1) **What are case tools? Explain its importance in se. OR explain building blocks for case.**

CASE Tools

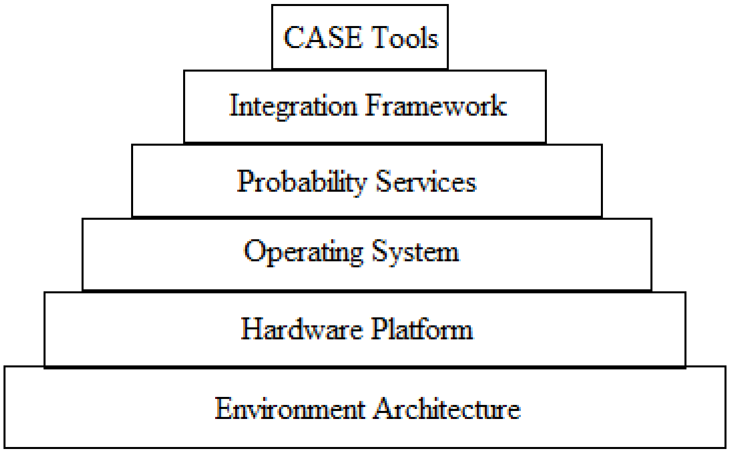
- A good workshop for a software engineer has three primary characteristics

  1) A collection of useful tools that will help in every step of building a product
  2) An organized layout that enables tools to be found quickly and used efficiently
  3) A skilled person who understands how to use the tools in an effective manner.

- Software engineers now recognize that they need more and varied tools along with an organized and efficient workshop in which to place the tools.
- The workshop for software engineering has been called an integrated project support environment and the tools that fill the workshop are collectively called computer aided software engineering.
- CASE provides the software engineer with the ability to automate manual activities and to improve engineering insight.
- Like computer aided engineering and design tools that are used by engineers in other disciplines, CASE tools help to ensure that quality is designed in before the product is built.

Building Blocks

- Computer aided software engineering can be simple that supports a specific software engineering activity.
- It can be complex that encompasses tools, a database, people, hardware, a network, operating systems, standards.
- Each building block forms a foundation for the next, with tools sitting at the top of the heap.
- The environment architecture composed of the hardware platform and system support. It lays the ground work for CASE. But the CASE environment itself demands other building blocks.
- A set of portability services provides a bridge between CASE tools and their integration framework and the environment architecture.
- 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.
- Portability services allow CASE tools and their integration framework to migrate across different hardware platforms and operating systems without significant adaptive maintenance.



Building Blocks of CASE

- A tool is used to assist in a particular software engineering activity but does not directly communicate with other tools, is not tied into a project database, is not part of an integrated CASE environment.
- When individual tools provide facilities for data exchange, 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.
- 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.

2) Explain taxonomy of case tools

A number of risks are inherent whenever we attempt to categorize CASE tools.

- CASE tools can be classified by function, by their role as instruments for managers or technical people, by their use in the various steps of the software engineering process, by the environment architecture that supports them, or even by their origin or cost. The taxonomy presented here uses functions a primary criterion.

1. **Business process engineering tools:**
   By modelling the strategic information requirements of an organization, business process engineering tools provide a "meta-model" from which specific information systems are derived.

2. **Process modelling and management tools:**
   If an organization works to improve a business (or software) process, it must first understand it. Process modelling tools are used to represent the key elements of a process so that it can be better understood.

3. **Project planning tools:**
   Tools in this category focus on two primary areas: software project effort and cost estimation and project scheduling. Estimation tools compute estimated effort, project duration, and recommended number of people for a project. Project scheduling tools enable the manager to define all project tasks.

4. **Risk analysis tools:**
   Identifying potential risks and developing a plan to mitigate, monitor, and manage them is of paramount importance in large projects. Risk analysis tools enable a project manager to build a risk table.

5. **Project management tools:**
   The project schedule and project plan must be tracked and monitored on a continuing basis. In addition, a manager should use tools to collect metrics that will ultimately provide an indication of software product quality.

6. **Requirements tracing tools:**
   When large systems are developed, things "fall into the cracks." That is, the delivered system does not fully meet customer specified requirements. The objective of requirements tracing tools is to provide a systematic approach to the isolation of requirements, beginning with the customer request for proposal or specification.

7. **Metrics and management tools:**
   Software metrics improve a manager's ability to control and coordinate the software engineering process and a practitioner's ability to improve the quality of the software that is produced.

8. **Documentation tools:**
   Document production and desktop publishing tools support nearly every aspect of software engineering and represent a substantial "lever-age" opportunity for all software developers like, word processing unit.

9. **System software tools:**
   CASE is a workstation technology. Therefore, the CASE environment must accommodate high-quality network system software, object management services, distributed component support, electronic mail, bulletin boards and other communication capabilities.

10. **Quality assurance tools:**
    The majority of CASE tools that claim to focus on quality assurance are actually metrics tools that audit source code to determine compliance with language standards.

11. **Database management tools:**
    Database management software serves as a foundation for the establishment of a CASE database (repository) that we have called the project database.

12. **Software configuration management tools:**
    Software configuration management lies at the kernel of every CASE environment. Tools can assist in all five major tasks identification, version control, change control, auditing, and status accounting.

13. **Analysis and design tools:**
    Analysis and design tools enable a software engineer to create models of the system to be built. The models contain a representation of data, function, and behaviour and characterizations of data, architectural, component-level, and interface design.

14. **PRO/SIM tools:**
    PRO/SIM (prototyping and simulation) tools provide the software engineer with the ability to predict the behaviour of a real-time system prior to the time that it is built.

15. **Interface design and development tools:**
    Interface design and development tools are actually a tool kit of software components (classes) such as menus, buttons, window structures, icons, scrolling mechanisms, device drivers, and so forth.

16. **Prototyping tools:**
    A variety of different prototyping tools can be used. Screen painters enable a software engineer to define screen layout rapidly for interactive applications. More sophisticated CASE prototyping tools enable the creation of a data design.

17. **Programming tools:**

The programming tools category encompasses the compilers, editors, and debuggers that are available to support most conventional programming languages.

18. **Web development tools:**

    The activities associated with Web engineering are sup-ported by a variety of tools for WebApp development. These include tools that assist in the generation of text, graphics, forms, scripts, applets, and other elements of a Web page.

19. **Integration and testing tools:**

    In their directory of software testing tools, Software Quality Engineering defines the following testing tools categories:

    - Data acquisition
    - Static measurement
    - Dynamic measurement
    - Simulation
    - Test management
    - Cross-functional tools

20. **Static analysis tools:**

    Static testing tools assist the software engineer in deriving test cases. Three different types of static testing tools are used in the industry: code-based testing tools specialized testing languages, and requirements-based testing tools.

21. **Dynamic analysis tools:**

    Dynamic testing tools interact with executing program, checking path coverage, testing assertions about the value of specific variables, and otherwise instrumenting the execution flow of the program.

22. **Test management tools:**

    Test management tools are used to control and coordinate software testing for each of the major testing steps. Tools in this category man-age and coordinate regression testing perform comparisons between actual and expected output.

23. **Client/server testing tools:**

    The c/s environment demands specialized testing tools that exercise the graphical user interface and the network communications requirements for client and server.

24. **Reengineering tools:**

    Tools for legacy software address a set of maintenance activities that currently absorb a significant percentage of all software-related effort.

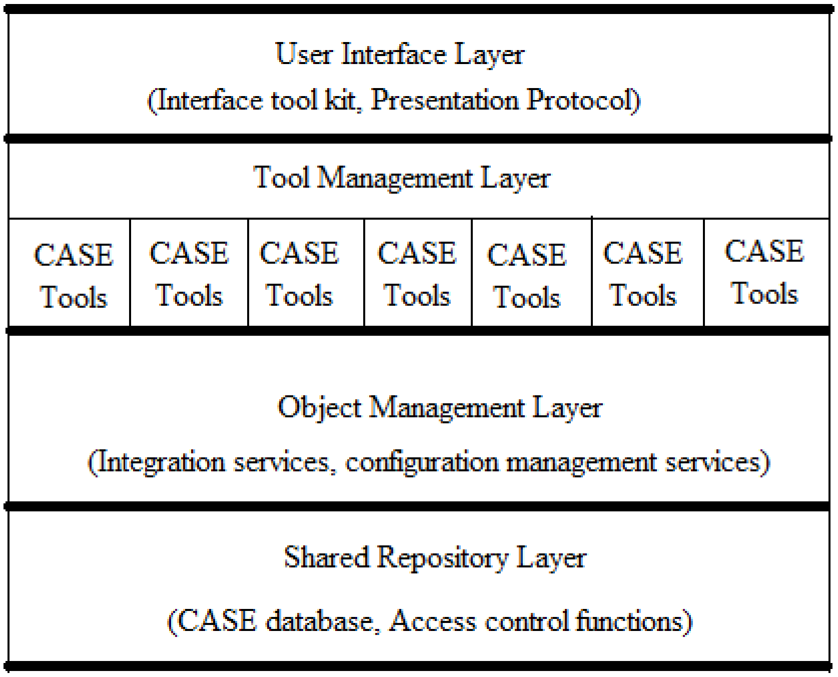
3) Describe integrated case environment.

Effective software project management focuses on the four P’s: people, product, process, and project.

- CASE tools create a pool of software engineering information. The integrated CASE environment allows a transfer of information into and out of this pool.

- **For such transfer there is a need for some architectural components are :**

Figure :Integrated CASE Environment



1. Database for storing of information.
2. Object management system. Before using the objects information can be transferred in the information pool.
3. Control mechanism
4. User interface

- The fig shows the simple model for integrated CASE environment. This environment consists of various levels.
- The user interface layer consists of interface tool kit and presentation protocols.
- The interface tool kit consists of collection of software required for interface management and display objects.
- The presentation protocol decides a common look and feel of the presentation interface. Then it comes a tools layer. IT consists of set of tools management services (TMS).
- The next layer is object management layer (OML). It performs the configuration management.
- The services of this layer allow the identification of all the configuration objects. So that the case tool can be plugged into the integrated CASE environment.
- The bottom most layer is shared repository layer. It consists of CASE database and access control functions. These access control functions help the object management layer to access the CASE database.