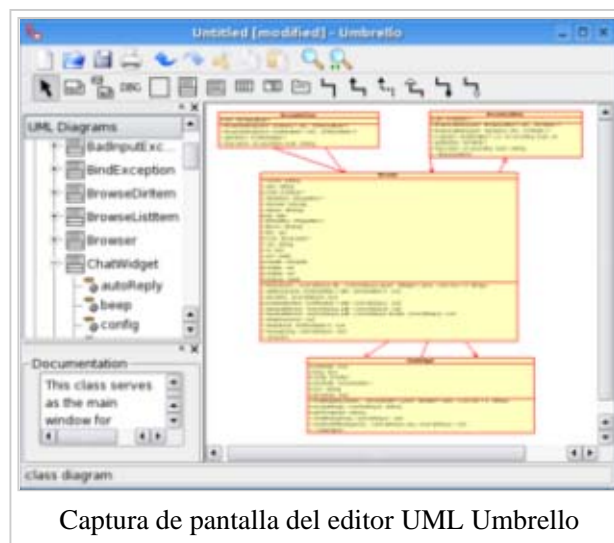


Herramienta CASE

De Wikipedia, la enciclopedia libre

Las **herramientas CASE** (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) 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 proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.



Captura de pantalla del editor UML Umbrello

Sistema de software que intenta proporcionar ayuda automatizada a las actividades del proceso de software. Los sistemas CASE a menudo se utilizan como apoyo al método.

Contenido

- 1 Historia
- 2 Objetivos
- 3 Clasificación
- 4 Véase también
- 5 Enlaces externos
 - 5.1 En Español
 - 5.2 En Ingles

Historia

Ya en los años 70 un proyecto llamado ISDOS diseñó un lenguaje y por lo tanto un producto que analizaba la relación existente entre los requisitos de un problema y las necesidades que éstos generaban, el lenguaje en cuestión se denominaba PSL (Problem Statement Language) y la aplicación que ayudaba a buscar las necesidades de los diseñadores PSA (Problem Statement Analyzer).

Aunque éstos son los inicios de las herramientas informáticas que ayudan a crear nuevos proyectos informáticos, la primera herramienta CASE fue Excelerator que salió a la luz en el año 1984 y trabajaba bajo una plataforma PC.

Las herramientas CASE alcanzaron su techo a principios de los años 90. En la época en la que IBM había conseguido una alianza con la empresa de software AD/Cycle para trabajar con sus mainframes, estos dos gigantes trabajaban con herramientas CASE que abarcaban todo el ciclo de vida del software. Pero poco a poco los mainframes han ido siendo menos utilizados y actualmente el mercado de las Big CASE ha muerto completamente abriendo el mercado de diversas

herramientas más específicas para cada fase del ciclo de vida del software.

Objetivos

1. Mejorar la productividad en el desarrollo y mantenimiento del software.
2. Aumentar la calidad del software.
3. Reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
4. Mejorar la planificación de un proyecto
5. Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
6. Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
7. Ayuda a la reutilización del software, portabilidad y estandarización de la documentación
8. Gestión global en todas las fases de desarrollo de software con una misma herramienta.
9. Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Clasificación

Aunque no es fácil y no existe una forma única de clasificarlas, las herramientas CASE se pueden clasificar teniendo en cuenta los siguientes parámetros:

1. Las plataformas que soportan.
2. Las fases del ciclo de vida del desarrollo de sistemas que cubren.
3. La arquitectura de las aplicaciones que producen.
4. Su funcionalidad.

La siguiente clasificación es la más habitual basada en las fases del ciclo de desarrollo que cubren:

- *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 si, 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.

Véase también

- Entorno de desarrollo integrado
- Ingeniería del software
- Lenguaje Unificado de Modelado

Enlaces externos

En Español

- monografias.com artículo interesante con algo de la historia y génesis de las herramientas CASE
- Universidad Jaume I Definición de glosario, uso en el contexto de bases de datos.

En Ingles

- Herramientas CASE un directorio de herramientas CASE, bastante útil y actualizado, con una wiki y folksonomy sobre herramientas CASE.
- CASE tool index - un poco desactualizada pero útil
- UML CASE tools -otra lista muy interesante orientada a herramientas CASE especificas para trabajar con UML. Está un poco desactualizada
- Herramientas CASE ágil Algunos lineamientos útiles para trabajar y seleccionar herramientas CASE de manera ágil

Obtenido de "http://es.wikipedia.org/wiki/Herramienta_CASE"

Categoría: CASE

- Esta página fue modificada por última vez el 01:09, 10 feb 2010.
- El texto está disponible bajo la Licencia Creative Commons Reconocimiento Compartir Igual 3.0; podrían ser aplicables cláusulas adicionales. Lee los términos de uso para más información.