Practical 6

AIM: Study of Software Development tools and techniques (CASE).

CASE TOOLS:

History:

The term CASE was originally coined by software company, Nastec Corporation of Southfield, Michigan in 1982 with their original integrated graphics and text editor GraphiText, which also was the first microcomputer-based system to use hyperlinks to cross-reference text strings in documents — an early forerunner of today's web page link. GraphiText's successor product, DesignAid was the first microprocessor-based tool to logically and semantically evaluate software and system design diagrams and build a data dictionary. The next entrant into the market was Excelerator from Index Technology in Cambridge, Mass. While DesignAid ran on Convergent Technologies and later Burroughs Ngen networked microcomputers, Index launched Excelerator on the IBM PC/AT platform. While, at the time of launch, and for several years, the IBM platform did not support networking or a centralized database as did the Convergent Technologies or Burroughs machines, the allure of IBM was strong, and Excelerator came to prominence. Hot on the heels of Excelerator were a rash of offerings from companies such as Knowledgeware, Texas Instrument's IEF and Accenture's FOUNDATION toolset (METHOD/1, DESIGN/1, INSTALL/1, FCP)

Introduction:

Computer-Aided Software Engineering (CASE) technologies are tools that provide automated assistance for software development**.** The goal of introducing CASE tools is the reduction of the time and cost of software development and the enhancement of the quality of the systems developed. The interest in CASE tools and environments is based on expectations about increasing productivity, improving product quality, facilitating maintenance, and making software engineers' task less odious and more enjoyable. A survey of the CASE tool market showed that the annual worldwide market for CASE fools was $4.8 billion in 1990 and grew to $12.11 billion in 1995. Behind such a prosperous CASE market, however, another result gained from the real investigation about the use of CASE tools revealed that CASE tools seem to be sparsely used after being bought in many enterprises.

CASE is the use of computer-based support in the software development process; a CASE tool is a computer-based product aimed at supporting one or more software engineering activities within a software development process; a CASE environment is a collection of CASE tools and other components together with an integration approach that supports most or all of the interactions that occur among the environment components, and between the users of the environment and the environment itself.

Are CASE Tools being used?

Many prior studies have reported limited use of CASE tools. In a survey of 53 companies, found that 39 (73.5%) had never used CASE. Of the 14 companies who had tried CASE, five had subsequently abandoned use of the tools. People within these fourteen companies believed that use of CASE tools improved documentation quality, improved analysis, and resulted in systems that were easier to test and maintain. However, they also found use of CASE tools difficult and time consuming. In another cross organization survey, found that only 24% of companies were using CASE tools. In a follow-up survey of thirteen managers who had been using CASE tools two years earlier, reported that continued CASE use could only be verified for four managers. The reasons for abandonment included cost, lack of measurable turns, and unrealistic expectations. Looking within organizations that used CASE tools it was found that large numbers of their systems developers were not using CASE tools.

Popular features of CASE tools

The term Computer-Aided Software Engineering (CASE) encompasses many different products with different functionalities. In the International Workshop on Computer-Aided Software Engineering (IWCASE) definition of CASE very broad terms are used: "...tools and methods to support engineering approach to systems development at all stages of the process". When the term CASE is used, it is important to clarify what is being discussed. Most classifications of CASE tools start by considering whether the tool is upper CASE, lower CASE, or integrated CASE [3]. An upper CASE tool (front end CASE) provides support for the early stages in the systems development life cycle such as requirements analysis and design. A lower CASE tool (back end CASE) provides support for the later stages in the life cycle such as code generation and testing. Integrated CASE tools support both the early and later stages. Further classifications usually list which functionalities are supported by the tool, such as data flow diagrams, entity relationships data models, etc. provides a different type of model of CASE functionality which helps organize CASE tools.

Automated Diagram Support

CASE Tools offer an excellent array of features that support the development and business community through its Automated Diagram Support feature. The various popular features that aid the development community are listed below:

- Checks for syntactic correctness
- Data dictionary support
- Checks for consistency and completeness
- Navigation to linked diagrams
- Layering
- Requirements traceability
- Automatic report generation
- System simulation
- Performance analysis

CASE Tools and its scope

CASE technology is the automation of step by step methodologies for software and system development. CASE tools are characterized by the stage or stages of software development life cycle on which they focus. Since different tools covering different stages share common information, it is required that they integrate through some central repository system (data dictionary) to have a consistent view of such information. In phases of software development life cycle integrated through a central data dictionary. Case Tools are used in many ways in our organizations. Case tools can be broadly classed into these broader areas:

- Requirement Analysis Tool
- Structure Analysis Tool
- Software Design Tool
- Code Generation Tool
- Test Case Generation Tool
- Document Production Tool
- Reverse Engineering Tool

While many organizations still use the SDLC methodology, it is often supplemented with other methods. Many systems developers use the CASE tools in various stages of the Software Development Life Cycle. They mainly use it while developing the following methodologies:

- Life Cycle
- Object-oriented Approach
- Rapid Applications Development (RAD)
- Prototyping
- Joint Applications Development (JAD)

The job of a systems developer may contain requirements analysis, process design, data design, and programming among other activities. But, not all systems developers do the same activities. One may spend most of his or her time on analysis; another, on design. The various activities that the system developers involve include Systems Analysis (including feasibility studies and requirements definition), Systems Design (including user interface, data, and process design), Programming (or generating code),

Testing, Supervisory or other management tasks and Maintenance. CASE tools play an important role in helping the system developers to perform the task efficiently.