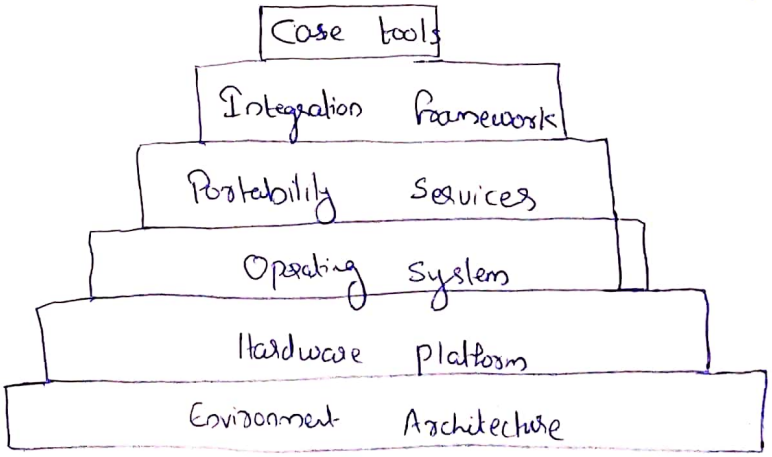
CASE BUILDING BLOCKS

Computer-Aided Software Engineering (CASE) tools assist software engineering managers & practitioners in every activity activity associated with the software engineering process.

CASE Building Blocks

→ CASE tools
→ Integration framework (specialized programs allowing CASE tools to communicate with one another).
→ Portability services (allow CASE tools & their integration frameworks to migrate across different operating systems & hardware platforms without significant adaptive maintenance.)
→ Operating system (database & object management services)
→ Hardware platform
→ Environmental architecture (hardware & system support).



CASE Tool Components:

→ Integration framework.
→ Specialized programs allowing CASE tools to communicate
→ Portability services.

→ Operating system
→ Database & object management services
→ Hardware platform

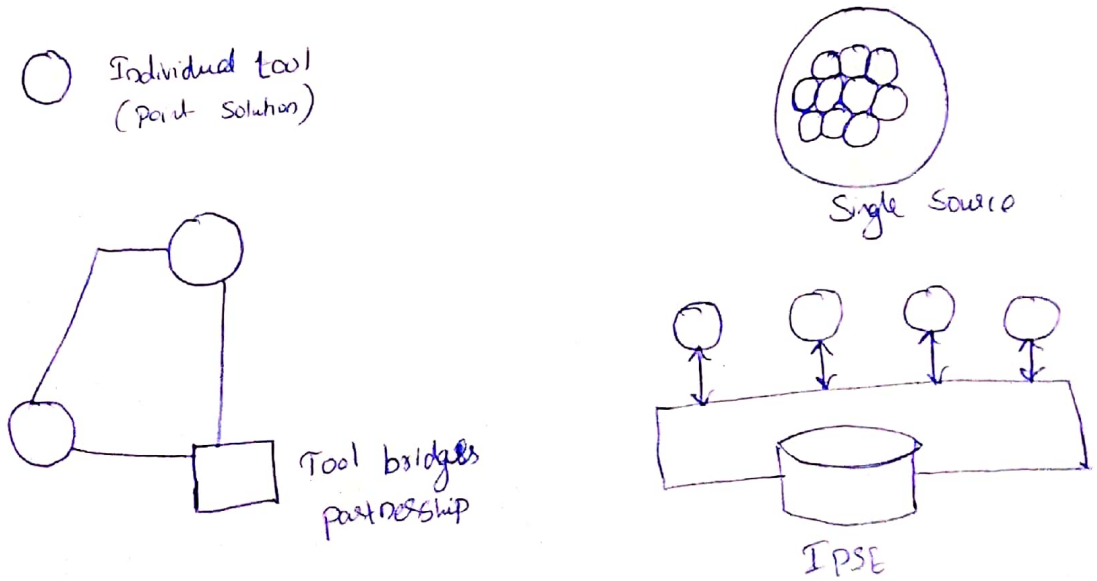
The environment architecture, composed of hardware platform & system support, lays the ground work for CASE. But the CASE environment itself demands other building blocks.

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

The integration framework is a collection of specialized program that enables individual CASE tools to communicate with one another, to create a project database, & to exhibit the same look & feel to the end-users..

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

Point-solution CASE tools can provide substantial individual benifit, but a software team needs tools that talk to one another. Integrated tools help the team develop, organize & control work products.



At the low end of the integration spectrum is the individual (point solution) tool. When individual tools provide facilities for data exchange, the integration level is improved slightly. Such tools produce outputs in standard format that should be compatible with other tools that can read the format. In some cases, the builders of complimentary CASE tools work together to form a bridge between the tools.

At high end of the integration spectrum is the integrated project support environment (IPSE). CASE tool vendors use IPSE standards to build tools that will be compatible with the IPSE & therefore compatible with one another.

TAXONOMY OF CASE TOOLS

Case tools can be classified by function, by their role as instrument 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 as a primary criterion.

→ Business process engineering tools: represents business data objects, their relationships & flow of the data objects between company business areas.

→ Process modeling & management tools: represents key elements of process and provide links to other tools that provide support to defined process activities.

→ Documentation tools:- provide opportunities for improved productivity by reducing the amount of time needed to produce work products.

→ Requirement tracing tools:- provides systematic database like approach to tracking requirement status begining with specification.

→ Project planning tools:- used for cost & effort estimation & project scheduling.

→ System software tools:- network system software, object management services, distributed component support, & communication software.

→ Database management tools:- RDMS & OODMS serve as the foundation for the establishment of CASE repository.

→ Software configuration management tools:- uses the CASE repository to assist with all SCM tasks.

→ Analysis & design tools:- enable the software engineer to create analyze & design models of the system to be built.

→ Interface design & development tools:- toolkits of interface components often past environment with a GUI to allow rapid prototyping of user interface designs.

→ Prototyping tools:- enabled rapid definition of screen layouts, data design & report generation.

→ Web development tools:- assist with the generation of web pages text, graphics, forms, scripts, applets, etc. ...

→ Static analysis tools:- code-based testing tools, specialized testing languages, requirement based testing tools.

→ Test management tools:- coordinate regression testing, compare actual & expected output, conduct batch testing & serve as generic test drivers.

→ Client server testing tools:- exercise the GUI & network communication requirement for the client & server.

Reengineering tools :-

→ Reverse engineering to specification tools - generate analysis & design models from source code.

→ Code restructuring & analysis tools - analyze program syntax, generate control flow graph, & automatically generate a structured program.

→ On-line system reengineering tools - used to modify online DBMS.

INTEGRATED CASE ENVIRONMENT

Integration of variety of tools & information that enables closure of ~~communication~~ communication among tools, between people & across the software process.

The benefits of integrated CASE (I-CASE) include

(1) Smooth ~~information~~ transfer of information from one tool to another and one software engineering step to the next.

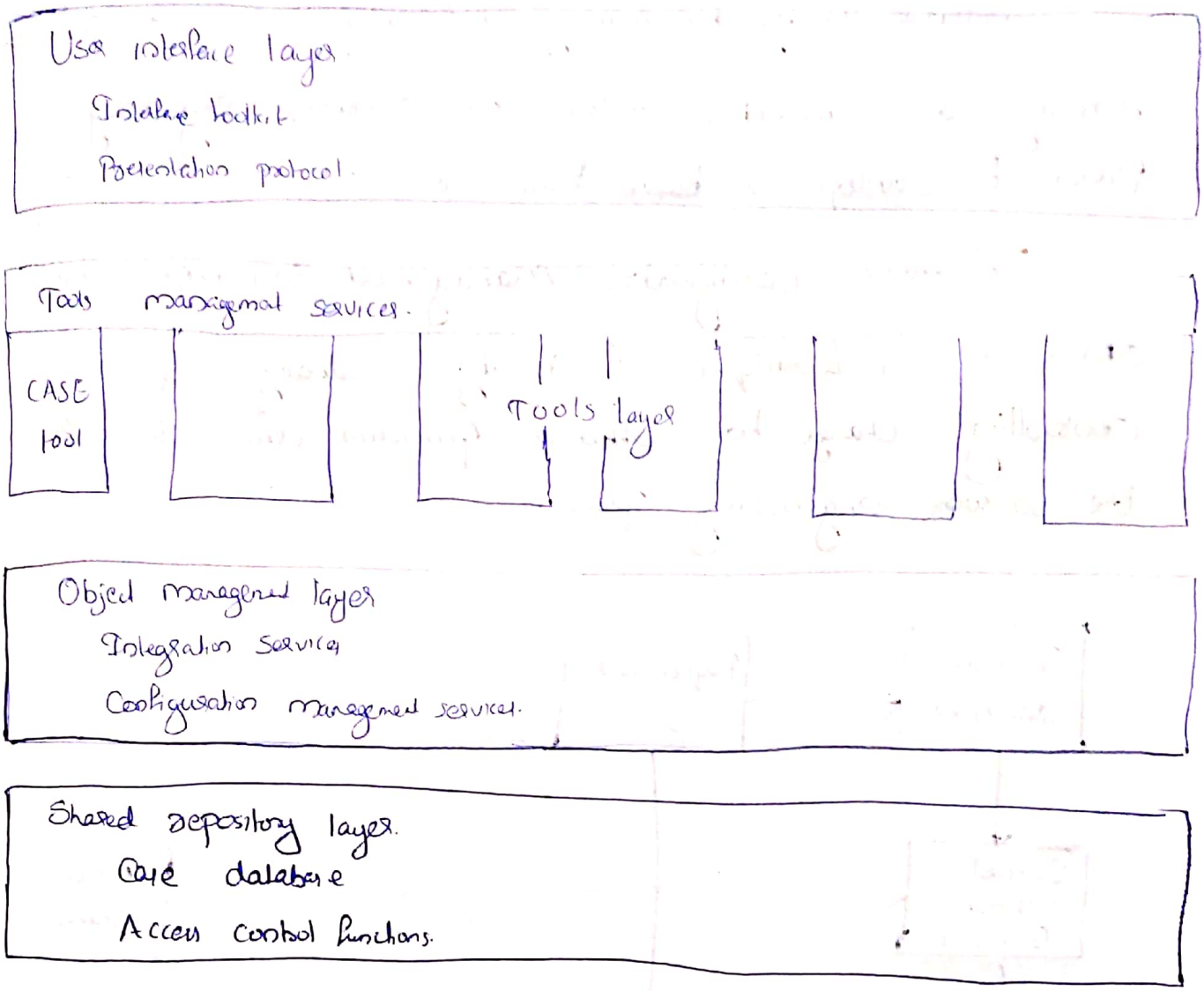
(2) A reduction in the effort to perform umbrella activities such as software configuration management, quality assurance & document production;

(3) An increase in project control that is achieved through better planning, monitoring & communication ; →

(4) Improved coordination among staff members who are working on a large software project.

But I-CASE also poses significant challenges. Integration demands consistent representations of software engineering information, standardized interfaces between tools, a homogeneous mechanism for communication between the software engineer & each tool & effective approach that will enable I-CASE to move among various hardware platform & operating systems.

Integration Framework Diagram



A software engineering team uses CASE tools, corresponding methods & a process framework to create a pool of software engineering information. The integration framework facilitates transfer of information into & out of the pool.

SOFTWARE CONFIGURATION MANAGEMENT

Software Configuration Management (SCM) is a set of management disciplines within the software engineering process to develop a ~~baste~~ base line.

Software configuration Management encompass the disciplines & techniques of initiating, evaluating & controlling change to software product during & after the software engineering process.