



Narzędzia CASE

Dzięki narzędziom CASE projekty tworzy się dokładniej, a praca nad diagramami, sprawdzanie ich poprawności oraz śledzenie wykonanych testów jest prostsze i szybsze.

Dzięki narzędziom CASE projekty tworzy się dokładniej, a praca nad diagramami, sprawdzanie ich poprawności oraz śledzenie wykonanych testów jest prostsze i szybsze.

1. Wstęp

Narzędzia CASE (czyli Computer Aided Software Engineering lub Computer Aided System Engineering) to systemy komputerowe, przeznaczone do wspomagania rutynowych czynności procesu tworzenia oprogramowania. Dzięki nim projekty tworzy się dokładniej, a praca nad diagramami, sprawdzanie ich poprawności oraz śledzenie wykonanych testów jest prostsze i szybsze.

Kiedyś takie systemy wykorzystywały modele strukturalne procesów. Dzięki rozwojowi metod obiektowych, twórcy tych systemów wprowadzali także takie modele, lecz używali różnych metodologii. Dopiero gdy wymyślono UML – stał się on dobrym standardem, a jego obsługa implementowana jest we wszystkich liczących się CASEach.

Kiedyś systemy były wykonywane w technologii z góry wybranej – wybrać trzeba było środowisko systemowe, język programowania czy też bazę danych. Teraz można wybrać kilka technologii (J2EE, .NET) i oprzeć się na kilku silnikach bazodanowych (lub też zuniwersalizować kwestię wyboru bazy danych).

2. Podział narzędzi CASE

Systemy CASE można podzielić według faz cyklu życia systemu na: Upper-CASE i Lower-CASE, a także według zakresu zastosowań na pakiety narzędziowe oraz pakiety zintegrowane.

Upper-CASE wspomaga pierwsze fazy budowy systemu – analizę organizacyjną i

funkcjonalną i procesową, modelowanie funkcji, procesów, obiektów, modelowanie struktur i potrafi tworzyć wszelkie diagramy. Te narzędzia zajmują się bardziej opisem i modelowaniem rzeczywistości, modelowaniem struktury systemu, bez wszelkich faz implementacji.

Lower-CASE natomiast wspomaga rzeczywiste budowanie oprogramowania – modelowanie bazy danych, generowanie kodu i testy.

Czasami wyróżnia się także systemy **Middle-CASE**, które pozwalają określić samą strukturę systemu informatycznego, oraz **Integrated-CASE**, czyli systemy łączące Upper- i Lower-CASE.

3. Funkcje

Od systemów CASE wymagamy bardzo wiele. Wspomaganie w każdej fazie cyklu projektu jest inne i wymaga różnych funkcjonalności. Można jednak wyróżnić kilka standardowych modułów, których istnienie świadczy o zaawansowaniu danego systemu i spełnieniu wymagań użytkownika.

- **Słowniki danych (repozytoria)** – bazy wszelkich danych o tworzonym systemie wraz z narzędziami edytującymi, zarządzającymi i wyszukującymi te dane.
- **Edytor Notacji Graficznych** – program graficzny, umożliwiający tworzenie i edycję diagramów dla faz określania wymagań systemu, analizy i projektowania. Powinien też umożliwiać powiązania między symbolami w modelu a innymi, zdekomponowanymi modelami, oraz wydruk tych diagramów.
- **Moduł Kontroli Poprawności** – narzędzie do wykrywania i poprawiania błędów w diagramach i repozytoriach. Bardzo często działa w czasie rzeczywistym, co znacząco wpływa na komfort pracy.
- **Moduł Kontroli Jakości** – narzędzie do oceny pewnych ustalonych miar jakości projektu – np. stopnia złożoności lub powiązań składowych modelu.
- **Generator Raportów** – narzędzie tworzące dowolny raport na podstawie danych z repozytorium.
- **Generator Kodu** – narzędzie transformujące projekt na szkielet kodu w wybranym języku programowania. Usprawnia pracę programistów, pozwala na zautomatyzowanie pewnych fragmentów kodu, a także na uzupełnienie kodu o dodatkowe informacje ze słownika danych.
- **Generator Dokumentacji Technicznej** – generator ustandaryzowanych dokumentów, zawierających specyfikację, opisy faz projektu, diagramy oraz wybrane raporty.
- **Moduł Projektowania Interfejsu Użytkownika** – narzędzie do projektowania menu, okien dialogowych oraz innych elementów interfejsu użytkownika.

- **Moduł Inżynierii Odwrotnej** – narzędzie pozwalające odtworzyć słownika danych oraz diagramów, na podstawie kodu źródłowego lub struktury bazy danych.
- **Moduł Importu/Eksportu Danych** – narzędzie służące do wymiany danych z innymi CASE'ami czy też innymi programami.
- **Moduł Zarządzania Pracą Grupową** – narzędzie umożliwiające współpracę grupy osób podczas pracy nad projektem.

Gdy chcemy zacząć pracę nad projektem, możemy to zrobić tworząc dowolny model od podstaw, ale możemy też za pomocą inżynierii odwrotnej stworzyć model, opierając się o istniejące struktury bazy danych, kod źródłowy z klasami, czy też struktury w XMLu. Gdy już będziemy mieli modele i będziemy równolegle pracować nad kilkoma etapami, może się okazać, że potrzebujemy wprowadzić zmiany w kilku modelach. Dobry system CASE potrafi powiązać zmiany w tych modelach z koniecznymi zmianami w innych modelach oraz dokonać automatycznie odpowiednich korekt.

4. Popularne narzędzia

- **Eclipse** – darmowe, otwarte środowisko programistyczne dla Javy, które za pomocą platformy modelowania Eclipse (Eclipse Modeling Framework) może posłużyć do budowania oprogramowania, wykorzystując także UML. EMF posiada także generator kodu.
- **NetBeans** – otwarty projekt zawierający wiele narzędzi wspomagających tworzenie oprogramowania. Dodatkowo "Enterprise Pack" umożliwia modelowanie UML oraz użycie schematów XML.
- **StarUML** – otwarta, dostępna na zmodyfikowanej licencji GPL platforma UML/MDA dla systemu Windows, która umożliwia import projektów z takich komercyjnych aplikacji jak Rational Rose czy Borland Together. Zapewnia forward i reverse engineering kodu w Javie, C# i C++.
- **Borland Together** – rodzina programów integrujących środowisko IDE Javy z narzędziami do UMLa. Posiada m.in. funkcje modelowania danych, szablony kodu, generator dokumentacji, czy też moduł weryfikacji kodu.
- **Enterprise Architect** - profesjonalne narzędzie, działające na platformach Windows i Linux. Obsługuje UML 2.0.
- **IBM Rational Rose** – jedno z najstarszych, profesjonalnych narzędzi. Bardzo rozbudowane, obsługujące UML 2.0.

Licencja: [Creative Commons - użycie niekomercyjne - na tych samych warunkach](#)