Analisis de sistemas

2do Sistemas Nocturno

Sebastian Jaramillo

Herramientas CASE

Ingeniería Inversa

Reingeniería

**Herramientas CASE**

(Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadoras).

Son diversas Aplicaciones informáticas destinadas a aumentar la productividad en el Desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, Compilación automática, documentación o detección de errores entre otras.

**Estructura general de una herramienta CASE**

La estructura CASE se basa en la siguiente terminología :

* CASE de alto nivel son aquellas herramientas que automatizan o apoyan las fases finales o superiores del ciclo de vida del desarrollo de sistemas como la planificación de sistemas, el análisis de sistemas y el diseño de sistemas.
* CASE de bajo nivel son aquellas herramientas que automatizan o apoyan las fases finales o inferiores del ciclo de vida como el diseño detallado de sistemas, la implantación de sistemas y el soporte de sistemas.
* CASE cruzado de ciclo de vida se aplica a aquellas herramientas que apoyan actividades que tienen lugar a lo largo de todo el ciclo de vida, se incluyen actividades como la gestión de proyectos y la estimación.

**Clasificación**

La clasificación basada en las fases del ciclo de desarrollo cubre:

* Upper CASE (U-CASE), herramientas que ayudan en las fases de planificación, análisis de requisitos y estrategia del desarrollo, usando, entre otros diagramas UML.
* Middle CASE (M-CASE), herramientas para automatizar tareas en el análisis y diseño de la aplicación.
* Lower CASE (L-CASE), herramientas que semi-automatizan la generación de código, crean programas de detección de errores, soportan la depuración de programas y pruebas. Además automatizan la documentación completa de la aplicación. Aquí pueden incluirse las herramientas de Desarrollo rápido de aplicaciones.

Existen otros nombres que se le dan a este tipo de herramientas, y que no es una clasificación excluyente entre sí, ni con la anterior:

* Integrated CASE (I-CASE), herramientas que engloban todo el proceso de desarrollo software, desde análisis hasta implementación.
* MetaCASE, herramientas que permiten la definición de nuestra propia técnica de modelado, los elementos permitidos del metamodelo generado se guardan en un repositorio y pueden ser usados por otros analistas, es decir, es como si definiéramos nuestro propio UML, con nuestros elementos, restricciones y relaciones posibles.
* CAST (Computer-Aided Software Testing), herramientas de soporte a la prueba de software.
* IPSE (Integrated Programming Support Environment), herramientas que soportan todo el ciclo de vida, incluyen componentes para la gestión de proyectos y gestión de la configuración.

Por funcionalidad podríamos diferenciar algunas como:

* Herramientas de generación semiautomática de código.
* Editores UML.
* Herramientas de Refactorización de código.
* Herramientas de mantenimiento como los sistemas de control de versiones.

**La ingeniería inversa.**

Es el proceso de construir especificaciones de un mayor nivel de abstracción partiendo del código fuente de un sistema software o cualquier otro producto (se puede utilizar como punto de partida cualquier otro elemento de diseño.

Estas especificaciones pueden volver a ser utilizadas para construir una nueva implementación del sistema utilizando por ejemplo, técnicas de ingeniería directa.

**Beneficios**

La aplicación de ingeniería inversa nunca cambia la funcionalidad del software sino que permite obtener productos que indican cómo se ha construido el mismo. Permite obtener los siguientes beneficios:

* Reducir la complejidad del sistema: al intentar comprender el software se facilita su mantenimiento y la complejidad existente disminuye.
* Generar diferentes alternativas: del punto de partida del proceso, principalmente código fuente, se generan representaciones gráficas lo que facilita su comprensión.
* Recuperar y/o actualizar la información perdida (cambios que no se documentaron en su momento): en la evolución del sistema se realizan cambios que no se suele actualizar en las representaciones de nivel de abstracción más alto, para lo cual se utiliza la recuperación de diseño.
* Detectar efectos laterales: los cambios que se puedan realizar en un sistema puede conducirnos a que surjan efectos no deseados, esta serie de anomalías puede ser detectados por la ingeniería inversa.
* Facilitar la reutilización: por medio de la ingeniería inversa se pueden detectar componentes de posible reutilización de sistemas existentes, pudiendo aumentar la productividad, reducir los costes y los riesgos de mantenimiento.
* La finalidad de la ingeniería inversa es la de desentrañar los misterios y secretos de los sistemas en uso a partir del código. Para ello, se emplean una serie de herramientas que extraen información de los datos, procedimientos y arquitectura del sistema existente.

**Tipos**

* Ingeniería inversa de datos: Se aplica sobre algún código de bases datos (aplicación, código SQL, etc) para obtener los modelos relacionales o sobre el modelo relacional para obtener el diagrama entidad-relación
* Ingeniería inversa de lógica o de proceso: Cuando la ingeniería inversa se aplica sobre código de un programa para averiguar su lógica o sobre cualquier documento de diseño para obtener documentos de análisis o de requisitos.
* Ingeniería inversa de interfaces de usuario: Se aplica con objeto de mantener la lógica interna del programa para obtener los modelos y especificaciones que sirvieron de base para la construcción de la misma, con objeto de tomarlas como punto de partida en procesos de ingeniería directa que permitan modificar dicha interfaz.

**Reingeniería**

Es el método mediante el cual una organización puede lograr un cambio radical de rendimiento medido por el costo, tiempo de ciclo, servicio y calidad, mediante la aplicación de varias herramientas y técnicas enfocadas en el negocio como una serie de procesos del producto principal del negocio.

**Objetivos de la Reingenieria**

La Reingeniería persigue definir criterios de simplificación y optimización que permiten alcanzar las metas del cambio:

* Racionalizar las operaciones
* Reducir los costos
* Mejorar la calidad
* Aumentar los ingresos
* Mejorar la orientación hacia los clientes.
* Principios de la Reingenieria
* Habilidad para utilizar el cambio con eficiencia.
* Utilizar el cambio de manera continua; habilidad de cambiar con rapidez para ganar ventaja competitiva. Desarrollo de enfoques para aplicar Reingeniería, con base en el concepto de cambio continuo y dirigido (Paradigma Cambiante).

**Fuentes**

**Monografias.com** Artículo interesante con algo de la historia y génesis de las herramientas.

**Universidad Jaume** I Definición de glosario, uso en el contexto de bases de datos.

**Herramientas CASE** Un directorio de herramientas CASE, bastante útil y actualizado, con una wiki y folksonomy sobre herramientas CASE.