Unit: IV
Lecture: 12
CASE (Computer Aided Software Engineering)

Computer-aided software engineering (CASE) is the implementation of computer-facilitated tools and methods in software development.

CASE tools are set of software application programs, which are used to automate SDLC activities. CASE tools are used by software project managers, analysts and engineers to develop software system.

There are number of CASE tools available to simplify various stages of Software Development Life Cycle such as Analysis tools, Design tools, Project Management tools, Database Management tools, Documentation tools are to name a few.

Use of CASE tools accelerates the development of project to produce desired result and helps to uncover flaws before moving ahead with next stage in software development.

CASE tools may support the following developmental steps:

- Creation of data flow and entity models
- Establishing a relationship between requirements and models
- Development of top-level design
- Development of functional and process description
- Development of test cases.

The CASE tools can help in automatically generating data base tables, forms and reports and user documentation. Thus, the CASE tools:

- support contemporary development of software systems, they may improve the quality of the software.
- help in automating the software development life cycles by use of certain standard methods
- create an organization wide environment that minimized repetitive work
- help developers to concentrate more on top level and more creative problem solving tasks
- support and improve the quality of documentation, testing process, project management and software maintenance.

Categories of CASE tools:

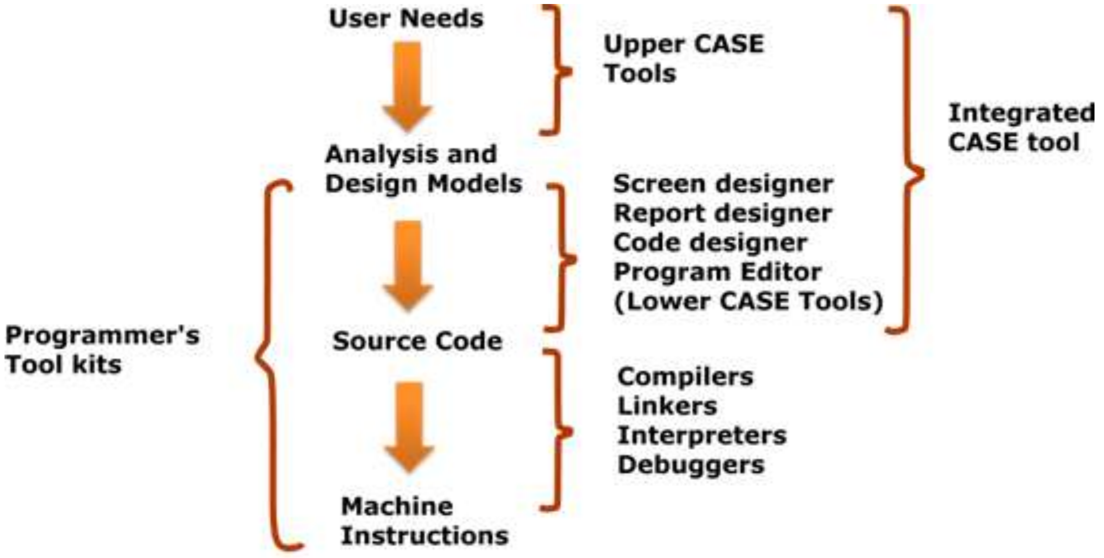
CASE tools are classified into the following categories:

1. Upper CASE tools
2. Lower CASE tools
3. Integrated CASE tools

Upper CASE: Upper CASE tools mainly focus on the analysis and design phases of software development. They include tools for analysis modeling, reports and forms generation.

Lower CASE: Lower CASE tools support implementation of system development. They include tools for coding, configuration management, etc.

Integrated CASE Tools: Integrated CASE tools help in providing linkages between the lower and upper CASE tools. Thus, creating a cohesive environment for software development when programming by lower CASE tools may automatically be generated for the design that has been developed in an upper CASE tool.



CASE Tool And Application Development

Types of CASE tools:

1. **Diagramming tools:** It helps in diagrammatic and graphical representations of the data and system processes. It represents system elements, control flow and data flow among different software

components and system structures in a pictorial form. e.g. 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.
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.
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:

- 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 increases the speed of processing.

Disadvantages of the CASE Approach:

- Using case tools 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.
- In most cases, programmers' productivity may fall in the initial phase of implementation, because users need time to learn the technology.