Proyecto Certificador de Desarrollo de SW 1

Tema Nº3:

Metodología de Desarrollo de Software.

Indicador de logro Nº3:Identifica las metodologías de desarrollo de software para construir proyectos informáticos.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº3:**

METODOLOGÍAS DE DESARROLLO DE SOFTWARE

MARCO TEÓRICO

1. Metodología de Desarrollo de Software
   1. **Metodología RUP**

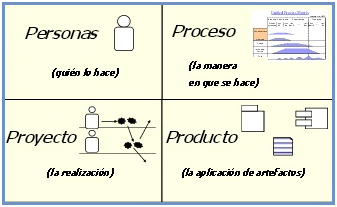
RUP es una metodología que tiene como objetivo ordenar y estructurar el desarrollo de software en la cual se tienen un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema.

La metodología de desarrollo RUP por sus siglas en inglés (Proceso de Desarrollo Unificado) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado - UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos

RUP proporciona a un profesional de desarrollo de software un entorno de proceso configurable basado en estándares.

RUP es un proceso de Ingeniería de Software que proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de un desarrollo organizado. Las actividades crean y mantienen modelos, que son representaciones, semánticamente ricas, de un sistema software en desarrollo.

Para capturar la dimensión temporal de un proyecto, RUP divide el proyecto en cuatro fases diferentes:

* Iniciación o Diseño: Énfasis en el alcance del sistema;
* Preparación: Énfasis en la arquitectura;
* Construcción: Énfasis en el desarrollo;
* Transición: Énfasis en la aplicación.

RUP se basa también en las 4 Ps:

Cuando se desea realizar una gestión

Adecuada eficaz y eficiente en la Gestión de

Proyectos de Software, es necesario que se

ponga en funcionamiento cuatro

características muy importantes en esta

gestión, las cuatro P:

Personal, Producto, Proceso y Proyecto.

**Personal**

El proceso de software está integrado por participantes, líderes de equipo, arquitectos, desarrolladores, ingenieros de prueba, personal de gestión, usuarios, clientes, etc. Los participantes se los puede clasificar en cinco categorías:

* Gestores ejecutivos: Definen los aspectos del negocio.
* Gestores del proyecto: Planifican, motivan, organizan y controlan a los profesionales que construyen el software.
* Profesionales: Proporcionan las habilidades técnicas necesarias.
* Clientes: Especifican los requerimientos.
* Usuarios finales: Interactúan con el software.

**Producto**

Se denomina productos a todos aquellos artefactos que se creen durante la vida del proyecto, modelos, códigos, ejecutable, documentación, diagramas UML, bocetos de la interfaz de usuario, prototipos, componentes, planes de prueba, ingeniería y gestión, colección de modelos, modelos de casos de uso, análisis, diseño, despliegue, implementación y prueba. Antes de planear un proyecto, se deben establecer los objetivos y el alcance que tendrá el proyecto, además de sus restricciones, herramientas, técnicas y planes de gestión. Con una buena planificación se puede estimar el tiempo que tomará desarrollar o construir el producto y redimensionar el valor cuantitativo del mismo.

**Proceso**

Se denomina proceso al conjunto de actividades que se realizan para crear el producto (Plantilla para crear el proyecto). El proceso se define en términos de flujos de trabajo (conjunto de actividades), se identifican trabajadores y artefactos, además de que se utilizan diagramas de actividad de UML para describir los flujos de trabajo. El equipo de desarrollo debe elegir el proceso adecuado y que le permita obtener una solución o producto que satisfaga las necesidades o requerimientos del cliente.

Seleccionado el modelo de procesos, se desarrolla una planeación preliminar del proyecto basado en las actividades del marco de trabajo. Esta planeación comienza con la combinación del producto y el proceso. Cuando el equipo de desarrollo de software ha definido correctamente el modelo de proceso, debe de asegurarse de que este sea flexible y adecuado para el proyecto.

**Proyecto**

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. Es un elemento organizativo de gestión que establece una secuencia de cambios, por el cual va evolucionando diariamente. El mismo establece una serie de iteraciones e hitos dentro de los cuales se desarrollan una serie de casos de usos o requisitos.

Debemos tener en cuenta muchos parámetros para realizar un proyecto, estos son:

1. El personal de software no entiende las necesidades de los clientes.
2. El ámbito del producto está mal definido.
3. Los cambios se gestionan mal.
4. La tecnología elegida cambia.
5. Las necesidades comerciales cambian.
6. Los plazos de entrega no son realistas.
7. Los usuarios se resisten a la utilización del software.
8. El equipo del proyecto carece de personal con las habilidades apropiadas.
9. Los gestores evitan las mejores prácticas y las lecciones aprendidas.
   1. Artefactos

En Rup en cada una de sus fases se realizan y utilizan una serie de artefactos para saber mejor la función y estructura del software.

Un artefacto puede ser:

* **Un documento:** como un Caso de Negocio o un documento de la arquitectura del Software.
* **Un modelo:** como un modelo de caso de uso.
* **Un elemento de un modelo:** como una sola clase de todo el diagrama de clases.

Los artefactos por cada fase de desarrollo son:

Inicio:

* Documento Visión
* Especificación de Requerimientos

Elaboración:

* Diagramas de caso de uso.

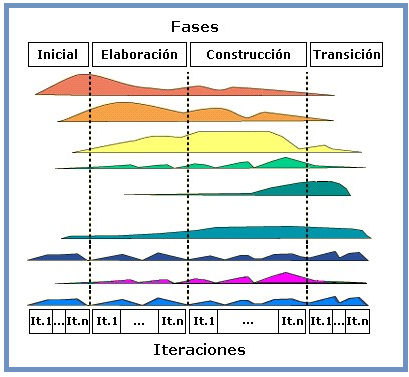
Construcción:

Descripción de la arquitectura que trabaja con las siguientes vistas:

* Vista Lógica:
  + Diagrama de clases
  + Modelo E-R (Si el sistema así lo requiere)
* Vista de Implementación:
  + Diagrama de Secuencia
  + Diagrama de estados
  + Diagrama de Colaboración
* Vista Conceptual:
  + Modelo de dominio
* Vista física:
  + Mapa de comportamiento a nivel de hardware.
  + Diseño y desarrollo de casos de uso, o flujos de casos de uso arquitectónicos.
  + Pruebas de los casos de uso desarrollados, que demuestran que la arquitectura documentada responde adecuadamente a requerimientos funcionales y no funcionales.

Transición:

* Pruebas finales de aceptación.
* Puesta en producción.
* Estabilización.
  1. El ciclo de vida RUP



Es una implementación del desarrollo

en espiral. Fue creado ensamblando

los elementos en secuencias

semiordenadas.

El ciclo de vida organiza las tareas en

fases e iteraciones.

RUP divide el proceso en cuatro fases

dentro de las cuales se realizan varias

iteraciones en número variable según

el proyecto.

|  |  |
| --- | --- |
|  | Fase de inicio: Tiene como objetivo establecer un acuerdo entre todos los interesados acerca de los objetivos del proyecto.  Se hacen mayor énfasis en actividades de modelado del negocio y de requisitos. |
|  | Fase de elaboración: Tiene como objetivo establecer la arquitectura base del sistema para proveer bases estables para el esfuerzo de diseño e implementación en la siguiente fase.  Abarcan más los flujos de trabajo de requisitos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base a la arquitectura. |
|  | Fase de construcción: Tiene como objetivo clarificar los requerimientos faltantes y completar el desarrollo del sistema basado en la arquitectura base. Se seleccionan algunos casos de uso, se refinan su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan iteraciones hasta que se termine la implementación de la nueva versión del software. |
|  | Fase de transición: Tiene como objetivo enfocarse en asegurar que el software esté disponible para sus usuarios  Se garantiza que se tiene un producto preparado para su entrega a la comunidad de usuarios  . |

* 1. Metodología EUP

Proceso unificado de empresa (EUP). es una metodología de desarrollo de software que ayuda a las empresas a crear software de manera estructurada y organizada. Es una extensión de la Proceso racional unificado (RUP), agregando dos nuevas fases de desarrollo: Producción y Retiro. Como el RUP incluye cuatro fases, el EUP consta de seis fases:

1. Comienzo - Se plantea la idea del proyecto. El equipo de desarrollo determina si vale la pena seguir el proyecto y qué recursos se necesitarán.
2. Elaboración - La arquitectura del proyecto y los recursos necesarios se evalúan más a fondo. Los desarrolladores consideran posibles aplicaciones del software y los costos asociados con el desarrollo.
3. Construcción - El proyecto está desarrollado y completado. El software está diseñado, escrito y probado.
4. Transición - El software se lanza al público. Los ajustes o actualizaciones finales se realizan en función de los comentarios de los usuarios finales.
5. Producción - El software se mantiene útil y productivo después de ser lanzado al público. Los desarrolladores se aseguran de que el producto continúe ejecutándose en todos los sistemas compatibles y el personal de asistencia proporciona asistencia a los usuarios.
6. Jubilación - El producto se retira de la producción, a menudo llamado "desmantelamiento". Puede ser reemplazado o simplemente ya no es compatible. El lanzamiento de una nueva versión de software a menudo coincide con la fase de retiro de una versión anterior.

**Actividad:**

Ingresa a la plataforma virtual, luego desarrolla la siguiente actividad propuesta:

1. **CUESTIONARIO TÉCNICO**
2. ¿Qué es y para qué sirve el RUP?
3. ¿Por qué utilizar RUP?
4. ¿Cómo se desarrolla un proyecto basado en RUP?

1. **CONCLUSIONES DE LA EXPERIENCIA**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_