realiza el diseño de la base de datos, comprende un esquema que contiene las entidades de datos, sus atributos y relaciones. La salida de este proceso son diseños de alto nivel del sistema.

- Definir los componentes.
- Definir las interfaces externas
- Definir las interfaces internas.
- Definir las restricciones de cada componente.

#### **Procesos Diseño de bajo nivel:**

Con el/los diseños de alto nivel se procede a realizar el diseño de bajo nivel, en este paso se detalla lo que hace cada componente del sistema, se especifican los datos, se detalla lo que puede hacer y no puede hacer cada componente, se definen las interfaces externas( como interactúa el usuario con el software ) y las interfaces internas( como interactúan los componentes entre sí ), se describen los algoritmos a utilizar. Una vez terminado el diseño de bajo nivel se obtiene una descripción detallada del sistema, de la base de datos, de las interfaces, de los algoritmos.

- Detallar la función de cada componente.
- Detallar cuales son los datos.

- Detallar las restricciones de cada componente.
- Detallar las interfaces externas.
- Detallar las interfaces internas.
- Describir detalladamente los algoritmos a utilizar.

#### ❖ **Proceso de implementación:**

Implementación, pruebas y documentación

La **implementación** es parte del proceso en el que programamos el código para el proyecto de trabajo que está en relación de las demanda del software, en esta etapa se realizan las pruebas de caja blanca y caja negra.

Las **pruebas** de software son parte esencial del proceso de desarrollo del software. Esta parte del proceso tiene la función de detectar los errores de software lo antes posible.

Tipos de pruebas (hay más pero estos son los más importantes:

- Unit tests: las pruebas unitarias son a bajo nivel. Éste tipo de testing consiste en probar de forma individual las funciones y/o métodos (de las clases, componentes y/o módulos que son usados por nuestro software).
- Integration tests (pruebas de integración): verifican que los módulos y/o servicios usados por nuestra aplicación funcione en armonía cuando trabajan en conjunto.
- Functional tests (pruebas funcionales): se centran en los requerimientos de negocio de una aplicación.
- End-to-end tests (pruebas de punta a punta): replican el comportamiento de los usuarios con el software, en un entorno de aplicación completo.
- Regression testing (pruebas de regresión): verifican un conjunto de escenarios que funcionaron correctamente en el pasado, para asegurar que continúen así.
- Smoke testing (pruebas de humo): verifican la funcionalidad básica de una aplicación.
- Acceptance testing (pruebas de aceptación): son pruebas formales, ejecutadas para verificar si un sistema satisface sus requerimientos de negocio.
- Performance testing (pruebas de rendimiento): verifican cómo responde el sistema cuando éste se encuentra bajo una alta carga.

La **documentación** del diseño interno del software con el objetivo de facilitar su mejora y su mantenimiento se realiza a lo largo del proyecto. Esto puede incluir la documentación de un API, tanto interior como exterior. Prácticamente es como una receta de cocina

#### ❖ **Operación y soporte:**

La **fase de Operación** es, sin duda, la más crítica entre todas. La percepción que los clientes y usuarios tengan de la calidad de los servicios prestados depende en última instancia de una correcta organización y coordinación de todos los agentes involucrados.

Los principales objetivos de la fase de Operación incluyen:

- Operar el sistema.

- Proveer de asistencia técnica y consultas.
- Mantener el histórico de peticiones de soporte.

### La fase de Soporte

- Proceso de Gestión de la Documentación del Software: consiste en registrar la documentación producida por un proceso o por las actividades del ciclo de vida de un proyecto, para que dichos documentos sean accesibles por miembros de la empresa o usuarios implicados.
- Proceso de Gestión de la Configuración del Software: establecer y mantener de la integridad de todos los productos de trabajo de un proceso o proyecto y hacerlos disponibles para las partes involucradas.
- Proceso de Aseguramiento de Calidad del Software: proporciona la seguridad necesaria para que los productos y procesos software implicados en los proyectos sean conformes a los requisitos especificados y se ajustan a los planes establecidos.
- Proceso de Verificación del Software: determina que los productos software cumplen con los requisitos especificados para ellos.
- Proceso de Validación del Software: determina si los requisitos especificados para un determinado proyecto cumplen su uso específico.
- Proceso de Revisión del Software: ayudar a asegurar que el desarrollo del software satisface a las partes involucradas.
- Proceso de Resolución de Problemas del Software: proveer mecanismos para la creación de procesos capaces de resolver problemas y tomar acciones correctivas para remover nuevos problemas detectados.

### ❖ Mantenimiento:

- Mantenimiento perfectivo
- Mantenimiento adaptativo
- Mantenimiento estructurado

Las peticiones de mantenimiento **adaptativo** y de mantenimiento **perfectivo** siguen un camino diferente. las adaptaciones se evalúan y se clasifican (se asigna prioridad) antes de situarlas en una cola de acciones de mantenimiento. Las mejoras siguen la misma evaluación. Sin embargo, no todas las peticiones de mejoras se llevan a cabo. La estrategia comercial, los recursos disponibles, la línea de los productos de software actuales y futuros y otros muchos aspectos pueden hacer que se rechace una petición de mejora. Las mejoras que sean aceptadas también se sitúan en una cola de mantenimiento. Se establece la

prioridad de cada petición y se planifica el trabajo requerido como si se tratara de otro esfuerzo de desarrollo más (para todos los propósitos e intenciones así es). El mantenimiento adaptativo se aplica cuando los cambios del entorno externo precipitan las modificaciones del software. El mantenimiento perfectivo incorpora mejoras solicitadas por la comunidad de usuarios.

Si sólo se dispone del código fuente como elemento de configuración, la actividad de mantenimiento comienza con una dolorosa evaluación del código, a menudo complicada por la pobre documentación interna. Características como la estructura del programa, las estructuras de datos globales, las interfaces del sistema, el rendimiento y las limitaciones del diseño, son difíciles de descubrir y, frecuentemente, mal interpretadas. Es difícil asegurar cuáles son las ramificaciones de los cambios realizados finalmente sobre el código. Es imposible llevar a cabo pruebas de regresión ya que no existe ningún registro de pruebas. Lo que hacemos es un mantenimiento **no estructurado**, lo cual implica un mayor esfuerzo asociado a todo software que no haya sido desarrollado mediante una metodología bien definida.

Si existe una completa configuración del software, la tarea de mantenimiento comienza con una evaluación de la documentación del diseño. Se determinan las importantes características estructurales, de rendimiento y de interfaz del software. Se estudia el impacto de las correcciones o modificaciones requeridas y se traza un plan de actuación. Se modifica el diseño y se revisa. Se desarrolla nuevo código fuente, se realizan pruebas de regresión mediante la información contenida en la especificación de prueba y se vuelve a lanzar el software. Esta secuencia de sucesos constituye el mantenimiento **estructurado** y aparece como resultado de una anterior aplicación de una metodología de ingeniería del software. Aunque la existencia de una configuración del software no garantiza un mantenimiento libre de problemas, sí que reduce la cantidad de esfuerzo requerido y mejora la calidad general del cambio o de la corrección.

## 2) Describa los estándares de procesos de gestión que posee su organización.

### PROCESO DE INICIACIÓN DE PROYECTO

Abarca aquellas actividades de creación de la estructura del proyecto. Se define el ciclo de vida del software para este proyecto y se establecen los planes para su gestión.

Las actividades a realizar serán las siguientes en las cuales cumplirán con la iniciación del proyecto:

Crear el documento que formalice la iniciación del proyecto(charter).

Definir los recursos humanos del proyecto.

Procesos para identificar, definir, combinar, unificar y coordinar los diversos procesos y actividades de dirección del proyecto.

- Desarrollar el acta de constitución del proyecto.
- Desarrollar el plan para la Dirección del proyecto.
- Dirigir y gestionar el trabajo del proyecto.
- Gestionar el conocimiento del proyecto.
- Monitorear y Controlar el trabajo del proyecto.
- Realizar el control integrado de Cambios.
- Cerrar el proyecto o fase.

Definir:

- WBS: Es un documento que descompone jerárquicamente todo el trabajo del proyecto en entregables. Con el objetivo de Subdividir el trabajo en componentes más pequeños y manejables.
- Objetivos: lo que el proyecto debe cumplir para alcanzar todos las metas.
- Riesgos: como oportunidades, establecer porcentajes de ocurrencia de los riesgos y planificar planes de acción y de contingencia.
- Cronograma: Incluye los procesos requeridos para administrar la finalización del proyecto a tiempo. Es un entregable que presenta actividades vinculadas con fechas planificadas,
- Costos: con la wbs calcular el tiempo que tomaría realizar las actividades y en base a eso estimar un costo.

Técnicas de Análisis de Camino Crítico (CPM).

Técnicas de Análisis PERT.

Técnicas de Diagrama de GANTT.

## **PROCESO DE SEGUIMIENTO Y CONTROL DE PROYECTO**

Aprender algunas de las herramientas que permiten controlar el avance del proyecto.

Elaborar informes de avance que permitan interactuar con los patrocinadores del proyecto.

En las cuales estas estarán marcados por indicadores:

- Son instrumentos que miden los resultados de los procesos en forma cuantitativa.
- Refleja cuáles fueron las consecuencias de acciones tomadas en el pasado en el marco de un proyecto.
- La idea es que estos indicadores sienten las bases para acciones a tomar en el presente y en el futuro.
- Permiten determinar el grado de cumplimiento de los objetivos

También controlar los riesgos para poder responder de manera efectiva con las posibles situaciones que puedan surgir, Incrementar las probabilidades de ocurrencia y el impacto de los eventos positivos y reducir las probabilidades y el impacto de los eventos adversos al proyecto.

Un riesgo puede aparecer en cualquier proceso o sobre cualquiera de los objetivos del proyecto

## **PROCESO DE GESTIÓN DE LA CALIDAD DEL SOFTWARE**

Su objetivo es la planificación y administración de las acciones necesarias para proveer una confianza adecuada en la calidad de los productos de software, es decir, satisfagan los requisitos técnicos establecidos.

Un plan de calidad incluye:

1. Introducción del producto una descripción del producto, la pretensión de su mercado y las expectativas de calidad para el producto.
2. Planes del producto indican las fechas de entrega críticas y las responsabilidades para el producto, junto con planes para distribución y servicio al producto.
3. Descripciones de procesos describen los procesos y estándares de desarrollo y servicio que deben usarse para diseño y gestión del producto.

4. Metas de calidad las metas y los planes de calidad para el producto, incluyendo una identificación y justificación de los atributos esenciales de calidad del producto.
5. Riesgos y gestión del riesgo los riesgos clave que pueden afectar la calidad del producto y las acciones a tomar para enfrentar dichos riesgos.

### 3) Describa un plan de acción para alcanzar el nivel 4 de madurez.

El plan de acción para alcanzar el nivel 4 de madurez se requerirá crear un portfolio de todos los programas y proyectos que ha realizado la organización para tener un control cuantitativo de todos los procesos, de esta manera se podrían evaluar y estimar de mejor manera los proyectos de software.

Se documentará cada proceso hecho por la organización, se establecen los procesos que mejores resultados han dado como también aquellos conocimientos nuevos descubiertos por los miembros de la organización y se crea un estándar para que lo use la organización como también integrantes que lo compongan.

### 4) Alcanzado el nivel 4, describa los procedimientos de medición incorporados y estándares.

En el nivel gerenciado, la organización establece métricas para productos y procesos y mide los resultados. Los proyectos llevan a cabo controles sobre sus productos y procesos, estrechando la variación en el desempeño de su proceso de forma tal de caer dentro de límites aceptables.

Las variaciones significativas en dicho desempeño pueden ser distinguidas de las variaciones al azar. Los riesgos involucrados en movilizar tecnologías de nuevos productos, procesos y mercados son conocidos y administrados cuidadosamente.

El proceso de desarrollo es predecible porque es medido y opera dentro de límites controlables. Este nivel de capacidad de proceso permite a una organización predecir tendencias en la calidad del proceso y el producto dentro de las fronteras cuantitativas de estos límites. Cuando estos límites son excedidos, una acción se ejecuta para corregir la situación. Como resultado, los productos son de una calidad predeciblemente alta.

### 5) ¿Qué plan de acción seguiría para alcanzar el nivel 5 de madurez?

Ya implementando todos los servicios estos pasarían un proceso de mantenimiento para su mejora continua en base a las necesidades que vayan surgiendo en el tiempo. De esta manera la organización podrá ir desarrollando sus procesos, con el fin de incorporar nuevo personal en generar innovaciones y así mejorar la calidad del producto, como establecer equipos dedicados solamente a una sola rama de software y la mejora continua de los proyectos actuales y futuros proyectos que brinden oportunidades a la organización de seguir creciendo.