Computer Aided Software Engineering (CASE) Tools /lec2

Marrwa alabajee

Contents

Advantage of CASE

There are many advantages of using CASE, Some of them are:

1. A key benefit of using a CASE environment is cost saving through all development phases. Different studies carry out to measure the impact of CASE put the effort reduction between 30% to 40%.

Advantage of CASE

There are many advantages of using CASE, Some of them are:

2. Use of CASE tools leads to considerable improvements to quality, This is mainly due the chances of human error are considerably reduced also CASE tools help to ensure that quality is designed in before the product is built.

Advantage of CASE

3. CASE tools help produce high quality and consistent documents. Since the important data relating to a software product are maintained in a central repository, redundancy in the stored data is reduced and therefore chances of inconsistent documentation is reduced to a great extent.

Advantage of CASE

4. CASE provides the software engineer with the ability to automate manual activities and to improve engineering insight , CASE tools have reduced the hard work of a software engineer.

Advantage of CASE

5. CASE tools have led to revolutionary cost saving in software maintenance efforts. This happen not only due to the tremendous value of a CASE environment in traceability and consistency checks, but also due to the systematic information capture during the various phases of software development as a result of adhering to a CASE environment.

6. information is illustrated through diagrams, which are easier to understand.

Advantage of CASE

7. Introduction of a CASE environment has an impact on the style of working of a company, and makes it oriented towards the structured and orderly approach, because the CASE described as software tools for enterprise support consisting of enterprise strategic planning, project planning, else.

8. The speed during the system development increased, With CASE tools, tasks can be completed and altered much faster.

Advantage of CASE

9. Many modern CASE tools that support object oriented systems development, supports round-trip engineering (a development technique). For example, a tool that supports UML diagrams can generate programming code that can be modified by the programmers, at which point the UML diagrams will need to be changed as they no longer accurately represent the code. Most tools for UML also supports the generation of UML diagrams from code.

In this way, the system can evolve (improve) via diagrams and via code. This is called round-trip engineering.

Disadvantages of CASE Tools

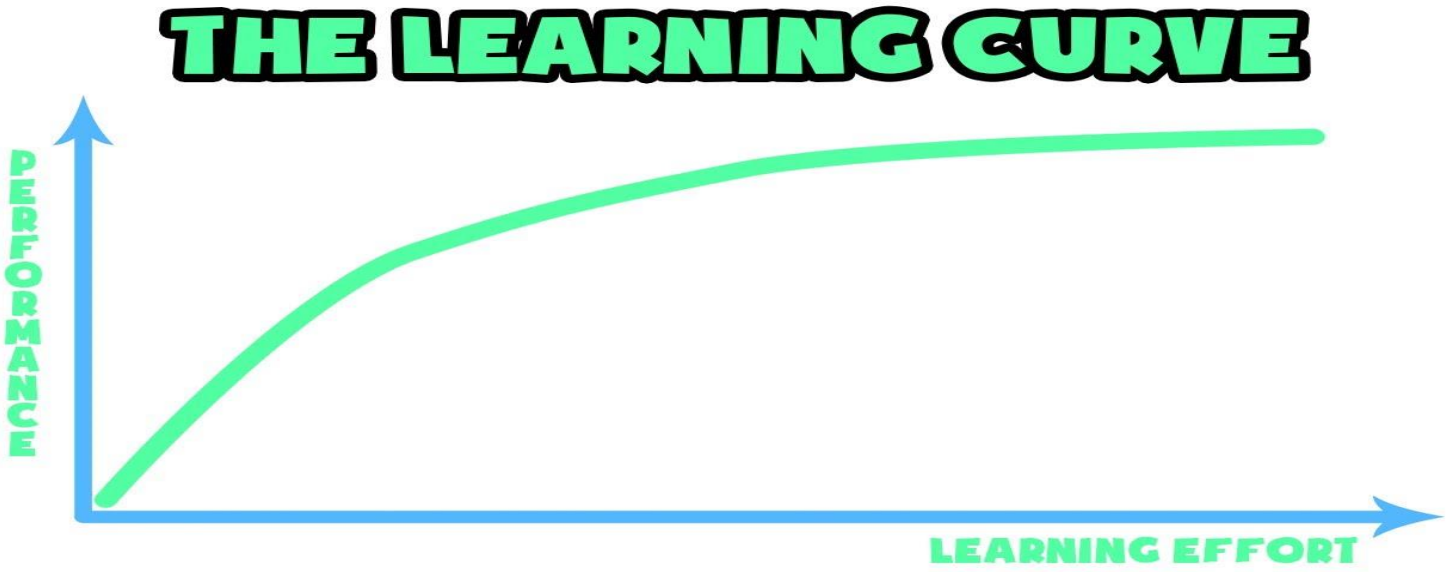
The major disadvantages of using CASE Tools in Software Engineering are :

1. Cost : The CASE tools available for use are very costly. Thus, most of the small scale organizations that develop software do not tend to invest in CASE tools because of the thinking that the benefits of CASE tools are worthed only in large organizations that develop big projects.

Disadvantages of CASE Tools

The major disadvantages of using CASE Tools in Software Engineering are :

2. Learning Curve : The programmer need time to learn how to use the CASE tools thus, the programmer productivity may fall in the initial phases of implementation.



Disadvantages of CASE Tools

3. Tool Mix : For getting the cost advantage of CASE integration and data integration it is important to make **an important selection of tool mix.** The ability of sharing the results of task performed with one CASE tool with another CASE tool is the most important type of CASE integration.

4. Trained maintenance staffs are needed , The systems that are produced primarily require great cost to maintain.

5. It requires more perfect and extensive definition of the user's needs.

Characteristics of CASE Tools

CASE tool should have the following characteristics :

1) Standard Methodology :

A CASE tool should support standard software development methodologies and modelling techniques. Presently, CASE tools use UML.

2) Flexibility :

CASE tool must provide flexibility and options to the user for editors and other tools.

Characteristics of CASE Tools

3) Strong Integration :

CASE tools must be integrated with all stages of software development. This means that if a change is made in a model, it must reflect in the code documentation and all related design. Hence, this offers an organized environment for software generation.

4) Support for Reverse Engineering :

CASE tools should be as that it can create complex models from existing code.

5) Online Help :

CASE tools offer online tutorials.

Building blocks for CASE

- Computer aided software engineering can be as simple as a **single tool** that supports a specific software engineering activity or as complex as a complete **"environment"** that encompasses tools, a database, people, hardware, a network, operating systems, standards, and myriad other components.

- The building blocks for CASE are illustrated in Figure 1.1. Each building block forms a foundation for the next, with tools sitting at the top of the heap.

Building blocks for CASE

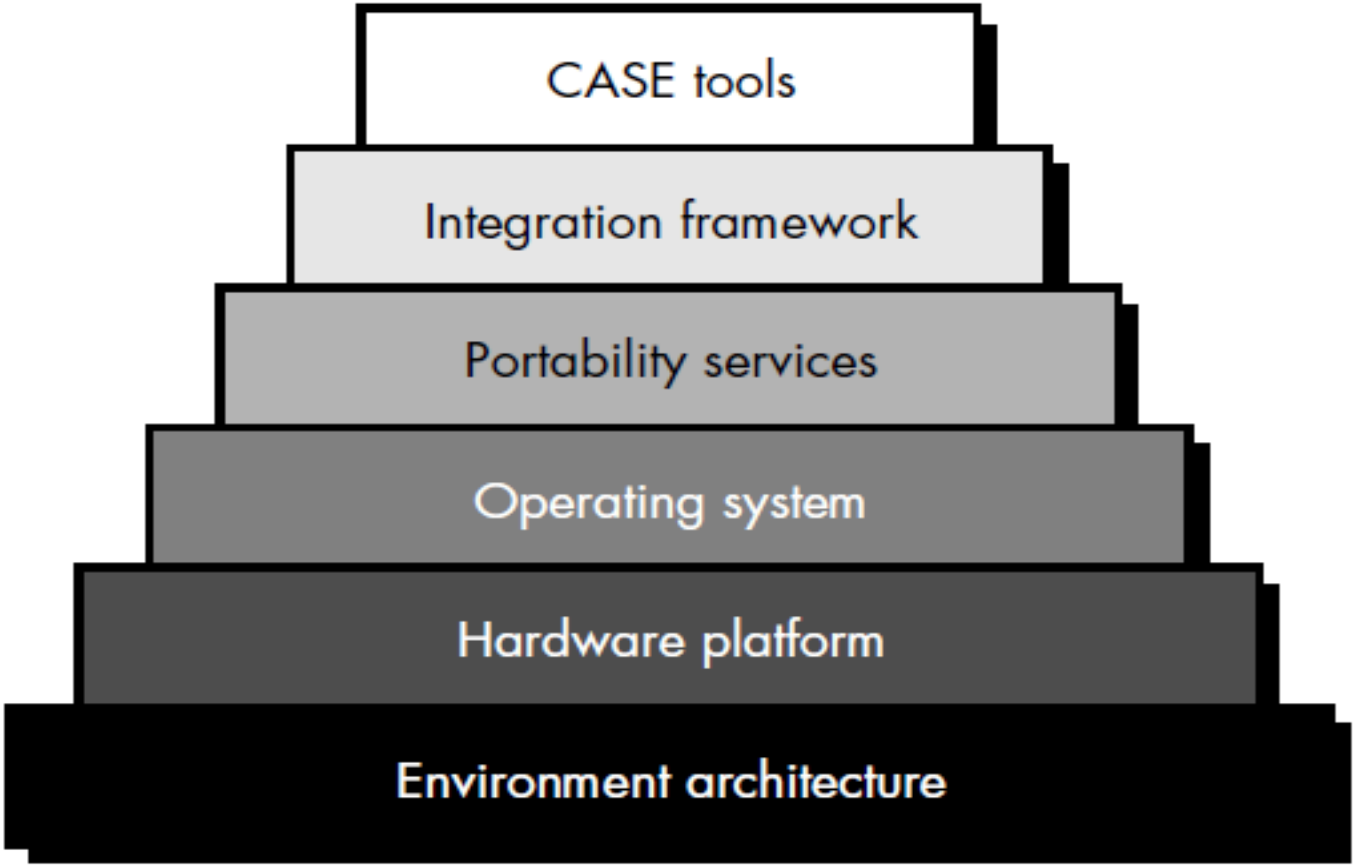


Figure 1.1 CASE building blocks

Building blocks for CASE

- It is interesting to note that the foundation for effective CASE environments has relatively little to do with software engineering tools themselves.

- Rather, successful environments for software engineering are built on an environment architecture that encompasses appropriate hardware and systems software. In addition, the environment architecture must consider the human work patterns that are applied during the software engineering process.

Building blocks for CASE

- The environment architecture, composed of the **hardware platform and system support** (including networking software, database management, and object management services), lays the ground work for CASE. But the CASE environment it self demands other building blocks.

- A set of portability services provides a bridge between CASE tools and their integration framework and the environment architecture.

Building blocks for CASE

- The integration framework is a collection of specialized programs that enables individual CASE tools to communicate with one another, to create a project database, and to exhibit the same look and feel to the end-user (the software engineer).

- Portability services allow CASE tools and their integration framework to migrate across different hardware platforms and operating systems without significant adaptive maintenance.

- The building blocks depicted in Figure 1.1 represent a comprehensive foundation for the integration of CASE tools.

.

Building blocks for CASE

- However, most CASE tools in use today have not been constructed using all these building blocks. In fact, some CASE tools remain "**point solutions**".

- That is, a tool is used to assist in a particular software engineering activity (e.g., analysis modeling) but does not directly communicate with other tools, is not tied into a project database, is not part of an integrated CASE environment (ICASE).

- Although this situation is not ideal, a CASE tool can be used quite effectively, even if it is a point solution. The relative levels of CASE integration are shown in Figure 1.2. At the low end of the integration spectrum is the individual (point solution) tool.

Building blocks for CASE

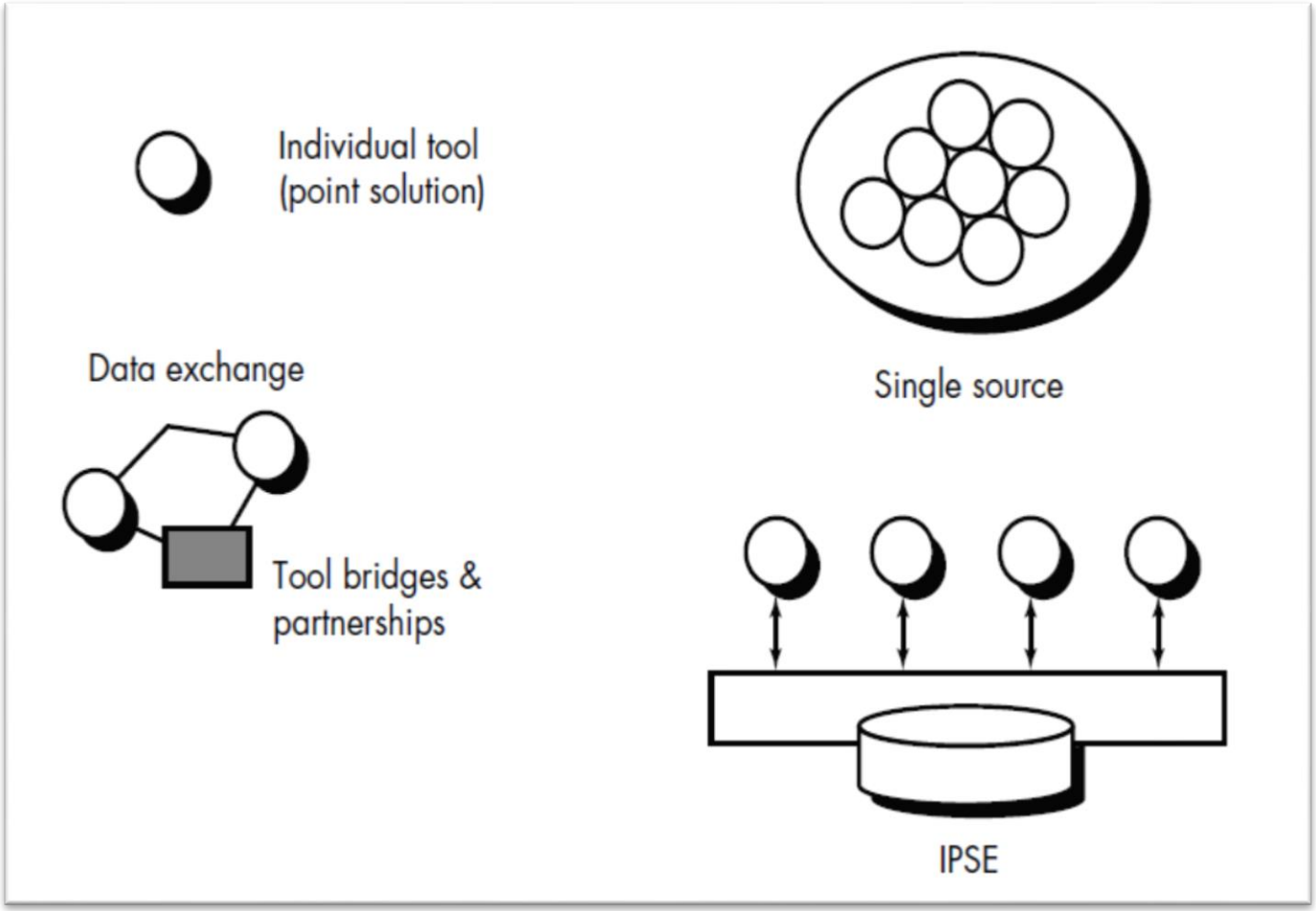


Figure 1.2 Integration option

Building blocks for CASE

- When individual tools provide facilities for data exchange (most do), the integration level is improved slightly.
- Such tools produce output in a **standard format** that should be compatible with other tools that can read the format.
- In some cases, the builders of complementary CASE tools work together to form a bridge between the tools (e.g., an analysis and design tool that is coupled with a code generator).
- Using this approach, the synergy between the tools can produce end products that would be difficult to create using either tool separately.

Building blocks for CASE

- Single-source integration occurs when a single CASE tools vendor integrates a number of different tools and sells them as a package.

- Although this approach is quite effective, the closed architecture of most single-source environments precludes easy addition of tools from other vendors.

Building blocks for CASE

- At the high end of the integration spectrum is the integrated project support environment (IPSE).

- Standards for each of the building blocks described previously have been created. CASE tool vendors use IPSE standards to build tools that will be compatible with the IPSE and therefore compatible with one another.

THANKS