
ESCOM-IPN

Reingeniería y Reutilización Investigación

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

Laura Andrea Morales López

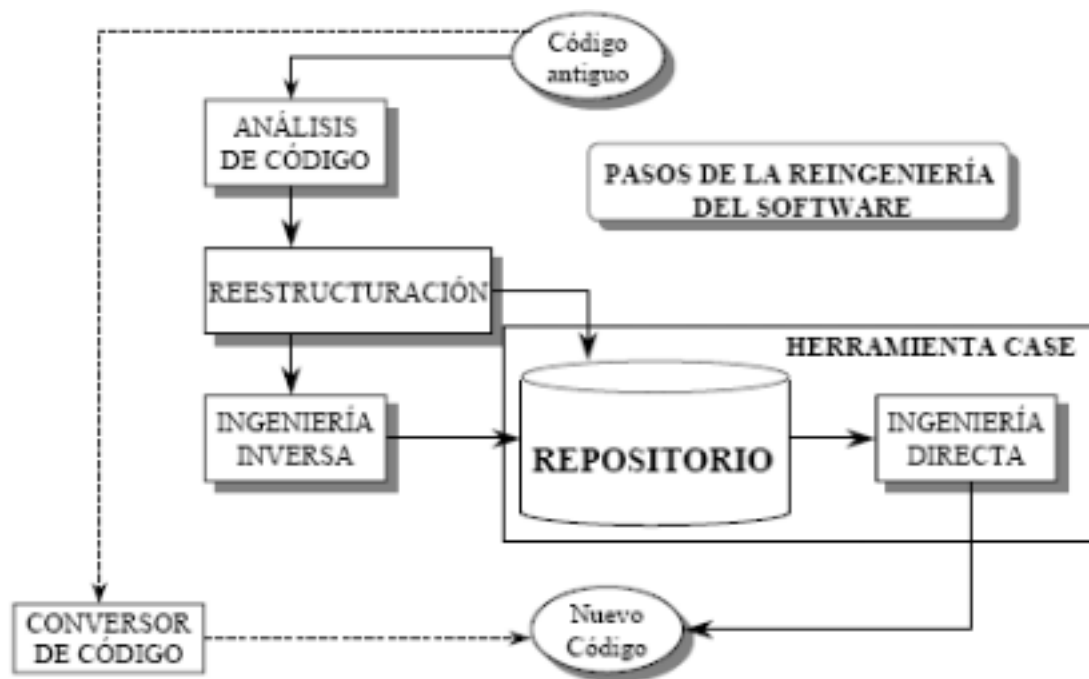
Mayo 2018

1. Reingeniería

1.1. Definición

Modificación de un producto software, o de ciertos componentes, usando para el análisis del sistema existente técnicas de ingeniería inversa y, para etapa de reconstrucción, herramientas de ingeniería directa, de tal manera que se oriente este cambio hacia mayores niveles de facilidad en cuanto a mantenimiento, reutilización, compresión o evaluación.

1.2. Actividades



1.2.1. Análisis de inventario

Los candidatos a la reingeniería aparecen cuando se ordena esta información en función de su importancia para el negocio, longevidad, mantenibilidad actual y otros criterios localmente importantes. Es entonces cuando es posible asignar recursos a las aplicaciones candidatas para el trabajo de reingeniería.

1.2.2. Reestructuración de documentos

Crear documentación consume mucho tiempo, si el sistema funciona vivirá con lo que tenga. La documentación debe actualizarse pero se tiene recursos limitados. Se utiliza un enfoque de “documentar cuando se toque”. El sistema es crucial para el negocio y debe volver a documentarse por completo incluso en este caso un enfoque inteligente es recortar la documentación a un mínimo esencial. Cada una de estas opciones es viable. Una organización de software debe elegir la más apropiada para cada caso.

1.2.3. Ingeniería inversa

Es el proceso de análisis de un programa con el fin de crear una representación de programa con un nivel de abstracción más elevado que el código fuente.

La Ingeniería inversa es un proceso de recuperación de diseño. Con las herramientas de la ingeniería inversa se extraerá del programa existente información del diseño arquitectónico y de proceso, e información de los datos.

1.2.4. Reestructuración de código

El tipo más común de reingeniería es la reestructuración de código, se puede hacer con módulos individuales que se codifican de una manera que dificultan comprenderlos, probarlos y mantenerlos.

1.2.5. Reestructuración de datos

La reestructuración de datos es una actividad de reingeniería a gran escala. En la mayoría de los casos, la reestructuración de datos comienza con una actividad de ingeniería inversa. La arquitectura de datos actual se analiza con minuciosidad y se define los modelos de datos necesarios, se identifican los objetivos de datos y los atributos, y después se revisa la calidad de las estructuras de datos existentes.

1.2.6. Ingeniería directa

La ingeniería directa no solo recupera la información de diseño a partir del software existente, también utiliza esta información para alterar o reconstruir el sistema existente con la finalidad de mejorar su calidad global. En la mayoría de los casos el software sometido a reingeniería vuelve a implementar la función del sistema existente y también añade nuevas funciones o mejoras.

2. Reutilización

2.1. Definición

Es el proceso de creación de sistemas de software a partir de un software existente, en lugar de tener que rediseñar desde el principio.

2.2. Elementos

- Especificaciones de requerimientos previamente concebidas
- Diseños previamente definidos.
- Código probado y depurado con anterioridad.
- Planes y casos de prueba previamente utilizados.
- Paquetes de software de proposito general.

2.3. Ventajas

- Reducir el tiempo de desarrollo y costos
- Incrementar la productividad.
- Facilitar la compartición de productos del ciclo de vida.
- Mayor eficiencia y fiabilidad.
- Consistencia y la familiaridad, los patrones dentro del software serán más consistentes, tendiendo a facilitar el mantenimiento del producto.

2.4. Desventajas

- Necesidad de invertir antes de obtener resultados y carencia de métodos adecuados.
- Necesidad de formar al personal.
- Dificultad para institucionalizar los procesos.

2.5. Tipos de reutilización

2.5.1. Oportunista

El ingeniero de software reutiliza piezas que él sabe que se ajustan al problema

2.5.2. Sistemática

- Esfuerzo a nivel organizacional y planificado de antemano
- Todo componente reutilizado ha de ser ideado, a priori, para ser reutilizado
- Implica inversiones iniciales para recoger frutos en el futuro
- Diseñar componentes genéricos para que sean reutilizados con facilidad

2.5.3. Bottom-Up

- Se desarrollan pequeños componentes para una determinada aplicación
- Se incorpora a un repositorio

2.5.4. Top-Down

- Se determinan las piezas necesarias que encajan unas con otras
- Se van desarrollando poco a poco
- Requiere alta inversión a comienzo
- Se recogerán beneficios en el futuro