

Proceso Unificado para Desarrollo de Software (RUP)

1. Generalidades y aportes:

Proceso de desarrollo propuesto por "Rational Software Corporation" resultado del esfuerzo de las tres últimas décadas en desarrollo de software y de la experiencia de sus creadores Ivar Jacobson, Grady Booch y James Rumbaugh.

2. Orígenes

El antecedente más importante lo ubicamos en 1967 con la Metodología Ericsson (Ericsson Approach), ésta es una aproximación de desarrollo basada en componentes, que introdujo el concepto de caso de uso; entre los años de 1987 a 1995 Jacobson funda la compañía "Objectory AB" y lanza el proceso de desarrollo Objectory (abreviación de Object Factory), posteriormente en 1995 "Rational Software Corporation" adquiere "Objectory AB" y es entre 1995 y 1997 que se desarrolla "Rational Objectory Process (ROP)" fruto del encuentro y evolución de Objectory 3.8 y la Metodología Rational (Rational Approach) que adopta por primera vez UML como lenguaje de modelamiento.

A principios de los noventas, la guerra de los métodos hizo evidente la necesidad de unificar criterios, es así como Grady Booch autor del método Booch y James Rumbaugh (desarrollador para General Electric) se unieron en Rational en 1994, después en 1995 se une Jacobson y gracias al esfuerzo de varias compañías y metodologistas evolucionó UML hasta ser un estándar en 1997, el cual es adoptado en todos los modelos del ROP. Desde ese entonces y a la cabeza de Booch, Jacobson y Rumbaugh, Rational ha desarrollado e incorporado diversos elementos para expandir el ROP, destacándose especialmente el flujo de trabajo conocido como modelamiento del negocio, es así como en junio del 1998 se lanza Rational Unified Process 5.0 evolucionado hasta el momento de elaboración de este documento bajo el nombre de RUP. La evolución y orígenes de este proceso de desarrollo se puede visualizar mejor en la figura 2.1

3 Características principales

a.) **Guiado/Manejado por casos de uso:** La razón de ser de un sistema software es servir a usuarios ya sean humanos u otros sistemas; un caso de uso es una

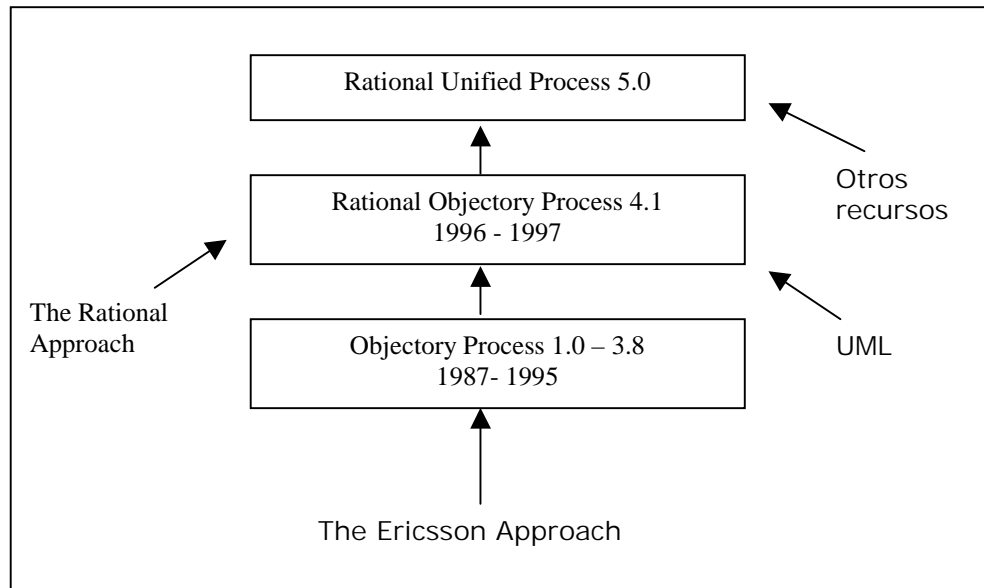


Figura 1 Orígenes del proceso Unificado[JBR99]

facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.

b.) Centrado en arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.

c.) Iterativo e Incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un miniproyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

- d.) **Desarrollo basado en componentes:** La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o que se desarrollan y maduran sus componentes.
- e.) **Utilización de un único lenguaje de modelamiento:** UML es adoptado como único lenguaje de modelamiento para el desarrollo de todos los modelos.
- f.) **Proceso Integrado:** Se establece una estructura que abarque los ciclos, fases, flujos de trabajo, mitigación de riesgos, control de calidad, gestión del proyecto y control de configuración; el proceso unificado establece una estructura que integra todas estas facetas. Además esta estructura cubre a los vendedores y desarrolladores de herramientas para soportar la automatización del proceso, soportar flujos individuales de trabajo, para construir los diferentes modelos e integrar el trabajo a través del ciclo de vida y a través de todos los modelos.

4 Modelo de Referencia Organizacional

Esta metodología propone un modelo para la organización de personal que se resume en la tabla 1.

CARGOS/POSICIONES	TAREAS ASIGNADAS
Gestor del proyecto	Establecer Condiciones de Trabajo
Analista del sistema	Encontrar Actores y Casos de Uso Estructurar el Modelo de Casos de Uso
Arquitecto del sistema	Priorizar los Casos de Uso Efectuar el Análisis Arquitectural Efectuar el Diseño Arquitectural Efectuar la Implementación Arquitectural
Especificador de casos de uso	Detallar un Caso de Uso
Diseñador de interfaz de usuario	Prototipar una Interfaz de Usuario
Ingeniero de casos de uso	Analizar un Caso de Uso Diseñar un Caso de Uso
Ingeniero de componentes	Analizar una Clase Analizar un Paquete Diseñar una Clase Diseñar un Subsistema Implementar un Subsistema Implementar una Clase Realizar una Prueba de Unidad Implementar una Prueba
Integrador del sistema	Integrar el Sistema
Ingeniero de pruebas	Planear las Pruebas Diseñar las Pruebas Evaluar las Pruebas
Verificador de integración	Realizar una Prueba de Integración
Verificador del sistema	Realizar las Pruebas del Sistema

Tabla1 Conformación del equipo humano de un proyecto según RUP

5 Fases en el ciclo de Desarrollo

Este proceso de desarrollo considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases que se describirán a continuación, la figura 2 muestra las fases de desarrollo y los diversos flujos de trabajo involucrados dentro de cada fase con una representación gráfica en cual de los flujos se hace mayor énfasis según la fase, cabe destacar el flujo de trabajo concerniente al negocio.

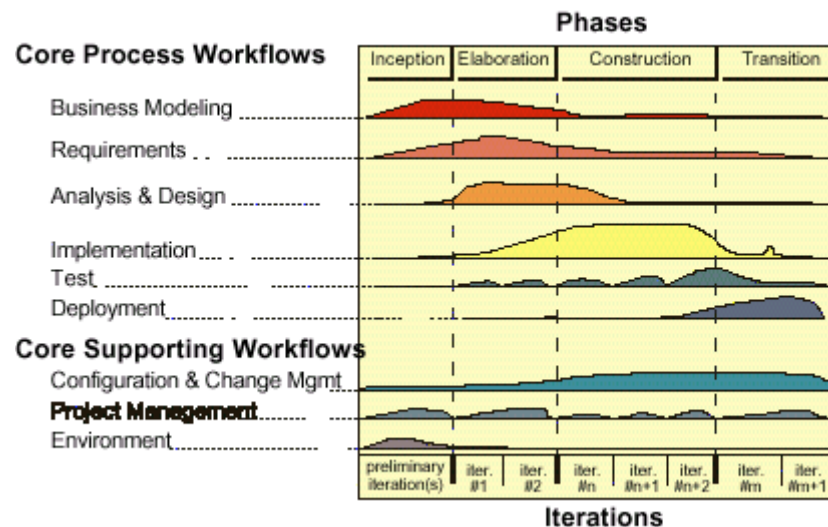


Figura 2. Fases de desarrollo de RUP [Ra00]

Fase 1 :Preparación Inicial ("Inception")

Su objetivo principal es establecer los objetivos para el ciclo del vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber qué se cubrirá y delimitar el alcance del proyecto.

Fase 2 : Preparación Detallada ("Elaboration")

Su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

Fase 3: Construcción ("Construction")

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado versión beta.

Fase 4 :Transición ("Transition")

Su objetivo principal es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos.

6 Subproductos

La tabla 2 muestra los subproductos (denominados en RUP "artifacts" (artefactos)) claves de cada una de las fases de desarrollo.

FASE	Subproductos
PREPARACIÓN INICIAL	<ul style="list-style-type: none"> 1.- Alcance del Sistema 1.- Lista de Características 2.- Modelo del Dominio o Modelo del Negocio (1ª. versión) 3.- Modelo de Casos de Uso, Modelo de Análisis y Modelo de Diseño (1ª. versión) 4.- Requerimientos Suplementarios (1ª. Versión) 2.- Arquitectura Inicial (propuesta) 3.- Lista Inicial de Riesgos (riesgos críticos más importantes) y Lista Priorizada de los Casos de Uso 4.- Prototipo para Validación de Conceptos (prototipo de descarte) 5.- Entorno de Desarrollo Configurado (proceso y herramientas) (configuración inicial) 6.- Plan Inicial del Proyecto 7.- Caso Inicial del Negocio (1ª. versión) (contexto del negocio y criterios de éxito) (costo, tiempos, calidad, utilidades)
PREPARACIÓN DETALLADA	<ul style="list-style-type: none"> 1.- Contexto del Sistema (Modelo del Dominio o Modelo del Negocio preferiblemente completo) 2.- Captura del 80% de los Requerimientos Funcionales <ul style="list-style-type: none"> 1.- Modelo de Casos de Uso (aprox. el 80%) y Modelo de Análisis (realización de los casos de uso más significativos) 2.- Modelo de Diseño, Modelo de Despliegue y Modelo de Implementación (menos del 10%) 3.- Niveles para los Atributos de Calidad y Requerimientos Suplementarios Actualizados 4.- Manual Preliminar de Usuario 3.- Arquitectura de Referencia (línea de base) (descripción de las vistas arquitecturales de los modelos del sistema) 4.- Lista Actualizada de Riesgos (críticos y significativos) y Riesgos Críticos Mitigados 5.- Plan del Proyecto para las fases de Construcción y Transición 6.- Entorno de Desarrollo Adecuado (proceso y herramientas) 7.- Caso del Negocio Completo (y "Contrato" o declaración del negocio)

CONSTRUCCIÓN	<ul style="list-style-type: none">1.- Modelos Completos (Casos de Uso, Análisis, Diseño, Despliegue e Implementación)2.- Arquitectura Íntegra (mantenida y mínimamente actualizada)3.- Riesgos Presentados Mitigados4.- Plan del Proyecto para la fase de Transición5.- Manual Inicial de Usuario (con suficiente detalle)6.- Prototipo Operacional – beta7.- Caso del Negocio Actualizado
TRANSICIÓN	<ul style="list-style-type: none">1.- Prototipo Operacional2.- Documentos Legales3.- Caso del Negocio Completo4.- Línea de Base del Producto completa y corregida que incluye todos los modelos del sistema5.- Descripción de la Arquitectura completa y corregida6.- Manuales para Usuario Final, Operador y Administrador del Sistema, y Materiales para Entrenamiento

Tabla 2 Subproductos del RUP

En construcción -----