



CARRERA:

Técnico Superior Universitario en Tecnologías de la Información y Comunicación.

AREA:

Área Sistemas Informáticos

MATERIA:

Ingeniería del Software I

PROFESOR (a):

M.C.C. Aguilar Galicia Honorato

NOMBRE DEL ALUMNO:

Espinoza Gómez Héctor René

MATRICULA:

7641

Periodo cuatrimestral: Mayo-Agosto 2015.



Contenido

| | |
|---|---|
| 1. PROCESO UNIFICADO: DIRIGIDO POR CASOS DE USO, CENTRADO EN LA ARQUITECTURA, INTERACTIVO E INCREMENTAL | 4 |
| 1.1 CASOS DE USO | 4 |
| 1.2 CENTRADO EN LA ARQUITECTURA..... | 4 |
| 1.3 INTERACTIVO E INCRMENTAL | 4 |
| 1.4 LA VIDA DEL PROCESO UNIFICADO | 5 |
| 1.4.1 PRODUCTO | 5 |
| 1.4.2 FASES DENTRO DE UN CICLO..... | 5 |
| 2. LAS CUATRO “P” EN EL DESARROLLO DE SOFTWARE: PERSONAS, PROYECTO, PRODUCTO Y PROCESO | 6 |
| 2.1 LAS PERSONAS SON DECISIVAS | 6 |
| 2.1.1 LOS PROCESOS DE DESARROLLO AFECTAN A LAS PERSONAS..... | 6 |
| 2.1.2 LOS PAPELES CAMBIARAN..... | 6 |
| 2.1.3 CONVIRTIENDO “RECURSOS” EN “TRABAJADORES” | 7 |
| 2.2 LOS PROYECTOS CONSTRUYEN PRODUCTOS..... | 7 |
| 2.3 EL PRODUCTO ES MAS QUE CODIGO | 7 |
| 2.3.1 ¿QUE ES UN SISTEMA DE SOFTWARE?..... | 7 |
| 2.3.2 ARTEFACTOS..... | 7 |
| 2.3.3 UN SISTEMA POSEE UNA COLECCIÓN DE MODELOS | 7 |
| 2.3.4 ¿QUE ES UN MODELO?..... | 8 |
| 2.3.5 CADA MODELO ES UNA VISTA AUTOCONTENIDA DEL SISTEMA..... | 8 |
| 2.3.6 RELACION ENTRE MODELOS | 8 |
| 2.4 EL PROCESO DIRIGE LOS PROYECTOS..... | 8 |
| 2.4.1 EL PROCESO: UNA PLANTILLA | 8 |
| 2.4.2 LAS ACTIVIDADES RELACIONADAS CONFORMAN FLUJOS DE TRABAJO | 8 |
| 2.4.3 PROCESOS ESPECIALIZADOS..... | 8 |
| 2.4.4 MERITOS DEL PROCESO..... | 9 |
| 2.5 LAS HERRAMIENTAS SON ESENCIALES EN EL PROCESO..... | 9 |
| 2.5.1 LAS HERRAMIENTAS INFLUYEN EN EL PROCESO..... | 9 |

2.5.2 EL PROCESO DIRIGE LAS HERRAMIENTAS 9

2.5.3 EL EQUILIBRIO ENTRE PROCESO Y LAS HERRAMIENTAS 9

2.5.4 LAS HERRAMIENTAS DAN SOPORTE AL CICLO DE VIDA COMPLETO..... 9

1. PROCESO UNIFICADO: DIRIGIDO POR CASOS DE USO, CENTRADO EN LA ARQUITECTURA, INTERACTIVO E INCREMENTAL

Con el paso de los años, la necesidad de crear software mejor desarrollados para cumplir nuestras necesidades, ya que el usuario requiere computadoras más potentes y rápidas, hoy en día la mayoría de los desarrolladores ocupa los mismo métodos que se utilizan hace varios años, y esto genera un problema, el desarrollador necesita un método que le ayude a dirigir y ordenar las actividades de un equipo, esto haría la diferencia entre un proyecto que logre el éxito o uno que fracase. La solución al problema del software es el Proceso Unificado de Desarrollo que utiliza el Lenguaje Unificado de Modelado (UML), ya que hace un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, el proceso unificado ocupa tres frases para poder ejecutarlo de la mejor manera: a) Casos de uso b) Centrado en la arquitectura c) Iterativo e incremental

1.1 CASOS DE USO

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona la usuario una resultado importante, al referirnos como usuario no solo son los usuarios humanos también pueden ser otros sistemas un ejemplo muy conocido seria cuando vamos a un cajero interactuamos con el sistema y esperamos que nos dé una respuesta en este caso sería una suma de dinero. Los casos de uso representan los requisitos funcionales, los cuales describen la funcionalidad total del sistema, podremos realizar los casos de uso ocupando la pregunta ¿Qué debe de hacer el sistema..... después se le añade a la preguntapara cada usuario?. También guían el diseño, implementación y prueba, entendemos que los casos de uso no solo inician el proceso de desarrollo si no y también crean un hilo de que avanza a través de los flujos del trabajo, o sea, guían la arquitectura del sistema y la arquitectura influye en la selección de los casos de uso.

1.2 CENTRADO EN LA ARQUITECTURA

El concepto de arquitectura de software incluye los aspectos estáticos y dinámicos más significativos del sistema surge de las necesidades de la empresa y se refleja en los casos de uso, otros factores que la componen pueden ser: el sistema operativo, sistema de gestión de base de datos también entran los requisitos no funcionales. La arquitectura y los casos de uso deben de ir de la mano ya que la función le corresponde a los casos de uso y la forma a la arquitectura, en pocas palabras deben de evolucionar en paralelo. El arquitecto del sistema debe poseer una comprensión general de los casos de uso antes de comenzar la creación del esquema arquitectónico, a medida que los casos de uso maduran, se descubre más de la arquitectura, esto a su vez, lleva a la maduración de más casos de uso y este proceso continua hasta que se considere que la arquitectura es estable.

1.3 INTERACTIVO E INCRMENTAL

Un proyecto de software puede desarrollarse durante muchos meses, un año o más, haciéndolo pesado y tedioso y no se pueda lograr el objetivo deseado. Por esto mismo es práctico dividir el proyecto en mini proyecto, cada mini proyecto sería una iteración que irá incrementando. Cada iteración trata un grupo de casos de usos primero y después continúan con las siguientes fases del proyecto. Para alcanzar el mayor grado de economía, un equipo de proyecto intentara solo seleccionar las iteraciones necesarias para lograr el objetivo del proyecto, una iteración debe estar controlada y traerá muchas ventajas:

- Acelera el ritmo del esfuerzo del desarrollo en su totalidad ya que los desarrolladores trabajan de manera más eficiente para obtener resultados claros y a corto plazo.
- Reduce el riesgo de no sacar al mercado el producto en el calendario previsto, mediante la identificación de riesgos en fases tempranas del desarrollo.
- Si los desarrolladores tienen que repetir la iteración, la organización solo pierde el esfuerzo mal empleado en la iteración y no el de todo el producto.

1.4 LA VIDA DEL PROCESO UNIFICADO

El proceso unificado se repite a lo largo de una serie de ciclos y cada ciclo concluye con una versión del producto. Los ciclos constan de cuatro fases: inicio, elaboración, construcción y transición, a su vez cada fase se divide en iteraciones.

1.4.1 PRODUCTO

El producto se genera cuando un ciclo genera una versión, pocas palabras, es un entregable el cual tiene un cuerpo: código fuente, manuales, casos de usos, especificaciones no funcionales, casos de prueba, el modelo de prueba y visual, es más son esos elementos los que permiten a los usuarios utilizar y modificar el sistema de generación en generación.

1.4.2 FASES DENTRO DE UN CICLO

Cada ciclo se divide en 4 fases (inicio, elaboración, construcción y transición), cada fase termina con un HITO, que es cuando ciertos modelos o documentos han sido desarrollados hasta alcanzar un estado predefinido.

Fase de inicio.- se desarrolla una descripción del producto final a partir de una buena idea, esta fase responde a las siguientes preguntas: ¿Cuáles son las principales funciones del sistema para los usuarios? Esta se responde con los casos de uso y se va desarrollando el producto; ¿Cómo podría ser la arquitectura del sistema? Y ¿Cuál es el plan del proyecto y cuanto costara desarrollar el producto?

Fase de elaboración.- durante esta fase, se realizan los casos de uso más críticos, el resultado de esta fase es un línea base de la arquitectura. Al final de la fase el director planifica las actividades y estima los recursos necesarios para terminar el proyecto.

Fase de construcción.- se crea el producto, la arquitectura crece hasta convertirse en el sistema completo, al final de esta fase, el producto contiene todos los casos de uso que se han acordado para esta versión, aunque no está completamente libre de defectos, estos ya se verán en la siguiente fase (transición).

Fase de transición.- el producto se convierte en una versión beta, donde ciertos usuarios prueban el producto e informan sus defectos y los desarrolladores se encargaran de corregir ese tipo de problemas y por qué no añadirle incluir unas mejoras sugeridas.

2. LAS CUATRO “P” EN EL DESARROLLO DE SOFTWARE: PERSONAS, PROYECTO, PRODUCTO Y PROCESO

Un proceso de desarrollo de software guía los esfuerzos de las personas implicadas en el proyecto.

Personas.- son realmente seres humanos, arquitectos, desarrolladores, ingenieros de prueba usuarios, clientes y otros interesados.

Proyecto.- es por el cual se gestiona el desarrollo del software su resultado es una versión de producto.

Productos.- se generan durante el proyecto (código fuente, ejecutables y documentación).

Proceso.- conjunto de actividades para transformar los requisitos de usuario en un producto.

Herramientas.- software para utilizar las actividades del proceso.

2.1 LAS PERSONAS SON DECISIVAS

Las persona implicadas en el producto, lo planifican, lo financian, lo desarrollan, etc. por tanto el proceso debe orientarse a las personas, más bien debe funcionar bien para las persona que lo utilizan.

2.1.1 LOS PROCESOS DE DESARROLLO AFECTAN A LAS PERSONAS

Viabilidad del proyecto.- muchos no disfrutan trabajar en proyectos que parecen imposibles, estos proyectos pueden detenerse en una fase temprana.

Gestión del riesgo.- es cuando lo riesgos no son analizados correctamente haciendo incomodo el proyecto.

Estructura de los equipos.- una buena arquitectura, grupos pequeños de 7 o 8 integrantes, hace posible una división del esfuerzo de este tipo.

Planificación del proyecto.- esto es, en las fases de inicio y elaboración, se puede dar una idea de cómo quedara el proyecto y el tiempo que se llevara realizarlo.

Facilidad de comprensión del proyecto.- la descripción de la arquitectura proporción a una visión general para cualquiera que se encuentre en el proyecto.

Sensación de cumplimiento.- es la retroalimentación y las conclusiones.

2.1.2 LOS PAPELES CAMBIARAN

Mediante un soporte de herramientas y un Lenguaje Unificado de Modelado, hacer más eficaces a las personas que se encuentran desarrollando el software, permitir especificar los requisitos que mejor se ajusten a las necesidades de los usuarios, que les permita elegir una arquitectura que permita construir el sistema de forma económica y puntual. Para comprender y dar soporte a esos procesos de negocio más complejos y para implementarlos en software, los desarrolladores deberán trabajar con muchos más desarrolladores y para esto se necesita un proceso que sirva como guía.

2.1.3 CONVIRTIENDO “RECURSOS” EN “TRABAJADORES”

Un trabajador es aquel que puede desempeñar un papel 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 de integración. Cada trabajador es responsable de un conjunto complejo de actividades, para trabajar eficazmente, necesitan la información requerida para llevar a cabo sus actividades, al mismo tiempo, las herramientas que emplean deben de ser las adecuadas. Al asignar los trabajadores a un proyecto, el jefe debe identificar las aptitudes de las personas, no es fácil la primera vez que se emplea el Proceso Unificado. Al repartir los recursos, el jefe debe minimizar el trasiego de artefacto de un recurso a otro de manera que el flujo del proceso sea continuo como se pueda.

2.2 LOS PROYECTOS CONSTRUYEN PRODUCTOS

El primer proyecto dentro del ciclo de vida desarrolla y obtiene el sistema o producto inicial, a través de su ciclo de vida, un equipo de proyecto debe preocuparse del cambio, de las iteraciones y el patrón organizativo dentro del cual se conduce el proyecto:

- a) Una secuencia de cambio: cada ciclo, cada fase y cada iteración modifica el sistema de ser una cosa a otra distinta. El primer ciclo de desarrollo es un caso especial que convierte el sistema de nada a algo.
- b) Una serie de iteraciones: en una iteración los desarrolladores progresan a través de una serie de flujos de trabajo: requisitos, diseño, implementación y prueba.
- c) Un patrón organizativo: la idea de “proceso” es proporcionar un patrón dentro del cual las personas en su papel de trabajadores ejecutan un proyecto, este patrón indica los tipos de trabajadores que el proyecto necesita.

2.3 EL PRODUCTO ES MAS QUE CODIGO

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

2.3.1 ¿QUE ES UN SISTEMA DE SOFTWARE?

Son todos los artefactos que se necesitan para representar el sistema en una forma comprensible por maquinas u hombres, para las maquinas, los trabajadores y los interesados. En términos de subsistemas, clases, diagramas de interacción y diagramas de estado.

2.3.2 ARTEFACTOS

Son cualquier información creada, producida o cambiada. Un artefacto puede ser por ejemplo un diagrama UML, los bocetos de la interfaz y sus prototipos. Hay dos tipos de artefactos: artefactos de ingeniería se crea sobre las distintas fases (requisitos, análisis, diseño, implementación y prueba); artefactos de gestión, solo duran lo que dure la vida del proyecto, como puede ser el análisis de negociación, el plan de desarrollo, etc. Estos artefactos se definen mediante texto o diagramas, utilizando cualquier visualización.

2.3.3 UN SISTEMA POSEE UNA COLECCIÓN DE MODELOS

El tipo de artefacto más interesante utilizado en Proceso Unificado es el modelo. Las perspectivas recogidas de todos los trabajadores se estructuran en unidades más grandes, es decir, en modelos, la construcción de un sistema es por tanto un proceso de construcción de modelos. Este conjunto de modelos hace claro el sistema para todos los trabajadores, incluyendo a los clientes, usuarios y jefes de proyectos.

2.3.4 ¿QUE ES UN MODELO?

Los modelos son abstracciones del sistema que construyen los arquitectos y desarrolladores, no se preocupan de como es el sistema por dentro, solo se preocupan de lo que puede hacer para sus usuarios

2.3.5 CADA MODELO ES UNA VISTA AUTOCONTENIDA DEL SISTEMA

La idea de ser auto contenido 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. Un modelo debe describir las interacción entre el sistema y los que lo rodean, por lo tanto el modelo también debería incluir elementos que describan partes relevantes de su entorno

2.3.6 RELACION ENTRE MODELOS

Ya que un sistema contiene todas las relaciones, un sistema no es solo la colección de sus modelos sino que contiene también las relaciones entre ellos, a esto en UML se le llama dependencia de traza.

2.4 EL PROCESO DIRIGE LOS PROYECTOS

En el contexto del Proceso Unificado, el término proceso se refiere a los procesos de negocio claves en una empresa de desarrollo de software.

2.4.1 EL 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. Un proceso cubre no solamente el primer ciclo de desarrollo, sino también los ciclos posteriores más comunes. En versiones posteriores, una instancia del proceso toma cambios incrementales en los requisitos y produce cambios incrementales en el conjunto de artefactos.

2.4.2 LAS ACTIVIDADES RELACIONADAS CONFORMAN FLUJOS DE TRABAJO

Un flujo de trabajo es un conjunto de actividades, la forma adecuada de obtener los flujos de trabajo, es primero, identificando los distintos tipos de trabajadores, después identificar los artefactos que se necesitan y una vez identificados podemos describir como fluye el proceso a través de los diferentes trabajadores y la forma conveniente de ocupar los artefactos, a todo esto se le llama flujo de trabajo.

2.4.3 PROCESOS ESPECIALIZADOS

Un proceso de desarrollo de software del mundo real debe de ser adaptable y configurable para cumplir con las necesidades reales de un proyecto concreto. El Proceso Unificado puede especializarse para cumplir diferentes necesidades de aplicación o de organización. Los factores principales que influyen en cómo se diferenciara el proceso son:

- Factores organizativos: la estructura de la empresa.
- Factores del dominio: el dominio de la aplicación.
- Factores del ciclo de vida: el tiempo de salida al mercado.
- Factores técnicos: lenguaje de programación.

2.4.4 MERITOS DEL PROCESO

- Todos en el equipo comprenden lo que tienen que hacer para desarrollar el producto.
- Los desarrolladores pueden comprender mejor lo que otros están haciendo.
- Los supervisores y directores comprenden lo que hacen los desarrolladores.
- Se pueden cambiar de proyecto sin aprender un nuevo proceso.
- La formación puede estandarizarse.

2.5 LAS HERRAMIENTAS SON ESENCIALES EN EL PROCESO

Las herramientas soportan los procesos de desarrollo de software modernos, las herramientas son esenciales en el proceso.

2.5.1 LAS HERRAMIENTAS INFLUYEN EN EL PROCESO

Las herramientas se desarrollan para automatizar actividades, de manera completa o parcial, para incrementar la productividad y la calidad, para reducir el tiempo de desarrollo. Se puede utilizar un lenguaje de modelado formal para asegurar que cada modelo es consistente internamente y en relación con otros modelos.

2.5.2 EL PROCESO DIRIGE LAS HERRAMIENTAS

Las herramientas que implementan un proceso automatizado deben de ser fáciles de usar, hacerlas manejables significa que los desarrolladores de herramientas deben considerar seriamente la forma en que se lleva a cabo el desarrollo de software. Las herramientas deben de ser capaces de soportar la automatización de actividades repetitivas como de gestionar la formación representada por la serie de modelos y artefactos.

2.5.3 EL EQUILIBRIO ENTRE PROCESO Y LAS HERRAMIENTAS

El desarrollo del proceso y su soporte por herramientas debe tener lugar de manera simultánea, al alcanzar este equilibrio nos llevara a varias iteraciones. En pocas palabras si el proceso aún no está bien desarrollado, en consecuencia, las herramientas no funcionarían como se espera en el proceso

2.5.4 LAS HERRAMIENTAS DAN SOPORTE AL CICLO DE VIDA COMPLETO

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

- Gestión de requisitos.- se utiliza para almacenar, examinar, revisar, hacer el seguimiento y navegar por los diferentes requisitos del software.
- Modelo visual.- se consigue la integración con entornos de programación y se asegura que el modelo y la implementación siempre sean constantes.
- Herramientas de programación.- incluyen editores, compiladores, depuradores detectores de errores y analizadores de rendimiento.
- Aseguramiento de la calidad.- se utiliza para probar aplicaciones y componentes.

(Ivar Jacobson, Grady Booch, James Rumbaugh, 1998)

Bibliografía

Ivar Jacobson, Grady Booch, James Rumbaugh. (1998). *El Proceso Unificado de Desarrollo de Software*. Addison Wesley.