**ST. XAVIER’S COLLEGE**

**Affiliated to Tribhuvan University**

Maitighar, Kathmandu



**LAB ASSIGNMENT#7**

**Submitted by**

**Aishwarya Rai**

**014BIM002**

**Submitted to**

**Mr.Balkrishna Subedi**

**Maitighar, Kathmandu, Nepal**

**July 3rd , 2017**

**CASE (computer-aided software engineering)**

Definition

CASE (computer-aided software engineering) is the use of a computer-assisted method to organize and control the development of software, especially on large, complex projects involving many software components and people.

Using CASE allows designers, code writers, testers, planners, and managers to share a common view of where a project stands at each stage of development. CASE helps ensure a disciplined, check-pointed process. A CASE tool may portray progress (or lack of it) graphically.

It may also serve as a repository for or be linked to document and program libraries containing the project's business plans, design requirements, design specifications, detailed code specifications, the code units, test cases and results, and marketing and service plans. (margaret, 2014)

## CASE Tools

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.

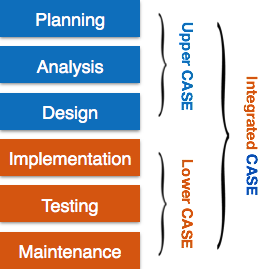
There are number of CASE tools available to simplify various stages of Software Development Life Cycle such as Analysis tools, Design tools, Project management tools, Database Management tools, Documentation tools are to name a few.

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. (john, 2012)

## Components of CASE Tools

CASE tools can be broadly divided into the following parts based on their use at a particular SDLC stage:

* **Central Repository** - CASE tools require a central repository, which can serve as a source of common, integrated and consistent information. Central repository is a central place of storage where product specifications, requirement documents, related reports and diagrams, other useful information regarding management is stored. Central repository also serves as data dictionary.



* **Upper Case Tools** - Upper CASE tools are used in planning, analysis and design stages of SDLC.
* **Lower Case Tools** - Lower CASE tools are used in implementation, testing and maintenance.
* **Integrated Case Tools** - Integrated CASE tools are helpful in all the stages of SDLC, from Requirement gathering to Testing and documentation.

CASE tools can be grouped together if they have similar functionality, process activities and capability of getting integrated with other tools. (john, 2012)

## Case Tools Types

Now we briefly go through various CASE tools

### Diagram tools

These tools are used to represent system components, data and control flow among various software components and system structure in a graphical form. For example, Flow Chart Maker tool for creating state-of-the-art flowcharts.

### Process Modeling Tools

Process modeling is method to create software process model, which is used to develop the software. Process modeling tools help the managers to choose a process model or modify it as per the requirement of software product. For example, EPF Composer

### Project Management Tools

These tools are used for project planning, cost and effort estimation, project scheduling and resource planning. Managers have to strictly comply project execution with every mentioned step in software project management. Project management tools help in storing and sharing project information in real-time throughout the organization. For example, Creative Pro Office, Trac Project, Basecamp.

### Documentation Tools

Documentation in a software project starts prior to the software process, goes throughout all phases of SDLC and after the completion of the project.

Documentation tools generate documents for technical users and end users. Technical users are mostly in-house professionals of the development team who refer to system manual, reference manual, training manual, installation manuals etc. The end user documents describe the functioning and how-to of the system such as user manual. For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.

### Analysis Tools

These tools help to gather requirements, automatically check for any inconsistency, inaccuracy in the diagrams, data redundancies or erroneous omissions. For example, Accept 360, Accompa, CaseComplete for requirement analysis, Visible Analyst for total analysis.

### Design Tools

These tools help software designers to design the block structure of the software, which may further be broken down in smaller modules using refinement techniques. These tools provides detailing of each module and interconnections among modules. For example, Animated Software Design

### Configuration Management Tools

An instance of software is released under one version. Configuration Management tools deal with –

* Version and revision management
* Baseline configuration management
* Change control management

CASE tools help in this by automatic tracking, version management and release management. For example, Fossil, Git, Accu REV.

### Change Control Tools

These tools are considered as a part of configuration management tools. They deal with changes made to the software after its baseline is fixed or when the software is first released. CASE tools automate change tracking, file management, code management and more. It also helps in enforcing change policy of the organization.

### Programming Tools

These tools consist of programming environments like IDE (Integrated Development Environment), in-built modules library and simulation tools. These tools provide comprehensive aid in building software product and include features for simulation and testing. For example, Cscope to search code in C, Eclipse.

### Prototyping Tools

Software prototype is simulated version of the intended software product. Prototype provides initial look and feel of the product and simulates few aspect of actual product.

Prototyping CASE tools essentially come with graphical libraries. They can create hardware independent user interfaces and design. These tools help us to build rapid prototypes based on existing information. In addition, they provide simulation of software prototype. For example, Serena prototype composer, Mockup Builder.

### Web Development Tools

These tools assist in designing web pages with all allied elements like forms, text, script, graphic and so on. Web tools also provide live preview of what is being developed and how will it look after completion. For example, Fontello, Adobe Edge Inspect, Foundation 3, Brackets.

### Quality Assurance Tools

Quality assurance in a software organization is monitoring the engineering process and methods adopted to develop the software product in order to ensure conformance of quality as per organization standards. QA tools consist of configuration and change control tools and software testing tools. For example, SoapTest, AppsWatch, JMeter.

### Maintenance Tools

Software maintenance includes modifications in the software product after it is delivered. Automatic logging and error reporting techniques, automatic error ticket generation and root cause Analysis are few CASE tools, which help software organization in maintenance phase of SDLC. For example, Bugzilla for defect tracking, HP Quality Center. (john, 2012)

**The advantages of the CASE approach include:**

* As special emphasis is placed on 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. (james, 2014)

**Disadvantages**  
There are several disadvantage to using CASE tools, but these drawbacks are common to any situation where a specialized tool is used.

* One of the disadvantage is that CASE tools do not necessarily prevent people from making bad designs. The output from the CASE tool may look nice, but if it was poorly done, it will still be bad.
* Another disadvantage has to do with training. If a person is new to using CASE tool, they might need a lot of time to study the concept. They most probably need a guidance from other people that used the tool before. But if you already know how to use one tool, learning how to use a new one is not as difficult because the same concepts still apply. If the CASE tool undergoes a major upgrade/version release, there may be old features lost and new features gained, neither of which may be desirable for your user community.
* Another disadvantage is the cost of using these tools, it may vary. Sometimes, we can just use PowerPoint to draw simple relationships and table. The database design may be a one-time effort, so why not use something cheaper. In addition to the cost of the tools is overhead of applying patches. The number of users may be small, but nonetheless, applying a patch is one more thing to keep track of. (unknown, 2010)

**which factors may contribute to the successful adoption of CASE tools within an organisation?**

The answer to the above question is as following:

* There should be established methods and procedures across the organisation that can be supported by the tools.
* Adequate investment should be put into training managers and developers.
* Deployment should occur over a long period of time, as the benefits of CASE tools are not short term.
* Clear procedures should be established for the use of different tools within development and standards should be followed for information exchange among tools.
* To overcome some of the deficiencies of current tools, they should be integrated within practices that support project management and customer/developer communication.