



**Universidad Tecnológica del Centro de Veracruz**

**Programa educativo de:**

***TÉCNICO SUPERIOR UNIVERSITARIO EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN.***

**Materia:**

***“INGENIERA DE SOFTWARE”***

**Área:**

***“Sistemas Computacionales”***

**Presenta:**

**Luis Ángel Fernández Vásquez Luis Ángel.**

**Grado y grupo: 5A TICS I DZ**

**Periodo cuatrimestral: Mayo-Agosto 2015.**

**Cuitláhuac, Veracruz, 28 de Mayo del 2015**



## Contenido

.....	1
Capítulo 1 .....	3
El Proceso Unificado: dirigido por casos de uso, centrado en la arquitectura .....	3
EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE .....	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 .....	4
EL PROCESO UNIFICADO ES ITERATIVO E INCREMENTAL.....	4
LA VIDA DEL PROCESO UNIFICADO .....	5
EL PRODUCTO.....	5
EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE .....	6
FASES DENTRO DE UN CICLO.....	6
UN PROCESO INTEGRADO .....	6
Capítulo 2 .....	7
EL PRODUCTO ES MÁS QUE CÓDIGO.....	8
EL PROCESO DIRIGE LOS PROYECTOS.....	8
Bibliografía .....	9

## Capítulo 1

### El Proceso Unificado: dirigido por casos de uso, centrado en la arquitectura

Bueno la tendencia del software nos lleva a la construcción de sistemas más grandes y más complejos. Esto debido a que a que las computadoras de la actualidad han ido evolucionando cada año que transcurre son más sofisticadas y por ello los usuarios esperan más de ellos.

En la actualidad se sigue produciendo software de alta demanda pero lo la mayoría de la gente lo hace con métodos que están desde hace 25 años esto puede ser un problema al menos que sé que se renueven los métodos, no se podrán cumplir con el objetivo de desarrollar software más completo que se necesita en la actualidad.

En la actualidad los desarrolladores afrontan un gran problema que es coordinar múltiples tareas dentro de un proyecto, esto se puede resolver con un método que acepte múltiples facetas de desarrollo.

### EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

\*Proporcione una guía para ordenar las actividades de un equipo

\*Dirigirá las tareas de cada desarrollador por separado y del equipo como todo.

\*Ofrezca criterios para el control y la medición de los productos y actividades del proyecto.

Un método bien definido y bien gestionado crea la diferencia entre proyectos hiperproductivos y otros que fracasan.

### EL PROCESO UNIFICADO EN POCAS PALABRAS

El proceso unificado el proceso de desarrollo de software es el conjunto de actividades necesarias para la formar los requisitos de un usuario en un sistema de software.

Un proceso está basado en componentes, lo cual quiere decir que el sistema de software está conformado por componentes de software.

### EL PROCESO UNIFICADO ESTÁ DIRIGIDO POR CASOS DE USO

En el término usuario no solo nos referimos a un ser humano ni no a otros sistemas que puedan interactuar con el mismo sistema.

Los casos de uso representan los requisitos funcionales. Todos los casos de uso constituyen el modelo de casos de uso la cual describe toda la funcionalidad de todo el sistema.

Los casos de uso no guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso.

## EL PROCESO UNIFICADO ESTÁ CENTRADO EN LA ARQUITECTURA

La arquitectura es un sistema software se describe mediante diferentes visitas del sistema en construcción.

La arquitectura siempre surge de las necesidades del cliente y estos se reflejan en los casos de uso a la vez esta es influida por varios factores, como la plataforma en la que se tiene que funcionar todo el software, los bloques son reutilizados de que se dispone como por ejemplo un marco de trabajo. La arquitectura le da vista del diseño con las características más importantes resaltadas. Estos dependen de una valoración que se adquiere con la experiencia, la arquitectura se depende de las personas que se hayan responsabilizado de su creación.<sup>4</sup>

¿Cómo se relacionan los casos de uso y la arquitectura?

Cada producto tiene como tanto una función como una forma, ya que ninguna es suficiente por sí misma, esta dos se tienen que equilibrar para obtener todo un producto con éxito, estas formas hacen a la arquitectura.

Por lo tanto los desarrolladores tienen que modelar todo el sistema para darle forma al sistema, a esta forma se le considera arquitectura la que permite que el sistema evolucione y no solo su desarrollo inicial sino que también a lo largo de las futuras generaciones. Para esta comprensión los desarrolladores tienen que se necesita conocer las funciones claves.

## EL PROCESO UNIFICADO ES ITERATIVO E INCREMENTAL

En el desarrollo de un software comercial supone un gran esfuerzo que puede durar entre varios meses hasta que posiblemente un año o más. En este caso es práctico dividir el trabajo en partes más pequeñas para mini-proyectos.

Para lograr alcanzar un mayor grado de economía en el desarrollo, un equipo de proyecto intentará seleccionar solo las iteraciones requeridas para lograr el objetivo del proyecto, intentará secuenciar solo pequeñas desviaciones del curso de los desarrolladores planifican inicialmente.

Son muchos beneficios de un proceso iterativo controlado:

- \*La iteración controlada reduce el coste del riesgo a los costes de un solo incremento.

- \*La iteración controlada reduce el riesgo de no sacar al mercado un producto en el calendario previsto. Mediante la identificación de riesgos en fases tempranas del desarrollo, el tiempo que se gasta en resolverlos se emplea al principio de la planificación.

- \*La iteración controlada acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados claros a corto plazo, en lugar de tener un calendario largo, que se prolonga eternamente.

- \* La iteración controlada reconoce una realidad que a menudo se ignora que las necesidades del usuario y sus correspondientes requisitos no pueden definirse completamente al principio.

## LA VIDA DEL PROCESO UNIFICADO

Se repite a lo largo de una serie de ciclos que comprenden la vida de un sistema. Cada ciclo consta de cuatro fases, desde su inicio hasta su muerte: inicio, elaboración, construcción y transición. Cada ciclo produce una nueva versión del sistema y cada versión es un producto preparado para su entrega. El producto terminado incluye los requisitos, casos de uso, especificaciones no funcionales, y casos de prueba. Estos elementos son importantes para clientes, usuarios, analistas, diseñadores, programadores, ingenieros de prueba; estos permiten hacer cambios a través del tiempo.

Los desarrolladores deben afrontar los nuevos ciclos y los clientes deben financiarlos.

Cada ciclo se divide en cuatro fases, dentro de cada una de ellas los diferentes desarrolladores termina con un hito. El hito se determina cuando se alcanza un estado predefinido. En los hitos los directores deben tomar decisiones con respecto a la continuación de la siguiente fase. Estos permiten controlar el progreso del trabajo.

\*

\* En la fase de inicio se desarrolla una descripción del producto final y se presenta el análisis del negocio.

\* En la fase de elaboración se especifican en detalle la mayoría de los casos de uso y se diseñan la arquitectura del sistema. El resultado es una línea base de la arquitectura.

\* Durante la fase de construcción la línea base de la arquitectura crece hasta convertirse en el sistema completo.

\* EN la fase de transición el producto se convierte en versión beta. Los desarrolladores corrigen problemas e incorporan mejoras.

## EL PRODUCTO

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

El producto ya terminado tiene que incluir los requisitos de caso de uso y sus especificaciones no funcionales y casos de prueba incluye un modelo de la arquitectura y el modelo visual artefactos como he mencionado con el lenguaje unificado de modelado.

No importa que los componente ejecutables sean los artefactos más importantes desde la perspectiva del usuario, no son insuficientes por si solos, esto se debe a que el entorno cambia y se pueden mejorar los sistemas operativos, base de datos, y las computadoras que soportar el software.

## EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

\* Un modelo de casos de uso, con todos los casos de uso y su relación con los usuarios.

\*Un modelo de análisis, con dos propósitos; refinar los casos de uso con más detalle y establecer la asignación inicial de profundidad del sistema a un conjunto que proporcionan el componente.

\*Un modelo de diseño que define la estructura estática del sistema en forma de subsistemas, clases e interfaces,

\*Un modelo de implementación, que se incluye componentes que representan al código fuente y las clases que corresponden.

\*Un modelo de despliegue que redefine los nodos físicos y la correspondencia de los componentes con esos nodos.

\*Un modelo de prueba, que describe los casos de uso de prueba que verifican los casos de uso.

## FASES DENTRO DE UN CICLO

Cada ciclo se desarrolla a lo largo del tiempo. Este tiempo, a su vez se divide en cuatro fases. Dentro de cada fase los directores o los desarrolladores pueden descomponer adicionalmente el trabajo en iteraciones con sus incrementos resultantes. Cada fase termina con un hito. Cada hito determina por la disponibilidad el trabajo en conjunto de artefactos es decir ciertos modelos o documentos han sido disponibles hasta alcanzar un estado predefinido.

Durante la fase de construcción se crea el producto se añade toda la arquitectura del software. En esta fase, la línea base de la arquitectura cree hasta convertirse en el sistema completo. La descripción evoluciona hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios. El grueso de los recursos requeridos se emplea durante esta fase del desarrollo.

## UN PROCESO INTEGRADO

El proceso unificado está basado en componentes, Utiliza el nuevo estándar de modelado visual el lenguaje unificado de modelado y se sostiene sobre tres ideas funcionales, se necesita un proceso polifacético, que tenga en cuenta los ciclos, fases y flujos de trabajo, gestión en riesgo, control de calidad, gestión del proyecto.

## Capítulo 2

Las cuatro P en el desarrollo de software: Personas, Proyecto, Producto y Proceso

- \* Personas. Son seres humanos, son los principales autores del proyecto de software.
- \* Proyecto. Elemento organizativo a través del cual se desarrolla el proyecto.
- \* Producto. Modelos, código fuente, ejecutables, documentación.
- \* Proceso. Conjunto completo de actividades para transformar los requisitos del usuario en un producto. Es una plantilla para crear proyectos.
- \* Herramientas: Software para automatizar las actividades del proceso.

### LAS PERSONAS SON DECISIVAS

Durante todo el ciclo de vida de un producto software hay personas implicadas. Es muy importante lo siguiente con respecto al proyecto:

- \* Viabilidad del proyecto. Los proyectos no viables pueden detenerse en sus fases iniciales.
- \* Gestión del riesgo. Explorar los riesgos disminuye problemas.
- \* Estructura de equipos. Grupos de 6 a 8 personas, una buena arquitectura lo hace posible.
- \* Planificación del proyecto. Definición de tiempo para cumplir los objetivos.
- \* Facilidad de comprensión del proyecto. La descripción de la arquitectura proporciona una visión global.
- \* Sensación de cumplimiento. La retroalimentación y las conclusiones aceleran el ritmo de trabajo.

Para que las personas sean eficaces, el proceso debe estar sobre un UML. Permite especificar los requisitos de acuerdo a la necesidad. Un buen proceso ayuda a construir sistemas más complejos. Se elige una arquitectura para construir sistemas de forma económica y puntual. Un trabajador es una persona designada para realizar algo y ellas lo aceptan. Cada uno de ellos tiene responsabilidades y lleva a cabo un conjunto de actividades en el desarrollo del software. Pueden ser grupos de trabajo dentro de un proyecto. Los proyectos construyen el producto. Se da como resultado de una nueva versión del producto. Se desarrolla y se obtiene el producto inicial para representar la gestión del proyecto. Se debe preocupar por lo siguiente:

- \* Una secuencia de cambio. Se obtienen productos con una serie de cambios hasta obtener el proyecto final. Cada ciclo, fase e iteración modifican el sistema de ser una cosa a otra distinta.

- \* Una serie de iteraciones. Se lleva a cabo actividades de la fase a través de una serie de iteraciones. En cada iteración se pasa por los flujos de trabajo (requisitos, diseño, implementación y pruebas) como si fuese un mini proyecto.
- \* Un patrón organizativo. Se proporciona un patrón o plantilla dentro del cual las personas en su diferente papel, ejecutan el proyecto. Indica a que personas necesita y los artefactos por los cuales hay que trabajar. Ofrece muchas guías, heurísticas y prácticas de documentación que ayudan a realizar el trabajo asignado.

## EL PRODUCTO ES MÁS QUE CÓDIGO

El producto que se obtiene es un sistema software. Un sistema es todos los artefactos que se necesitan para representarlo en una forma comprensible por máquinas u hombres, para las máquinas, los trabajadores y los interesados. Entre las máquinas están herramientas, compiladores, ordenadores destinatarios. Los trabajadores son arquitectos, desarrolladores, ingenieros de prueba, personal de marketing, administradores y otros. Los interesados son inversionistas, usuarios, comerciales, jefes de proyecto, directores de línea de producto, personal de producción, otros.

Artefactos son los diagramas UML y su texto asociado, los bocetos de interfaz de usuario, los prototipos, componentes, planes de prueba y los procedimientos de prueba.

Hay dos tipos de artefactos: de ingeniería y de gestión. Un artefacto utilizado en proceso unificado es el modelo.

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

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

La mayoría de modelos de ingeniería se define mediante un subconjunto de UML cuidadosamente seleccionado.

## EL PROCESO DIRIGE LOS PROYECTOS

Se refiere al proceso de negocio claves en una empresa desarrolladora de software. En el Proceso Unificado, hace referencia a un contexto que sirve de plantilla que puede reutilizarse para crear instancias de ella. Un proceso no solo cubre el primer ciclo de desarrollo sino también los ciclos posteriores. Es una definición de un conjunto de actividades.

El proceso Unificado puede especializarse para cumplir diferentes necesidades de aplicación o de organización.

Los factores principales de cómo se diferenciara el proceso son:

- \* Factores organizativos. Estructura organizativa, la cultura, la organización, la gestión del proyecto en una empresa.
- \* Factores del dominio. Procesos de negocio que se deben soportar, comunidad de usuarios y las ofertas disponibles de la competencia.



\* Factores del ciclo de vida. Tiempo de salida al mercado, tiempo de vida esperado del software, las personas involucradas en el desarrollo del software y las versiones planificadas y futuras.

\* Factores técnicos. Lenguaje de programación, herramientas de desarrollo, bases de datos.

Las herramientas soportan los procesos de desarrollo de software modernos. Es indispensable desarrollar software sin utilizar un proceso basado en herramientas. Las herramientas son esenciales en el proceso.

(IVAR JACABSON, 1998)

## Bibliografía

IVAR JACABSON, G. B. (1998). *El Proceso Unificado de Desarrollo de Software*. addison Wesley.