



# Ingeniería del Software

## APLICACIÓN DEL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE A NIVEL LOCAL

### RESUMEN

Este artículo presenta una breve historia del surgimiento de Proceso Unificado de Desarrollo de Software, seguido de las bases sobre las que trabaja. También habla de las ventajas y desventajas que ha presentado su aplicación a nivel local, basándose para esto, en la opinión de estudiantes, pertenecientes a la Universidad Industrial de Santander, han usado esta metodología en el desarrollo de su trabajo de grado.

### PALABRAS CLAVES

Proceso Unificado, desarrollo de software, arquitectura, casos de uso, Universidad industrial de Santander.

### INTRODUCCIÓN

Los estudiantes y profesionales en el ámbito informático, se enfrentan a menudo a la necesidad de desarrollar software que agregue valor a las organizaciones que deseen implementarlo, lo cual, nos exige un conocimiento amplio respecto a los procesos necesarios para desarrollar dichos productos. Para emprender la labor del desarrollo de software cabe citar el Proceso Unificado como una de las metodologías más completas, el Proceso Unificado trae mayores beneficios en cuanto a la reducción de riesgos críticos durante el desarrollo de un proyecto y una estructura muy sólida para el mismo.

### CONTENIDO

#### Breve Historia del Proceso Unificado

El Proceso Unificado tiene varias décadas de desarrollo y uso práctico, desde sus inicios en el proceso *Objectory* (primera publicación en 1987) hasta el *Proceso Unificado de Rational* (Rational Unified Process, o RUP, publicado en 1998). Ver figura 1.

El RUP (Proceso Unificado de Rational) es un proceso de desarrollo de software dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. RUP pretende implementar las mejores prácticas en ingeniería de software, con el objetivo de asegurar la producción de software de calidad, dentro de plazos y presupuestos predecibles.

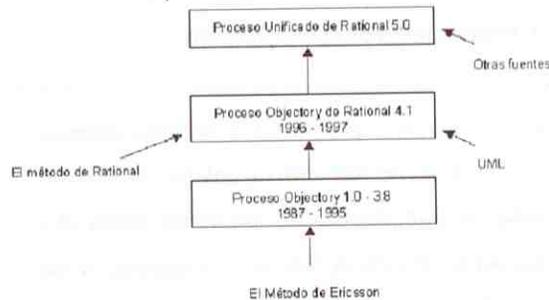


Figura No. 1

Basado inicialmente en las propuestas metodológicas de Ivar Jacobson en Ericsson, quien en 1987 fundó Objectory AB en Estocolmo, y desarrolló un proceso denominado Objectory, extendiendo su uso en otras industrias además de la de las telecomunicaciones, y en otros países aparte

Proceso Unificado



de Suiza, éste producto llegó a ser visto como un sistema. Esta forma de describir un proceso (como un producto en forma de sistema) proporcionaba una mejor manera de desarrollar una nueva versión de un producto específico.

Rational Software Corporation compró Objectory AB a finales de 1995 y la tarea de unificar los principios básicos en los procesos de desarrollo existentes adquirió una urgencia especial, por ello, recibe aportaciones de otros autores (liderados por Philippe Kruchten). Muchos libros, artículos y documentos detallan los procesos desarrollos por Rational desde 1981, pero las contribuciones más importantes fueron los énfasis en la arquitectura y en el desarrollo iterativo. UML estaba en fase de desarrollo y se incorporó como el lenguaje de modelado del proceso Objectory de Rational (Rational Objectory Process, ROP). En junio de 1998 se publica RUP y este Proceso Objectory de Rational se había convertido en un proceso muy sólido, capaz de soportar el ciclo de vida del desarrollo en su totalidad.

### Visión General del Proceso Unificado

El Proceso Unificado proporciona normas para el desarrollo eficiente de software de calidad debido, en gran parte, a que desde las primeras fases del desarrollo se centra en la detección y corrección de riesgos críticos para el proyecto, además porque se basa en tres conceptos fundamentales: está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental.

- **Dirigido por casos de uso**

Al construir un sistema, es muy importante conocer lo

que sus futuros usuarios necesitan, para ello es necesario, por lo tanto, capturar los requisitos del sistema software, y es precisamente aquí donde se entra a adoptar los *casos de uso*.

El hecho de capturar los requisitos lleva implícito dos objetivos: encontrar los verdaderos requisitos y representarlos de un modo adecuado para los usuarios, clientes y desarrolladores; cuando se habla de verdaderos requisitos, se hace referencia a aquellos que cuando se implementan, añaden el valor esperado para los usuarios. Se sabe que normalmente, un sistema tiene muchos usuarios, cada uno de ellos denominado *actor*, cuando éste interactúa con el sistema, recibe un resultado importante, después de que este último realizara una serie de secuencias de acciones; a esta clase de interacción es a lo que se denomina *caso de uso*, por lo tanto, están describiendo la funcionalidad del sistema.

Los casos de uso no solamente se usan para capturar los requisitos del sistema, también dirigen el proceso de desarrollo en su totalidad. El proceso Unificado de Desarrollo de software consiste en una serie de flujos de trabajo que van desde los requisitos hasta las pruebas, y dentro de cada flujo, los casos de uso van a tener un papel indispensable. Los desarrolladores inician capturando los requisitos del cliente en forma de casos de uso, luego analizan y diseñan el sistema para cumplir dichos casos y por último preparan un modelo de prueba que va a permitir comprobar que el sistema proporciona la funcionalidad descrita en los casos de uso.

- **Centrado en la Arquitectura**

La arquitectura de un sistema software es útil para



presentar una clara visión global del mismo. Para ella no son importantes los detalles propios de la labor de cada uno de los desarrolladores del sistema software en cuestión, en vez de esto, se centra en la descripción de los elementos estructurales que compondrán el sistema y sus interfases. Lo anterior da a entender que la arquitectura es el soporte del desarrollo total de un proyecto.

La arquitectura se describe mediante diferentes vistas del sistema en construcción, dichas vistas deben contener los elementos del sistema más importantes para los desarrolladores, tales como nodos e interfaces, entre otros. Además esta debe incluir tanto los aspectos estáticos como dinámicos más significativos del sistema. La arquitectura y los casos de uso tienen una interacción de dependencia mutua, esto es, los casos de uso modelan la arquitectura y esta, a su vez, puede llegar a modificarlos e incluir nuevos casos de uso.

## ● Iterativo e Incremental

Cuando se emprende un proyecto por lo general su resultado final no es del todo claro, por lo que se necesita de una metodología que se ajuste a este tipo de requerimientos, además se desea que dicha metodología sea controlable en el tiempo. Las iteraciones ayudan a controlar el tiempo disponible para la ejecución de un proyecto.

Las iteraciones hacen referencia a diversos pasos en el flujo de trabajo, y los pasos, al incremento del producto. Las iteraciones deben estar controladas, es decir, deben ejecutarse en forma planificada. En cada iteración, se identifican y especifican los casos de uso relevantes, se crea un diseño utilizando la arquitectura seleccionada como guía, se implementa el diseño mediante componentes, y se verifican que

los componentes satisfacen los casos de uso.

Si una iteración cumple sus objetivos el desarrollo continúa con la siguiente iteración, en caso contrario, se revisa las decisiones previas y se prueba un nuevo enfoque. Esto hace que el desarrollo a través de las iteraciones sea incremental.

El Proceso Unificado lleva a cabo el desarrollo de un proyecto en ciclos, donde se obtiene una nueva versión del producto al final de cada ciclo. Cada ciclo se divide a su vez en cuatro fases: Inicio, Elaboración, Construcción, y Transición. Cada una de estas fases concluye con un hito, cada hito es determinado por la disponibilidad de un conjunto de artefactos. Además cada una de las fases se dividen en iteraciones. Cada una de las iteraciones se asemeja a un pequeño ciclo de vida en cascada, pasando a través de los cinco flujos de trabajo fundamentales: Requisitos, Análisis, Diseño, Implementación y Prueba. Cada iteración además incluye una planificación que precede a los flujos de trabajo y una evaluación de ella misma.

## Ventajas y desventajas de su aplicación, un punto de vista local.

En la Escuela de Ingeniería de Sistemas e Informática (EISI), adscrita a la Universidad industrial de Santander, se ha venido planteando, especialmente en el último año, la necesidad de aplicar nuevas metodologías para el desarrollo de software, con el fin de prepara mejores profesionales. Para apoyar esta idea se ha incluido el Proceso Unificado como tema central de la materia *Diseño de Análisis y Documentación*, lo que asegura que todos los estudiantes que cursen esta materia a partir del

Arquitectura



segundo período académico de 2001 tengan una idea general sobre la aplicación de esta metodología.

Paralelo a lo anterior, se ha iniciado el desarrollo en la EISI de una serie de proyectos de grado utilizando esta metodología de desarrollo de software como base. Estos desarrollos permiten la aplicación del Proceso Unificado en un ámbito fuera del estrictamente académico del aula de clase.

Con base en testimonios de estudiantes, pertenecientes a la EISI, que actualmente se encuentran desarrollando su trabajo de grado usando el Proceso Unificado como metodología de desarrollo de software, se ha encontrado una serie de ventajas y desventajas al momento de aplicar dicha metodología.

Las ventajas más significativas son en general: la posibilidad de hacer un plan de trabajo bastante real, aunque el tema del proyecto no sea de total dominio por parte de los autores del mismo; la visión general que se consigue del proyecto a realizar desde fases tempranas de desarrollo; la rápida detección de errores y dificultades, y su relativamente fácil corrección; la posibilidad de desarrollar en paralelo por parte de varios desarrolladores; la misma organización y estructura del Proceso Unificado; entre otros.

Las desventajas expresadas son: el difícil acceso a bibliografía sobre el Proceso Unificado, ya que el libro *El Proceso Unificado de Desarrollo de Software* es prácticamente el único sobre el tema al que se tiene acceso; además, este libro es, en algunos apartes, de difícil interpretación; el hecho de que resulta tediosa la elaboración de los diferentes diagramas en UML, indispensables para la ejecución de un proyecto desarrollado con la metodología en cuestión; la falta

de "resultados rápidos" en fases tempranas del desarrollo, esto es, el hecho de que durante la fase de inicio y gran parte de la fase de elaboración no se implementa software.

Se puede observar fácilmente que las desventajas son relativamente de menor importancia que las ventajas presentadas, por lo que se puede concluir que la ejecución de un proyecto usando el Proceso Unificado como metodología de desarrollo de software, es, por demás, ventajosa.

#### AUTORES

Claudia López. e-mail: [claudia\\_evelin@hotmail.com](mailto:claudia_evelin@hotmail.com)

Feisar Moreno. e-mail: [rasief@hotmail.com](mailto:rasief@hotmail.com)

Elkin Méndez. e-mail: [angelm22@hotmail.com](mailto:angelm22@hotmail.com)

Oscar Miranda. e-mail: [oscaramiranda@hotmail.com](mailto:oscaramiranda@hotmail.com)

Federico Leal. e-mail: [nekkosoft@yahoo.com.mx](mailto:nekkosoft@yahoo.com.mx)

