



Computer Aided Software Engineering (CASE)

Difficulty Level : Medium • Last Updated : 03 Jan, 2023

Read

Discuss

Computer-aided software engineering (CASE) is the implementation of computer-facilitated tools and methods in software development. CASE is used to ensure high-quality and defect-free software. CASE ensures a check-pointed and disciplined approach and helps designers, developers, testers, managers, and others to see the project milestones during development.

CASE can also help as a warehouse for documents related to projects, like business plans, requirements, and design specifications. One of the major advantages of using CASE is the delivery of the final product, which is more likely to meet real-world requirements as it ensures that customers remain part of the process.

CASE illustrates a wide set of labor-saving tools that are used in software development. It generates a framework for organizing projects and to be helpful in enhancing productivity. There was more interest in the concept of CASE tools years ago, but less so today, as the tools have morphed into different functions, often in reaction to software developer needs. The concept of CASE also received a heavy dose of criticism after its release.

CASE Tools: The essential idea of CASE tools is that in-built programs can help to analyze developing systems in order to enhance quality and provide better outcomes. Throughout the 1990, CASE tool became part of the software lexicon, and big companies like IBM were using these kinds of tools to help create software.

Various tools are incorporated in CASE and are called CASE tools, which are used to support different stages and milestones in a software development life cycle.

 **Types of CASE Tools:**

Start Your Coding Journey Now!

[Login](#)[Register](#)

processes. It represents system elements, control flow and data flow among different software components and system structures in a pictorial form. For example, Flow Chart Maker tool for making state-of-the-art flowcharts.

2. **Computer Display and Report Generators:** These help in understanding the data requirements and the relationships involved.
3. **Analysis Tools:** It focuses on inconsistent, incorrect specifications involved in the diagram and data flow. It helps in collecting requirements, automatically check for any irregularity, imprecision in the diagrams, data redundancies, or erroneous omissions. For example:
 - (i) Accept 360, Accompa, CaseComplete for requirement analysis.
 - (ii) Visible Analyst for total analysis.
4. **Central Repository:** It provides a single point of storage for data diagrams, reports, and documents related to project management.
5. **Documentation Generators:** It helps in generating user and technical documentation as per standards. It creates documents for technical users and end users. For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.
6. **Code Generators:** It aids in the auto-generation of code, including definitions, with the help of designs, documents, and diagrams.

Advantages of the CASE approach:

- As the special emphasis is placed on the redesign as well as testing, the servicing cost of a product over its expected lifetime is considerably reduced.
 - The overall quality of the product is improved as an organized approach is undertaken during the process of development.
 - Chances to meet real-world requirements are more likely and easier with a computer-aided software engineering approach.
 - CASE indirectly provides an organization with a competitive advantage by helping ensure the development of high-quality products.
 - It provides better documentation.
- It improves accuracy.
- It provides intangible benefits.

Start Your Coding Journey Now!

- It impacts the style of working of the company.
- It reduces the drudgery in software engineer's work.
- It increases the speed of processing.
- It is easy to program software.

Disadvantages of the CASE approach:

- **Cost:** Using a case tool is very costly. Most firms engaged in software development on a small scale do not invest in CASE tools because they think that the benefit of CASE is justifiable only in the development of large systems.
- **Learning Curve:** In most cases, programmers' productivity may fall in the initial phase of implementation, because users need time to learn the technology. Many consultants offer training and on-site services that can be important to accelerate the learning curve and to the development and use of the CASE tools.
- **Tool Mix:** It is important to build an appropriate selection tool mix to urge cost advantage CASE integration and data integration across all platforms is extremely important.

Like 33

[Previous](#)

[Next](#)

Related Articles

1. [Difference between Software Engineering and Computer Engineering](#)
2. [Software Engineering | Introduction to Software Engineering](#)
3. [Software Engineering | Structured analysis and design with CASE tools](#)