

## **Proceso Unificado de Desarrollo (UP del Inglés Unified Process)**

El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el Proceso Unificado de Rational, también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del RUP. Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto.

El nombre **Proceso Unificado** se usa para describir el proceso genérico que incluye aquellos elementos que son comunes a la mayoría de los refinamientos existentes. También permite evitar problemas legales ya que Proceso Unificado de Rational o RUP son marcas registradas por IBM (desde su compra de Rational Software Corporation en 2003). El primer libro sobre el tema se denominó, en su versión española, El Proceso Unificado de Desarrollo de Software (ISBN 84-7829-036-2) y fue publicado en 1999 por Ivar Jacobson, Grady Booch y James Rumbaugh, conocidos también por ser los desarrolladores del UML, el Lenguaje Unificado de Modelado. Desde entonces los autores que publican libros sobre el tema y que no están afiliados a Rational utilizan el término Proceso Unificado, mientras que los autores que pertenecen a Rational favorecen el nombre de Proceso Unificado de Rational.

## **Proceso unificado a través del lenguaje Unificado de Modelado (UML)**

**UML** (Unified Modeling Language), es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado), pero no especifica en sí mismo qué metodología o proceso usar.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como

lo es la orientación a objetos, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas, algunos de estos diagramas son los siguientes:

- **Diagrama de Clases:** es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientados a objetos.
- **Diagrama de Componentes:** representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.
- **Diagrama de Casos de Uso:** es una forma de diagrama de comportamiento UML mejorado. El Lenguaje de Modelado Unificado (UML), define una notación gráfica para representar casos de uso llamada modelo de casos de uso.
- **Diagrama de secuencia:** es un tipo de diagrama usado para modelar interacción entre objetos en un sistema.
- **Un diagrama de colaboración:** en las versiones de UML 1.x es esencialmente un diagrama que muestra interacciones organizadas alrededor de los roles.
- **El Diagrama de Despliegue** es un tipo de diagrama que se utiliza para modelar la disposición física de los artefactos software en nodos (usualmente plataforma de hardware).

## **El proceso Unificado de Desarrollo de Software o simplemente Proceso Unificado**

Es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP.

### **Características**

- **Iterativo e Incremental:** El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio puede incluir varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación y Prueba. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.

- **Dirigido por los casos de uso:** En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc. El proceso dirigido por casos de uso es el rup. Nota: en UP se está Dirigido por requisitos y riesgos: de acuerdo con el Libro UML 2 de ARLOW, Jim que menciona el tema.
- **Centrado en la arquitectura:** El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. La analogía con la construcción es clara, cuando construyes un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc.
- **Enfocado en los riesgos:** El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

## ¿Por qué analizar y diseñar?

En resumidas cuentas, la cuestión fundamental del desarrollo del software es la escritura del código. Después de todo, los diagramas son solo imágenes bonitas. Ningun usuario va a agradecer la belleza de los dibujos; lo que el usuario quiere es software que funcione (UML Gota a Gota, Addison Wesley, Pag 7). Por lo tanto, cuando considere usar el UML es importante preguntarse por qué lo hará y como le ayudara a usted cuando llegue el momento de escribir el código. No existe una evidencia empirica adecuada que demuestre si estas tecnicas son buenas o malas; Pero lo que si es cierto es que es de considerable ayuda para las etapas de mantenimiento en proyectos de mediana/avanzada envergadura.

La metodología de UP, es un método iterativo de diseño de software que describe cómo desarrollar software de forma eficaz, utilizando técnicas probadas en la industria.

El Proceso Unificado de Desarrollo de Software o simplemente Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura, enfocado en el riesgo, y por ser iterativo e incremental.

El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos.

El nombre Proceso Unificado se usa para describir el proceso genérico que incluye aquellos elementos que son comunes a la mayoría de los refinamientos existentes. Es una metodología orientada a conducir el proceso de desarrollo de software en sus aspectos técnicos; los flujos y productos de trabajo de UP no incluyen la administración del proyecto.

## **Fases de Desarrollo**

### **Fase de Inicio.**

- Es la fase más pequeña del proyecto e, idealmente, debe realizarse también en un periodo de tiempo pequeño (una única iteración).
- El hecho de llevar a cabo una fase de inicio muy larga indica que se está realizando una especificación previa excesiva, lo que responde más a un modelo en cascada.
- Objetivos:
  1. Establecer una justificación para el proyecto.
  2. Establecer el ámbito del proyecto.
  3. Esbozar los casos de uso y los requisitos clave que dirigirán las decisiones de diseño.
  4. Esbozar las arquitecturas candidatas.
  5. Identificar riesgos.
  6. Preparar el plan del proyecto y la estimación de costes.
- El hito de final de fase se conoce como Hito Objetivo del Ciclo de Vida.

### **Fase de Elaboración.**

- Durante esta fase se capturan la mayoría de los requisitos del sistema.

- Los objetivos principales de esta fase serán la identificación de riesgos y establecer y validar la arquitectura del sistema.
- Base de Arquitectura Ejecutable:
  - La arquitectura se valida a través de la implementación de una Base de Arquitectura Ejecutable: se trata de una implementación parcial del sistema que incluye los componentes principales del mismo.
  - Al final de la fase de elaboración la base de arquitectura ejecutable debe demostrar que soporta los aspectos clave de la funcionalidad del sistema y que muestra la conducta adecuada en términos de rendimiento, escalabilidad y coste.
- Al final de la fase se elabora un plan para la fase de construcción.
- El hito arquitectura del ciclo de vida marca el final de la fase.

#### **Fase de construcción.**

- Es la fase más larga de proyecto.
- El sistema es construido en base a lo especificado en la fase de elaboración.
- Las características del sistema se implementan en una serie de iteraciones cortas y limitadas en el tiempo.
- El resultado de cada iteración es una versión ejecutable de software.
- El hito de capacidad operativa inicial marca el final de la fase.

#### **Fase de transición.**

- En esta fase el sistema es desplegado para los usuarios finales.
- La retroalimentación recibida permite incorporar refinamientos al sistema en las sucesivas iteraciones.
- Esta iteración también cubre el entrenamiento de los usuarios para la utilización del sistema.
- El hito de lanzamiento del producto marca el final de la fase.

## **Disciplinas**

### **Modelado del negocio.**

- El objetivo es establecer un canal de comunicación entre los ingenieros del negocio y los ingenieros del software.
- Los ingenieros del software deben conocer la estructura y dinámica de la organización objetivo (el cliente), los problemas actuales y sus posibles mejoras.
- Se plasma en la identificación del modelo del dominio en el que se visualizan los aspectos básicos del dominio de aplicación.

### **Requisitos.**

- El objetivo es describir que es lo que tiene que hacer el sistema y poner a los desarrolladores y al cliente de acuerdo en esta descripción.

### **Análisis y diseño.**

- Describe como el software será realizado en la fase de implementación.
- Se plasma en un modelo de diseño que consiste en una serie de clases (agrupadas en paquetes y subsistemas) con interfaces bien definidos.
- También contiene descripciones de cómo los objetos colaboran para realizar las acciones incluidas en los casos de uso.

### **Implementación.**

- Se implementan las clases y objetos en términos de componentes (ficheros fuentes, binarios, ejecutables, entre otros).

### **Prueba.**

- Se comprueba que el funcionamiento es correcto analizando diversos aspectos: los objetos como unidades, la integración entre objetos, la implementación de todos los requisitos, entre otros.

### **Despliegue.**

- Se crea la versión externa del producto, se empaqueta, se distribuye y se instala en el lugar de trabajo. También se da asistencia y ayuda a los usuarios.

### **Gestión de configuraciones y cambios.**

- Gestiona aspectos como los sistemas de control de versiones.
- Controla las peticiones de cambios clasificándolas según su estado (nueva, registrada, aprobada, asignada, completa, entre otros).

- Los datos se almacenan en una base de datos y se pueden obtener informes periódicos.
- Herramientas como Rational ClearQuest o Bugzilla.

### Gestión del proyecto.

- Encargada de definir los planes del proyecto global, los planes de fase y los planes de iteración.

### Entorno.

- Se centra en las actividades necesarias para configurar el proceso de un proyecto.
- El objetivo es proveer a la organización de desarrollo software de un entorno de trabajo (que incluye procedimientos y herramientas) que soporten al equipo de desarrollo.

El ciclo de vida del Proceso Unificado está sintetizado en la siguiente ilustración.

