

6 Anexos:

6.1 Definición de Rup:

Es un producto del proceso de ingeniería de *software* que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo.

6.2 Características:

- ✓ Forma disciplinada de asignar tareas y responsabilidades.
- ✓ Pretende implementar las mejores prácticas en Ingeniería de Software
- ✓ Desarrollo iterativo
- ✓ Administración de requisitos
- ✓ Uso de arquitectura basada en componentes
- ✓ Control de cambios
- ✓ Modelado visual del software
- ✓ Verificación de la calidad del software

6.3 Fases:

6.3.1. Ciclo de Vida de RUP

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades.

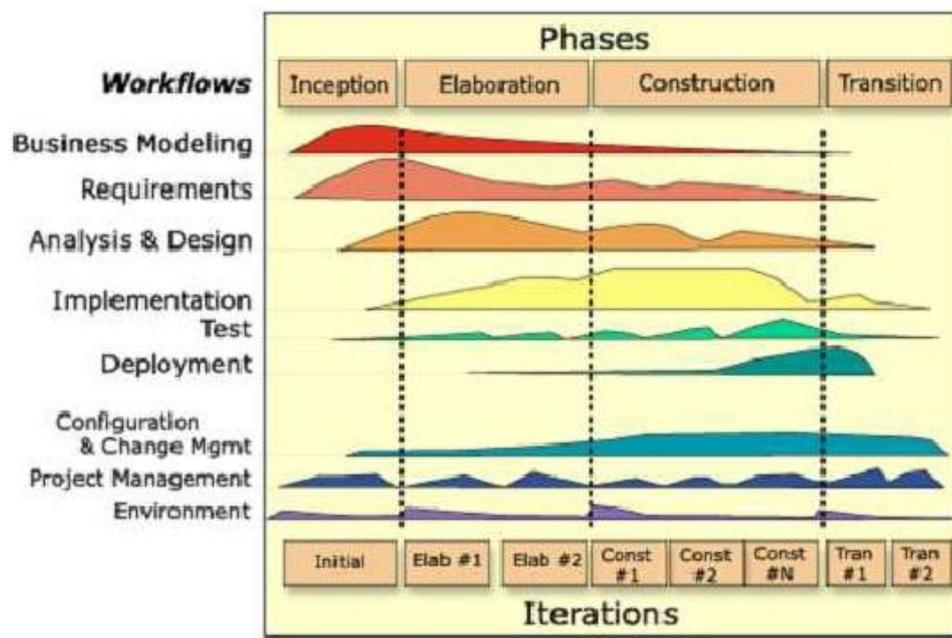


Fig. 6.3.1.1 Cuadro de las Fases de la Metodología RUP

6.3.1.1 Fase de Inicio:

Durante la fase de inicio las iteraciones hacen poner mayor énfasis en actividades modelado del negocio y de requisitos.

Modelado del negocio

En esta fase el equipo se familiarizará más al funcionamiento de la empresa, sobre conocer sus procesos.

- Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado
- Entender el problema actual en la organización objetivo e identificar potenciales mejoras.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo

Requisitos

En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos.

- Establecer y mantener un acuerdo entre clientes y otros stakeholders sobre lo que el sistema podría hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

6.3.1.2 Fase de Elaboración:

En la fase de elaboración, las iteraciones se orientan al desarrollo de la base line de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la base line de la arquitectura.

Análisis y Diseño

En esta actividad se especifican los requerimientos y se describen sobre como se van a implementar en el sistemas.

- Transformar los requisitos al diseño del sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación

6.3.1.3 Fase de Construcción:

Implementación

Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. El resultado final es un sistema ejecutable.

- Planificar qué subsistemas deben ser implementados y en que orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en que orden implementa los elementos del subsistema.
- Si encuentra errores de diseño, los notifica.
- Se integra el sistema siguiendo el plan

Pruebas

Este flujo de trabajo es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.
- Verificar que los requisitos tengan su apropiada implementación.

Despliegue

Esta actividad tiene como objetivo producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.
- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Formar a los usuarios y al cuerpo de ventas.
- Migrar el software existente o convertir bases de datos

6.3.1.4 Fase de Transición: sdfa

El propósito de la fase de la transición es justamente la transición del producto de software al ambiente de producción . Una vez que el producto se haya entregado al usuario final, surgen algunos temas que llevan al desarrollo de nuevas versiones, a corregir errores, o a terminar algunas características que habían sido postpuestas.

Se ingresa a esta fase cuando el producto está lo suficientemente maduro para comenzar a pasar a producción . Esto requiere que un cierto subconjunto del sistema se encuentre en un nivel aceptable de la calidad y que la documentación del usuario está disponible de modo que la transición proporcione resultados positivos para todas las partes.

Esto incluye:

- La “prueba beta” para validar el nuevo sistema contra las expectativas del usuario.
- Operación en paralelo con un sistema anterior que el nuevo sistema esté sustituyendo.
- La conversión de las bases de datos operacionales.
- Entrenamientos y capacitación de los usuarios y la gente de mantenimiento.
- Lanzar el producto a los equipos de marketing, distribución y ventas

La fase de transición se centra en las actividades requeridas para poner el software en manos de los usuarios. Típicamente, esta fase incluye varias iteraciones, incluyendo lanzamientos beta, lanzamientos de disponibilidad general, así como la reparación de errores y el lanzamiento de versiones mejoradas. Un esfuerzo considerable se realiza en la documentación orientada al usuario final, en entrenar a los mismos, en brindar apoyo en las primeras etapas del uso, y en reaccionar al feedback que generen los mismos usuarios. En este punto del ciclo de vida, sin embargo, el feedback del usuario se debe centrar sobre todo en el ajuste fino del producto, la configuración, instalación, y a las cuestiones de usabilidad. Esta fase puede variar -según el proyecto- de ser muy simple a muy compleja. Por ejemplo una nueva versión de un procesador de texto puede ser muy simple, mientras que substituir el sistema de control de tráfico aéreo de un país sería muy complejo.

6.4 Disciplinas:

6.4.1 Disciplina de desarrollo de RUP.

- Determina las etapas a realizar durante el proyecto de creación del software.
- Ingeniería o modelado del negocio: Analizar y entender las necesidades del negocio para el cual se está desarrollando el software.
- Requisitos: Proveer una base para estimar los costos y tiempo de desarrollo del sistema.
- Análisis y diseño: Trasladar los requisitos analizados anteriormente a un sistema automatizado y desarrollar una arquitectura para el sistema.
- Implementación: Crear software que se ajuste a la arquitectura diseñada y que tenga el comportamiento deseado.
- Pruebas: Asegurarse de que el comportamiento requerido es correcto y que todo lo solicitado está presente.
- Despliegue: Producir distribuciones del producto y distribuirlo a los usuarios.

6.4.2 Disciplina de soporte RUP.

- Determina la documentación que es necesaria realizar durante el proyecto.
- Configuración y administración del cambio : Guardar todas las versiones del proyecto.
- Administración del proyecto : Administrar los horarios y recursos que se deben de emplear.
- Ambiente : Administrar el ambiente de desarrollo del software.
- Distribución : Hacer todo lo necesario para la salida del proyecto.