



# **EL PROCESO UNIFICADO DEL DESARROLLO DE SOFTWARE**

Lectura Técnica

Universidad Tecnológica del Centro de Veracruz

TICSI 6° A DZ

Ingeniería de sistemas I

**Por: Contreras Ortiz José Manuel**

6886

# ÍNDICE

ÍNDICE.....	1
El Proceso unificado .....	3
El proceso unificado en pocas palabras .....	3
El Proceso Unificado está dirigido por casos de uso .....	3
El Proceso Unificado está centrado en la arquitectura.....	3
El Proceso Unificado es iterativo e incremental .....	4
La vida del Proceso Unificado .....	4
El producto .....	4
Fases dentro de un ciclo.....	5
El proceso integrado .....	5
Las cuatro “P ” en el desarrollo de software.....	6
Las personas son decisivas .....	6
Los procesos de desarrollo afectan a las personas .....	6
Los papeles cambiarán .....	6
Convirtiendo “recursos” en “trabajadores” .....	6
Los proyectos construyen el producto .....	7
El producto es más que código .....	7
¿Qué es un sistema de software? .....	7
Artefactos.....	7
Un sistema contiene una colección de modelos.....	7
¿Qué es un modelo? .....	7
Cada modelo es una vista autoncontenida del sistema.....	8
Dentro de un modelo .....	8
Relaciones entre modelos.....	8
El proceso dirige los proyectos.....	8
En proceso: una plantilla .....	8
Las actividades relacionadas conformar flujos de trabajo .....	8
Procesos especializados .....	9
Méritos de proceso .....	9
Las herramientas son esenciales en el proceso .....	9
Las herramientas influyen en el proceso .....	9
El proceso dirige las herramientas .....	9

El equilibrio entre el proceso y las herramientas.....	9
El modelado visual soporta UML.....	10
Las herramientas dan soporte al ciclo completo .....	10
Bibliografía .....	11

# **El Proceso unificado**

## **El proceso unificado en pocas palabras**

Los sistemas informáticos actuales son cada vez más grandes y complejos debido a las mejoras en el hardware de los equipos de cómputo, esto aunado a la creciente expansión y utilización de internet ha provocado que sea necesario ofrecer productos de software en un tiempo relativamente menor, esto ocasiona a su vez que los desarrolladores necesiten coordinarse cada vez mejor especialmente en proyectos de gran envergadura.

El Proceso Unificado es un proceso de desarrollo de software el cual se considera como un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, este proceso utiliza el Lenguaje Unificado de Modelado para preparar todos los esquemas de un sistema de software.

## **El Proceso Unificado está dirigido por casos de uso**

Un sistema de software se desarrolla para ofrecer un servicio a sus usuarios, cabe mencionar que el término usuarios hace referencia también a otros sistemas o cualquier elemento que interactúe con el sistema.

Este tipo de interacciones es considerada un caso de uso y representa una funcionalidad del sistema, por tanto son considerados requisitos funcionales y todos juntos constituyen el modelo de casos de uso, los casos de uso se especifica y diseñan, los casos de uso finales son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba.

## **El Proceso Unificado está centrado en la arquitectura**

La arquitectura en un sistema de software se describe mediante diferentes vistas del sistema en construcción, incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa y se reflejan en los casos de uso, pero también se ve influenciada por factores como la plataforma donde va a funcionar el software, los bloques de construcción reutilizables de los que se dispone, consideraciones de implantación, sistemas heredados y requisitos no funcionales.

Los caso de uso deben encajar con la arquitectura cuando se llevan a cabo y debe permitir el desarrollo de todos los casos de uso requeridos en su momento y a futuro, deben evolucionar de manera paralela.

El arquitecto debe:

- Crear un esquema en borrador de la arquitectura.
- Trabajar con un subconjunto de caso de usos especificados que representen funciones clave.
- A medida que los casos de uso madura se descubre más de la arquitectura.

## **El Proceso Unificado es iterativo e incremental**

Es práctico dividir el trabajo en partes más pequeñas o miniproyectos, cada miniproyecto es una iteración que resulta en un incremento, deben planificarse y ejecutarse de manera planificada.

Los desarrolladores basan la selección en dos factores:

- La iteración trata un grupo de casos de uso juntos que amplían la utilidad de los productos desarrollados hasta ahora.
- La iteración trata los riesgos más importantes.

En cada iteración los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño mediante componentes y verifican que los componentes satisfacen los casos de uso, si la iteración cumple con los objetivos se continúa con la siguiente iteración.

Entre los beneficios de un proceso iterativo están:

- Reduce el coste de riesgos a coste de un solo incremento.
- Reduce el riesgo de no sacar al mercado el producto en el calendario previsto.
- Acelera el ritmo del esfuerzo de desarrollo en su totalidad.
- Reconoce una realidad que a menudo se ignora.

## **La vida del Proceso Unificado**

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen un sistema, cada ciclo consta de cuatro fases (inicio, elaboración, construcción y transición) y cada fase se subdivide en iteraciones.

## **El producto**

Cada ciclo produce una nueva versión del sistema y cada versión es un producto preparado para su entrega. Este consta de un cuerpo de código fuente incluido en los componentes que puede compilarse y ejecutarse, además de los manuales y otros productos asociados.

El producto terminado incluye los requisitos, casos de uso, especificaciones no funcionales y casos de prueba, incluye el modelo de arquitectura y el modelo visual, al final los desarrolladores deben afrontar un nuevo ciclo y los directores deben financiarlos.

Para llevar a cabo el siguiente ciclo los desarrolladores necesitan todas las representaciones del producto de software.

- Un modelo de casos de uso.
- Un modelo de análisis.
- Un modelo de diseño.
- Un modelo de implementación.
- Un modelo de despliegue.
- Un modelo de pruebas.
- Una representación de la arquitectura.

## **Fases dentro de un ciclo**

Cada ciclo se desarrolla a través del tiempo, este se divide en cuatro fases y dentro de cada fase los directores o desarrolladores pueden descomponer adicionalmente su trabajo, cada fase termina con un hito.

Los hitos tienen diversos objetivos siendo el más crítico que los directores deben tomar decisiones cruciales antes de que el trabajo pueda continuar a la siguiente fase, los hitos también permiten a la dirección y a los mismos desarrolladores controlar el progreso del trabajo según se pasa por esos puntos clave.

Durante la fase de inicio se desarrolla una descripción final del producto a partir de una buena idea y se presenta el análisis de negocio para el producto.

Durante la fase de elaboración se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema, se planifican las actividades y se estiman los recursos necesarios para terminar el proyecto.

## **El proceso integrado**

Durante la fase de construcción se crea el producto, esta es la línea base donde la arquitectura crece hasta convertirse en el sistema completo evolucionando hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios.

La fase de transición cubre el periodo durante el cual el producto se convierte en una versión beta, los desarrolladores corrigen los problemas e incorporan algunas de las mejoras sugeridas en una versión general dirigida a la totalidad de la comunidad de usuarios.

## **Las cuatro “P ” en el desarrollo de software**

El resultado final de un proyecto de software es un producto que toma forma durante su desarrollo gracias a la intervención de distintos tipos de personas.

- Personas: Son los seres humanos autores de un proyecto de software.
- Proyecto: Elemento organizativo a través del cual se gestiona el desarrollo de software.
- Producto: Artefactos que se crean durante la vida de un proyecto.
- Herramientas: Software que se utiliza para agilizar las actividades definidas.

## **Las personas son decisivas**

Hay personas implicadas en el desarrollo de un producto de software durante todo su ciclo de vida.

## **Los procesos de desarrollo afectan a las personas**

El modo en que se organiza y gestiona un proyecto de software afecta a las personas implicadas en el.

- Viabilidad de proyecto: Definir las posibilidades de éxito del proyecto.
- Gestión de riesgo: Deducir los posibles riesgos que se puedan presentar.
- Estructura de equipo: División de tareas, organización del equipo de trabajo.
- Planificación del proyecto: Establecer objetivos, alcances y limitaciones.
- Factibilidad de comprensión del proyecto: El equipo de trabajo debe tener claro en que se están trabajando.
- Sensación de cumplimiento: Retroalimentación y conclusiones obtenidas.

## **Los papeles cambiarán**

Debido a que los cambios del mundo real seguirán sucediendo durante todos estos ciclos de vida, los sistemas tendrán que diseñarse de un modo que les permita crecer durante largos periodos de tiempo.

Para comprender y dar soporte a esos procesos de negocio cada vez más complejos y para implementarlos en software, los desarrolladores deberán trabajar con muchos otros desarrolladores.

## **Convirtiendo “recursos” en “trabajadores”**

Un trabajador es un papel que un individuo puede desempeñar en el desarrollo de software, como puede ser un especificador de casos de uso, un arquitecto, un ingeniero de componentes o un ingeniero de pruebas, un trabajador puede asumir roles con otros trabajadores en diferentes flujos de trabajo.

Cada trabajador es responsable de un conjunto de actividades, necesitan la información requerida para desempeñar sus actividades, necesitan comprender su rol con respecto a otros

trabajadores, un trabajador también puede definir a un conjunto de personas que trabajan juntas.

## **Los proyectos construyen el producto**

A través de su ciclo de vida, un equipo de proyecto debe preocuparse del cambio, las iteraciones y el patrón organizativo dentro del cual se conduce el proyecto:

- Una secuencia de cambio: Los proyectos de desarrollo de sistemas obtienen productos como resultados.
- Una serie de iteraciones: Dentro de cada fase de cada ciclo los trabajadores llevan a cabo las actividades de la fase a través de una serie de iteraciones.
- Un patrón organizativo: El proyecto implica a un equipo de personas asignadas para lograr un resultado dentro de las restricciones del negocio.

## **El producto es más que código**

El término producto hace referencia al sistema entero y no solo al código que se entrega.

### **¿Qué es un sistema de software?**

Un sistema es todos los artefactos que se necesitan para presentarlo en una forma comprensible por las máquinas y los trabajadores y los interesados. Las máquinas son las herramientas, compiladores u ordenadores destinatarios.

### **Artefactos**

Artefacto es un término general para cualquier tipo de información creada, producida, cambiada o utilizada por los trabajadores en el desarrollo del sistema.

## **Un sistema contiene una colección de modelos**

Cuando diseñamos el Proceso Unificado identificamos a todos los trabajadores y cada una de las perspectivas que podríamos necesitar, las perspectivas recogidas de todos los trabajadores se encuentran en unidades más grandes, es decir, modelos, de modo que un trabajador puede tomar una perspectiva concreta del conjunto de modelos.

La construcción de un sistema es por tanto un proceso de construcción de modelos, utilizando distintos modelos para describir todas las perspectivas diferentes del sistema.

### **¿Qué es un modelo?**

Un modelo es una abstracción del sistema, especificando el sistema modelado desde un cierto punto de vista y en un determinado nivel de abstracción



## **Cada modelo es una vista autoncontentida del sistema**

Un modelo es una abstracción semántica cerrada del sistema, es una vista autocontentida en el sentido de que un usuario de un modelo no necesita para interpretarlo más información.

La idea de ser autocontentido significa que los desarrolladores trataron de que hubiera una sola interpretación de lo que ocurrirá en el sistema cuando se dispare un evento descrito en el modelo. Además del sistema el modelo debe describir las interacciones entre el sistema que se está modelando, el modelo también debería incluir elementos que describan partes relevantes de su entorno, es decir, actores.

## **Dentro de un modelo**

Un modelo siempre identifica el sistema que se está modelando, este elemento del sistema es por tanto el contenedor de los demás elementos.

## **Relaciones entre modelos**

Un sistema contiene todas las relaciones y restricciones entre elementos incluidos en diferentes modelos. Por tanto un sistema no es solo la colección de sus modelos sino que contiene todas las relaciones entre ellos.

El hecho de que los elementos en dos modelos estén conectados no cambia lo que hacen en los modelos a los que pertenecen.

## **El proceso dirige los proyectos**

En el contexto del Proceso Unificado la palabra proceso se refiere a los procesos “de negocio” claves en una empresa de desarrollo de software, en estos negocios también existen otros procesos como el proceso de soporte o el de venta.

## **En proceso: una plantilla**

En el Proceso Unificado, proceso hace referencia a un contexto que sirve como plantilla que puede reutilizarse para crear instancias de ella, este proceso no cubre solamente el primer ciclo de desarrollo sino también los posteriores más comunes.

## **Las actividades relacionadas conformar flujos de trabajo**

Describimos el proceso entero en partes llamadas flujos de trabajo, un flujo de trabajo es un estereotipo de colaboración en el cual los trabajadores y los artefactos son participantes. Por tanto los trabajadores y artefactos que participan en un flujo de trabajo pueden participar también en otros flujos de trabajo.

## **Procesos especializados**

Un Proceso Unificado puede especializarse para cumplir diferentes necesidades de aplicación o de organización, al mismo tiempo, es deseable que el proceso se, al menos completamente consistente dentro de una organización.

Los factores principales que influyen en cómo se diferenciará el proceso son:

- Factores organizativos.
- Factores de dominio.
- Factores de ciclo de vida.
- Factores técnicos.

## **Méritos de proceso**

Un proceso común dentro y a lo largo de los equipos de desarrollo proporciona muchos beneficios:

- Todo el equipo comprende sus tareas a lo largo del desarrollo.
- Los desarrolladores comprenden el trabajo de otros.
- Los supervisores o directores comprenden lo que se está haciendo.
- Los desarrolladores, supervisores y directores pueden cambiar fácilmente de proyecto o división.
- La formación puede estandarizarse dentro de una empresa.
- El devenir del desarrollo de software es repetible.

## **Las herramientas son esenciales en el proceso**

### **Las herramientas influyen en el proceso**

Las herramientas son buenas para automatizar procesos repetitivos, mantener las cosas estructuradas, gestionar grandes cantidades de información y para guiarnos a lo largo de un camino de desarrollo en concreto, con menos soporte por herramientas un proceso tiende a ser más manual y menos formal.

### **El proceso dirige las herramientas**

La capacidad de automatizar un proceso depende de que tengamos una visión clara de que casos de uso necesita el usuario y que artefactos necesita manejar, las herramientas que implementan un proceso automatizado deben ser fáciles de usar. Hacerlas muy manejable significa que los desarrolladores de herramientas deben considerar seriamente la forma en que se lleva a cabo el desarrollo de software.

### **El equilibrio entre el proceso y las herramientas**

Por un lado, el proceso dirige el desarrollo de las herramientas. Por otro lado, las herramientas dirigen el desarrollo del proceso. El desarrollo del proceso y su soporte por herramientas debe tener lugar de manera simultánea. En cada versión del proceso debe hacer también una versión de las herramientas.

El desarrollo con éxito de una automatización de un proceso (herramientas) no puede hacerse sin el desarrollo paralelo de un marco de trabajo de proceso en el cual vayan a funcionar las herramientas.

## **El modelado visual soporta UML**

UML define reglas sintácticas que especifican como combinar elementos del lenguaje. Por tanto, la herramienta debe ser capaz de garantizar que se cumplen esas reglas, sin embargo el cumplir todas esas reglas sin excepción haría que la herramienta fuera imposible de utilizar.

UML incluye un cierto número de reglas semánticas que también requieren soporte. Estas herramientas pueden incluirse en la herramienta de modelado, bien como de cumplimiento obligado inmediato o como rutina bajo demanda que recorre el modelo y comprueban si hay errores comunes o buscan faltas de compleción semántica o sintáctica.

## **Las herramientas dan soporte al ciclo completo**

Hay herramientas que soportan todos los aspectos del ciclo de vida:

- Gestión de requisitos: Se utiliza para almacenar, examinar, revisar, hacer seguimiento y navegar por los diferentes requisitos de un proyecto de software.
- Modelado visual: Se utiliza para automatizar el uso de UML.
- Herramientas de programación: Proporciona una gama de herramientas.
- Aseguramiento de la calidad: Se utiliza para comprobar aplicaciones y componentes.

Además de estas herramientas hay otras que abarcan todo el ciclo de vida. Estas herramientas incluyen control de versiones, gestión de configuración, seguimiento de defectos, documentación, gestión del proyecto y automatización de procesos.

## **Bibliografía**

Jacobson I., Booch G., Rumbaugh J. (2000). *El proceso unificado de desarrollo de software*. Madrid. Addison Wesley.