



Universidad Tecnológica del Centro de Veracruz

Tecnologías de la información y la comunicación

Lectura Técnica:

Capítulo 1: El Proceso Unificado: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

Capítulo 2: Las cuatro “P” en el desarrollo de software: Personas, Proyecto, Producto y Proceso.

Presenta:

González Blanco María de los Ángeles

Matricula

7619

Cuitláhuac, Ver a 27 de mayo del 2015

El Proceso Unificado: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

El Proceso Unificado está centrado por casos de uso

La tendencia actual en el software lleva a la construcción de sistemas más grandes y más complejos. Esto debido en parte al hecho de que los computadores son más potentes cada año y los usuarios, por tanto esperamos más de ellos. Siendo así queremos un software que este adaptado a nuestras necesidades. El proceso unificado es un proceso de desarrollo de software que nos permite realizar actividades mediante una construcción formada por componentes de software en interconectados a través de interfaces bien definidas.

Los aspectos definidos del proceso unificado se resumen en tres frases, la primera es dirigida por casos de usos, segunda centrado en la arquitectura, tercera iterativo e incremental.

Un sistema de software se crea para servir a sus usuarios, el termino usuario no solo hace referencia a usuarios humanos sino a otros sistemas (el término usuario representa alguien o algo como otro sistema fuera del sistema en consideración que interactúa con el sistema que estamos desarrollando). Por lo tanto, para construir un sistema exitoso se debe conocer qué es lo que quieren y necesitan los usuarios prospectos.

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los requisitos funcionales representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso. Este modelo reemplaza la tradicional especificación funcional del sistema. Una especificación funcional tradicional se concentra en responder la pregunta: ¿Qué se supone que el sistema debe hacer? La estrategia de casos de uso puede ser definida agregando tres palabras al final de la pregunta: ¿por cada usuario? Estas tres palabras tienen una implicación importante, nos fuerzan a pensar en términos del valor a los usuarios y no solamente en términos de las funciones que sería bueno que tuviera. Sin embargo, los casos de uso no son solamente una herramienta para especificar los requerimientos del sistema, también dirigen su diseño, implementación y pruebas, esto es, dirigen el proceso de desarrollo.

Aún y cuando los casos de uso dirigen el proceso, no son elegidos de manera aislada. Son desarrollados a la par con la arquitectura del sistema, esto es, los casos de uso dirigen la arquitectura del sistema y la arquitectura del sistema influencia la elección de los casos de uso. Por lo tanto, la arquitectura del sistema y los casos de uso maduran conforme avanza el ciclo de vida.

El Proceso Unificado está centrado en la Arquitectura

El papel de la arquitectura de software es parecido al papel que juega la arquitectura en la construcción de edificios, La arquitectura es una vista del diseño completo con las características más importantes resaltantes dejando los detalles de lado.

La arquitectura de un sistema es la visión general común en la que todos los empleados (desarrolladores y otros usuarios) deben estar de acuerdo. La arquitectura da una clara perspectiva del sistema completo, necesaria para controlar el desarrollo.

Se necesita la arquitectura para:

- Comprender el sistema
- Organizar el desarrollo
- Fomentar la reutilización
- Hacer evolucionar el sistema

El Proceso Unificado es Iterativo e Incremental

Un proceso de desarrollo de software debe tener una secuencia de hitos claramente articulados para ser eficaz, que proporcionen a los directores y al resto del equipo del proyecto los criterios que necesitan para autorizar el paso de una fase a la siguiente dentro del ciclo del producto.

Dentro de cada fase, el proceso pasa por una serie de iteraciones e incrementos que nos conducen a estos criterios

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos el desarrollo continua con la siguiente iteración.

Para un buen desarrollo, el equipo de proyecto deberá intentar seleccionar solo las iteraciones requeridas para lograr el objetivo del proyecto.

Sus objetivos de la reducción del riesgo es minimizar los problemas inesperados.

Los beneficios son muchos, reducen el costo de riesgo a los costes de un solo incremento, acelera el ritmo de esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados, reconoce una realidad que a menudo se ignora.

Como se indicó, dos de las tres claves del proceso unificado son que el proceso de desarrollo de software debe estar dirigido por los casos de uso y centrado en la arquitectura. Estos aspectos tienen un impacto evidente sobre el producto del proceso. El estar dirigido por los casos de uso significa que cada fase en el camino al producto final está relacionada con lo que los usuarios hacen realmente. Lleva a los desarrolladores a garantizar que el sistema se ajusta a las necesidades reales del usuario. El estar centrado en la arquitectura significa que el trabajo de desarrollo se centra en obtener el patrón de la arquitectura que dirigirá la construcción del sistema en las primeras fases, garantizando un proceso continuo no sólo para la versión en curso del producto, sino para la vida entera del mismo.

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

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

- **Personas:** Los principales autores de un proyecto Software son los arquitectos, desarrolladores, ingenieros de prueba y el personal de gestión.
- **Proyecto:** Elemento organizativo a través del cual se gestiona el desarrollo del software.
- **Producto:** Artefactos que se crean durante la vida del proyecto.
- **Proceso:** Un proceso de ingeniería de software es una definición del conjunto completo de actividades necesarias para transformar los requisitos de usuario en un producto.
- **Herramientas:** Software que se utiliza para automatizar las actividades definidas en el proceso.

El equipo de software debe ser uno solo, es decir, funcionar como conjunto, apoyarse mutuamente con el fin de lograr el cumplimiento de los objetivos planteados. Todos los miembros del equipo deben tenerse confianza y distribuir la carga de trabajo según el problema que se esté tratando. No todo equipo es eficiente, pero se puede lograr esto con la suficiente motivación y el apoyo de un buen gestor de proyectos.

Muchas veces cuando un cliente pide que le construyan una solución, siempre pregunta ¿cuánto me va a costar? Pues bien, todo producto requiere estimaciones cuantitativas y una adecuada planificación. Una adecuada recolección de información y un análisis detallado de los requerimientos proporciona la información necesaria para dar una estimación del costo del producto. Antes de planear un proyecto, se debe establecer los objetivos y el alcance que tendrá el proyecto, además de sus restricciones técnicas y de gestión. Con una buena planificación se puede estimar el tiempo que tomará desarrollar o construir el producto y redimensionar el valor cuantitativo del producto.

El proceso del software proporciona un marco de trabajo desde el cual se puede establecer un plan detallado para la construcción del software. Todas las actividades del marco de trabajo se las pueden aplicar a la mayoría de proyectos de software, sino es a todos. El equipo de desarrollo debe elegir el proceso adecuado y que le permita obtener una solución o producto que satisfaga las necesidades o requerimientos del cliente.

Cuando se gestiona un proyecto exitoso, es necesario entender que este puede llegar a fracasar. Según John Reel, existen 10 razones por las cuales un proyecto puede fracasar:

1. El personal de software no entiende las necesidades de los clientes.
2. El ámbito del producto está mal definido.

3. Los cambios se gestionan mal.
4. La tecnología elegida cambia.
5. Las necesidades comerciales cambian.
6. Los plazos de entrega no son realistas.
7. Los usuarios se resisten a la utilización del software.
8. Se pierde el patrocinio.
9. El equipo del proyecto carece de personal con las habilidades apropiadas.
10. Los gestores evitan las mejores prácticas y las lecciones aprendidas.

Para tener éxito en la consecución de un proyecto es necesario comenzar con pie derecho, esto se lo logra trabajando duro para entender el problema y dar una solución adecuada. Se debe rastrear el proyecto conforme se elabora el producto y se aprueba por parte del grupo de control de calidad. Es importante que el gestor del proyecto tome decisiones inteligentes para no poner en riesgo el desarrollo de la solución. Por último, se debe analizar los resultados obtenidos para obtener la experiencia necesaria en la construcción de otros proyectos.