

Modelo de ingeniería de datos RUP

Rodríguez Rangel, Marco Antonio & Pérez
Rodríguez, Andrés Daniel
Maestría en Ciencia de datos
Universidad de la Ciudad de Aguascalientes
Aguascalientes, México
andres.perez@alumnos.ucags.edu.mx

Resumen — El Proceso Unificado de Rational (RUP) es una metodología de desarrollo de software que abarca desde la concepción hasta la entrega de sistemas funcionales, distinguiéndose por su flexibilidad y adaptabilidad. Es esencial para profesionales en ingeniería de datos y desarrollo de software, al proporcionar un marco estructurado que permite la colaboración, una planificación efectiva y una entrega gradual. Este artículo explora sus fases, roles clave, flujos de trabajo, arquitectura y diseño en sistemas de ingeniería de datos, así como la gestión de cambios y la calidad del software, incluyendo casos del mundo real y comparaciones con otros enfoques, con el objetivo de reflexionar sobre su relevancia continua en este campo en constante evolución.

Palabras clave: Proceso Unificado de Rational (RUP), desarrollo de software, ingeniería de datos, roles y responsabilidades, flujo de trabajo.

Abstract — The Rational Unified Process (RUP) is a software development methodology that spans from conception to delivery of functional systems, distinguished by its flexibility and adaptability. It is essential for data engineering and software development professionals by providing a structured framework that enables collaboration, effective planning, and gradual delivery. This article explores its phases, key roles, workflows, architecture and design in data engineering systems, as well as change management and software quality, including real-world cases and comparisons with other approaches, with the aim of reflect on its continued relevance in this ever-evolving field.

Keywords: Rational Unified Process (RUP), software development, data engineering, roles and responsibilities, workflow.

I. INTRODUCCIÓN

El Proceso Unificado de Rational (RUP) es un enfoque de desarrollo de software que abarca desde la concepción hasta la entrega de sistemas funcionales, destacando por su flexibilidad y adaptabilidad [1]. Es esencial para profesionales en ingeniería de datos y desarrollo de software, al proporcionar un marco estructurado que permite la colaboración, planificación efectiva y entrega gradual. El artículo explorará sus fases, roles clave, flujos de trabajo, con el objetivo de reflexionar sobre su relevancia continua en este campo en constante evolución.

II. FASES DEL PROCESO RUP

El Proceso Unificado de Rational (RUP), reconocido como un marco de trabajo esencial en el desarrollo de software, se despliega en cuatro fases consecutivas que engloban la totalidad del ciclo de vida del proyecto: Inicio, Elaboración, Construcción y Transición [2].

A. Inicio: Delimitación del Alcance y Propósitos del Proyecto

La fase inaugural, denominada Inicio, implica la definición precisa de los objetivos del proyecto, la identificación temprana de los riesgos inherentes y la evaluación de la viabilidad técnica y económica. En el contexto de la ingeniería de datos, esta etapa engloba la identificación de los requerimientos de datos, la configuración de las directrices del sistema y la formulación preliminar de la estrategia de administración de datos. Se analizan las fuentes de información, se priorizan las demandas y se traza la hoja de ruta para la gestión eficaz de los datos.

B. Elaboración: Diseño Detallado y Planificación Integral

La fase de Elaboración se concentra en la concepción detallada del sistema, la creación de modelos y la delineación de la arquitectura. En el dominio de la ingeniería de datos, este proceso se traduce en la elaboración de modelos de datos, la concepción de la estructura de almacenamiento y la traza minuciosa de las transformaciones de los datos. También se instituyen pautas para los procesos de prueba y se ajusta la planificación del proyecto en base a los descubrimientos realizados.

C. Construcción: Desarrollo Iterativo y Pruebas Continuas

La fase de Construcción materializa el desarrollo efectivo del sistema y la realización iterativa de pruebas, destinadas a garantizar la calidad del resultado final. Desde la perspectiva de la ingeniería de datos, esto supone la implementación de flujos de Extracción, Transformación y Carga (ETL), la concepción de los elementos de procesamiento de datos y la ejecución de pruebas de integración exhaustivas. Se introducen ajustes de acuerdo con los requerimientos y se generan elementos funcionales tangibles.

D. Transición: Preparación para la Entrega y Operación

La fase de Transición tiene como foco la preparación integral para la entrega del sistema al cliente. Las pruebas finales se llevan a cabo, los problemas residuales se solventan y los usuarios son capacitados. En la ingeniería de datos, esta etapa asegura la integridad y la calidad de los datos, ejecuta

pruebas integrales de todos los componentes y garantiza la disposición para la operación ininterrumpida del sistema.

Las fases del Proceso Unificado de Rational poseen una aplicabilidad inequívoca en el contexto de la ingeniería de datos. Guiando desde la identificación de requerimientos hasta la concreción de arquitecturas de datos y la instauración de soluciones, cada fase proporciona una estructura sólida que asegura un desarrollo eficaz, la mitigación de riesgos y la constante preservación de la calidad en los sistemas de ingeniería de datos.

III. MODELOS DE ROLES Y RESPONSABILIDADES

El Proceso Unificado de Rational (RUP) se caracteriza por un modelo de roles y responsabilidades bien definido que asigna tareas y funciones específicas a los miembros del equipo en cada fase del desarrollo [3][4][6].

A. Roles Clave en RUP

Dentro del contexto del RUP, destacan roles esenciales como el Analista de Requisitos, el Arquitecto de Software, el Diseñador y el Tester. Cada rol desempeña una función crucial en la ejecución efectiva del proyecto.

B. Responsabilidades Asociadas a Cada Rol

Los roles en RUP no solo delinean tareas, sino también responsabilidades concretas. El Analista de Requisitos, por ejemplo, tiene la responsabilidad de capturar y documentar los requisitos del sistema. El Arquitecto de Software, por otro lado, está a cargo del diseño de la estructura general del sistema.

C. Aplicación en Ingeniería de Datos

En el ámbito de la ingeniería de datos, estos roles se adaptan para abordar los desafíos específicos de la gestión de datos. El Rol del Analista de Datos implica la identificación de fuentes y requisitos de datos, mientras que el Arquitecto de Datos se concentra en el diseño de la infraestructura de almacenamiento y ETL.

E. Flexibilidad para Adaptarse a Diferentes Contextos

El enfoque de roles y responsabilidades en RUP es adaptable a diversas áreas, incluyendo la ingeniería de datos. A medida que las tecnologías evolucionan, los roles pueden ajustarse para abordar nuevos desafíos y oportunidades en este campo en constante cambio.

El modelo de roles y responsabilidades en el Proceso Unificado de Rational provee un marco cohesivo que dirige y organiza las acciones de los equipos de desarrollo. En la ingeniería de datos, estos roles garantizan una gestión y transformación de datos eficaz, promoviendo la colaboración y la entrega exitosa de sistemas de alta calidad.

IV. FLUJO DE TRABAJO Y ACTIVIDADES

El Proceso Unificado de Rational (RUP) se distingue por su flujo de trabajo estructurado y sus actividades detalladas, las cuales se interconectan con el objetivo de lograr el desarrollo exitoso de sistemas de ingeniería de datos [5][6].

A. Inicio: Sembrando las Bases

En la fase de Inicio, se identifican los requisitos clave y se traza un plan preliminar. Las actividades incluyen la definición de alcance, la identificación de riesgos iniciales y la estimación de recursos. Esta fase sienta las bases para el resto del proyecto y establece las pautas iniciales para la gestión de datos.

B. Elaboración: Diseño y Planificación Detallados

En la fase de Elaboración, las actividades se centran en el diseño detallado y la planificación exhaustiva. Se crean modelos de datos y arquitecturas técnicas, y se elabora una estrategia integral de pruebas. La interrelación entre estas actividades garantiza la

D. Colaboración y Sinergia de Roles

El modelo de roles en RUP fomenta la colaboración y sinergia entre los miembros del equipo. En la ingeniería de datos, la colaboración entre roles como el Desarrollador ETL y el Administrador de Bases de Datos es crucial para lograr una gestión y transformación de datos eficiente y precisa.

congruencia entre el diseño y las expectativas, estableciendo cimientos sólidos para la etapa de construcción.

C. Construcción: Creación y Verificación de Componentes

Durante la fase de Construcción, las actividades se orientan hacia la implementación real y la verificación constante. Los flujos de Extracción, Transformación y Carga (ETL) se codifican, y los componentes de procesamiento de datos se desarrollan y prueban iterativamente. Esta interacción entre el desarrollo y las pruebas asegura la calidad y la coherencia en la creación de componentes.

D. Transición: Pruebas Finales y Preparación para la Entrega

La fase de Transición involucra actividades finales y la preparación para la entrega. Se realizan pruebas exhaustivas de los sistemas y se corregirán los problemas restantes. Los usuarios son capacitados y se verifica que los datos estén listos para operar en el entorno de producción. La interconexión entre estas actividades garantiza una entrega sin problemas y la transición efectiva del sistema.

La interrelación entre las actividades en cada fase del RUP es crucial para el desarrollo de sistemas de ingeniería de datos exitosos. Cada fase construye sobre las bases establecidas en la anterior, asegurando la coherencia, la calidad y la eficacia en el proceso de desarrollo. El flujo de trabajo y las actividades detalladas en el RUP permiten que los equipos de ingeniería de datos avancen de manera estructurada hacia la creación de soluciones sólidas y bien fundamentadas.

V. CONCLUSIÓN

En resumen, el Proceso Unificado de Rational (RUP) se presenta como un enfoque altamente adaptable y estructurado para el desarrollo de sistemas de ingeniería de datos. Sus fases del ciclo de vida, roles definidos y flujo de trabajo meticulosamente diseñado ofrecen una guía sólida para la creación exitosa de soluciones de datos eficientes y de alta calidad. A medida que el entorno tecnológico evoluciona, el RUP demuestra su relevancia continua al proporcionar un marco robusto que aborda los desafíos cambiantes de la ingeniería de datos, promoviendo la colaboración, la coherencia y la excelencia en el desarrollo de sistemas de datos funcionales y efectivos.

REFERENCIAS

- [1] Ahmad, Gul & Soomro, Tariq & Brohi, M Nawaz. (2014). XSR: Novel Hybrid Software Development Model (Integrating XP, Scrum & RUP). *International Journal of Soft Computing and Engineering (IJSCE)*. 4. 126-130.
- [2] Ahmed, Faheem & Capretz, Luiz. (2008). Best practices of RUP (R) in software product line development. 10.1109/ICCCE.2008.4580828.
- [3] M. Fernandes, J., & Duarte J., F. (2005). *A reference framework for process-oriented software development*. Obtenido de Research Gate: https://www.researchgate.net/figure/Roles-activities-and-artefacts-for-business-modelling-in-RUP_tbl1_220059310
- [4] Zaat, R. (2007). *Use all RUP roles in small teams*. Obtenido de InfoSupport: <https://blogs.infosupport.com/use-all-rup-roles-in-small-teams/>
- [5] Ortega, L. (2022). *Metodología RUP: ¿Qué es, cuál es su objetivo y cómo se utiliza?* Obtenido de Lean Managment Blog: <https://lean-management.site/rup/>
- [6] Villalba Domínguez, E. (2012). *Proceso Unificado Rational Unificado*. Obtenido de Ptolomeo UNAM: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/175/A8%20Cap%C3%ADulo%205.pdf?sequence=8>

Flujo	Roles	Actividades	Artefactos	Ejemplo del Artefacto
Modelo del Negocio	1.- Analista de procesos de negocio. 2.- Diseñador del negocio. 3.- Stakeholders. 4.- Revisor del modelo de negocio.	1.- Comprender la estructura y la dinámica de la organización en la cual se va a implantar el sistema. 2.- Comprender los problemas actuales de la organización e identificar las mejoras potenciales. 3.- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización. 4.- Derivar los requerimientos del sistema que va a soportar la organización.	1.- Modelo de casos de uso del negocio. 2.- Modelo de objetos del negocio. 3.- Especificaciones complementarias del negocio. 4.- Glosario de términos.	Diagrama de Clases del Modelo de Objetos.
Requerimiento	1.- Analista de sistema. 2.- Especificador de casos de uso. 3.- Diseñador de interfaz de usuario. 4.- Arquitecto.	1.- Identificar y clasificar requerimientos. 2.- Encontrar actores y casos de uso. 3.- Priorizar casos de uso. 4.- Detallar casos de uso. 5.- Prototipar la interfaz de usuario. 6.- Estructurar el modelo de casos de uso.	1.- Modelo de casos de uso del sistema. 2.- Actores. 3.- Casos de uso. 4.- Descripción de la arquitectura (vista del modelo de casos de uso). 5.- Glosario de términos. 6.- Prototipo de interfaz de usuario.	Diagrama de Casos de Uso del Atlas de Medicamentos al Sistema

Análisis y Diseño	<p>Análisis:</p> <ol style="list-style-type: none"> 1.- Arquitecto de software. 2.- Ingeniero de casos de uso. 3.- Ingeniero de componentes. <p>Diseño:</p> <ol style="list-style-type: none"> 1.- Arquitecto. 2.- Diseñador. 3.- Diseñador de Bases de Datos. 	<ol style="list-style-type: none"> 1.- Transformar los requisitos en un diseño del sistema en creación. 2.- Evolucionar una arquitectura sólida para el sistema. 3.- Adaptar el diseño para que se ajuste al entorno de implementación, con un diseño pensado para el rendimiento. 	<p>Análisis:</p> <ol style="list-style-type: none"> 1.- Paquete de análisis. 2.- Clase del análisis. 3.- Diagrama de colaboración. 4.- Diagrama de secuencia. 5.- Diagrama de clases del análisis. 6.- Realización de casos de uso. <p>Diseño:</p> <ol style="list-style-type: none"> 1.- Modelo de diseño. 2.- Subsistemas de diseño. 3.- Paquetes de diseño. 4.- Modelo de despliegue. 5.- Descripción de la arquitectura. 6.- Modelo de datos. 	Diseño de la Base de datos que será utilizada por el sistema.
Implementación	<ol style="list-style-type: none"> 1.- Arquitecto de software. 2.- Diseñador. 3.- Diseñador de IU. 4.- Diseñador de cápsulas. 5.- Diseñador de base de datos. 6.- Programador del sistema. 7.- Integrador de los componentes desarrollados. 	<ol style="list-style-type: none"> 1.- Define la organización del código en capas. 2.- Implementa el diseño en términos de componentes de software (código fuente y código objeto, binarios, ejecutables, etc.) 3.- Prueba los componentes de manera individual (pruebas unitarias). 4.- Integra los resultados producidos por colaboradores o equipos en un sistema ejecutable completo. 	<ol style="list-style-type: none"> 1.- Código fuente operativo y documentado. 2.- Diagrama de paquetes. 3.- Modelo de Implementación. 	Archivo Readme con información adicional sobre como implementar de manera correcta el sistema.

Prueba (Testeo)	1.- jefe de pruebas. 2.- Gestor de pruebas. 3.- Especialista en pruebas (tester). 4.- Analista de pruebas. 5.- Diseñador de pruebas.	1.- Verificar la interacción. 2.- Verificar la integración de los componentes. 3.- Verificar que todos los recursos estén correctamente implementados. 4.- Identificar y asegurar que los defectos sean corregidos antes de desplegar el software.	1.- Especificación de casos de prueba.	Resultados de una prueba de estrés cuando se ejecutan varias consultas al mismo tiempo.
Instalación o despliegue	1.- jefe de despliegue. 2.- Jefe de configuración. 3.- Revisor técnico. 4.- Coordinador de revisiones.	1.- Producir releases externos. 2.- Empaquetar, distribuir e instalar el software. 3.- Brindar ayuda y asistencia a los usuarios. 4.- Planear y conducir pruebas beta. 5.- Migración de software existente y datos. 6.- Aceptación formal.	1.- Material de soporte para el usuario final. 2.- Material para entrenamiento. 3.- Software ejecutable. 4.- Productos de instalación. 5.- Notas de la distribución.	Tutorial de cómo realizar una alta de un producto al almacén.
Administración del proyecto	1.- Revisor de gestión del proyecto. 2.- jefe del Proyecto.	1.- Equilibrar los objetivos competitivos. 2.- Administrar el riesgo. 3.- Superar restricciones para entregar un producto que satisface las necesidades de ambos clientes con éxito (los que pagan el dinero) y los usuarios.	1.- Un marco conceptual para gestionar proyectos de software. 2.- Lineamientos prácticos para planear, asignar, ejecutando y monitoreando proyectos. 3.- Un marco conceptual para gestionar el riesgo.	Manual de qué hacer si se presenta algún riesgo en la seguridad del servidor debido a una configuración errónea.
Administración de configuración y cambios	1.- jefe de control de cambios.	1.- Mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener la información del proceso evolutivo que se ha seguido.	1.- Gestión de la configuración. 2.- Gestión de las peticiones de cambio. 3.- Métricas y status.	Documento para manejar de forma efectiva la petición del cliente de cambiar el caso de uso: Ventas al por mayor.

Ambiente	1.- jefe de Proyecto. 2.- jefe de control de cambios. 3.- Stakeholders.	1.- Preparar el entorno para el trabajo. 2.- Preparar el entorno para una iteración. 3.- Preparar las líneas de guía para una iteración. 4.- Dar soporte al entorno durante la iteración.	1.- Caso de desarrollo. 2.- Guías para el modelado del negocio. 3.- Guías para la interfaz de usuario. 4.- Guías para el diseño. 5.- Guías para la programación. 6.- Manual de usuario.	Instrucciones para el correcto uso del sistema.
----------	---	--	--	--