

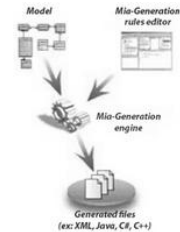
Computer-Aided Software Engineering

M8034 @ Peter Lo 2006

1

Definition of CASE

- Computer-Aided Software Engineering (CASE) can be defined as the disciplined and structured engineering approach to software and systems development.
- It emphasizes structured methods, with defined and standardized procedures.



M8034 @ Peter Lo 2006

2

CASE Tools

- CASE tools substantially reduce or eliminate many of the design and development problems inherent in medium to large software projects.
- CASE tools allow the software designer to focus on the systems architecture rather than on the actual implementation.

M8034 @ Peter Lo 2006

3

Classification of CASE Tools

- Information Engineering Tools
- Project Planning Tools
- Risk Analysis Tools
- Requirements Tracing Tools
- Metrics and Management Tools
- Analysis and Design Tools
- Programming Tools

M8034 @ Peter Lo 2006

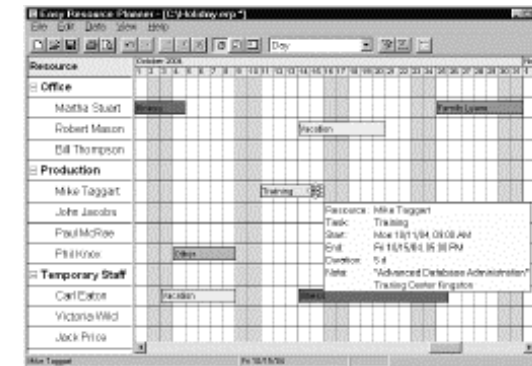
4

Information Engineering Tools

- **Information Engineering Tools** provide models from which specific information systems are derived.
- They typically represent business objects, their relationships, and how these data objects flow between different business areas within a company.

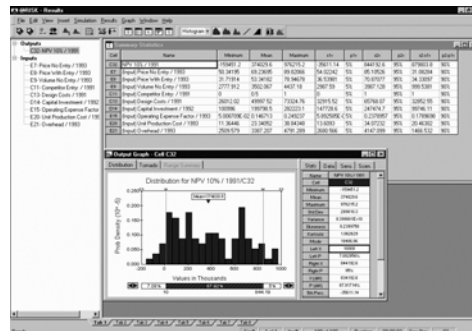
Project Planning Tools

- **Project Planning Tools** focuses on software project effort & cost estimation, and project scheduling.



Risk Analysis Tools

- **Risk Analysis Tools** identify potential risks and develop a plan to mitigate, monitor and manage them.



Requirements Tracing Tools

- **Requirements Tracing Tools** provide a systematic approach to the isolation of requirements, beginning with the customer Request for Proposal (RFP) or specification.

Metrics and Management Tools

- **Metrics and Management Tools** capture project-specific metrics (e.g. lines of code per person-month, defects per function point) and provide an overall indication of productivity or quality.



M8034 @ Peter Lo 2006

9

Analysis and Design Tools

- **Analysis and Design Tools** enable a software engineer to create models (e.g. representations of data, function and behavior) of the system to be built.

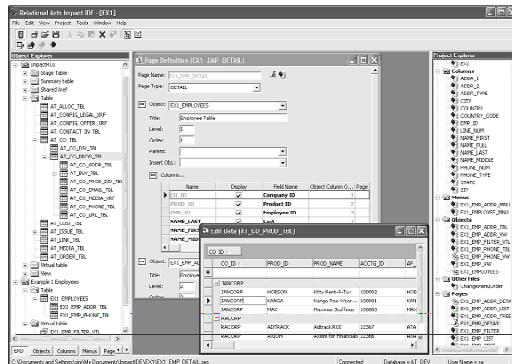


M8034 @ Peter Lo 2006

10

Programming Tools

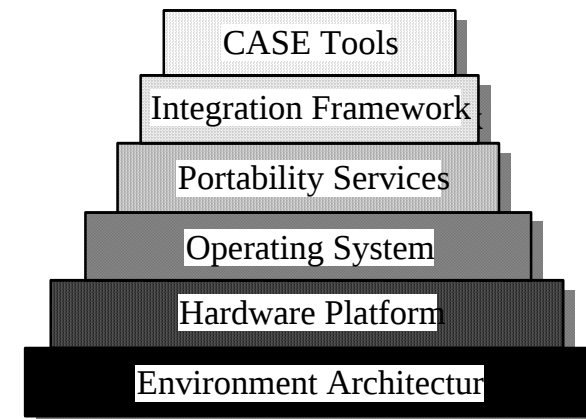
- **Programming Tools** typically encompass compilers, editors, and debuggers.



M8034 @ Peter Lo 2006

11

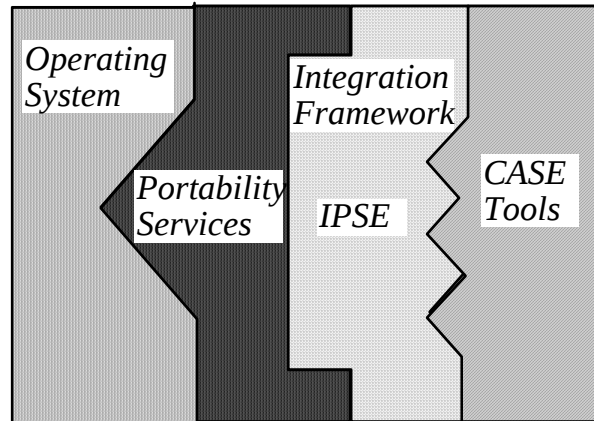
CASE Environment Model



M8034 @ Peter Lo 2006

12

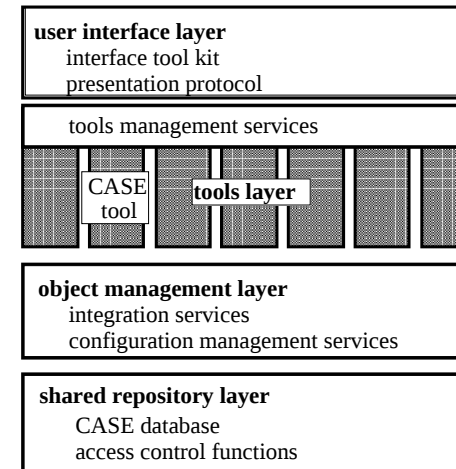
The Challenge: Putting it Together



M8034 @ Peter Lo 2006

13

An Integration Framework



M8034 @ Peter Lo 2006

14

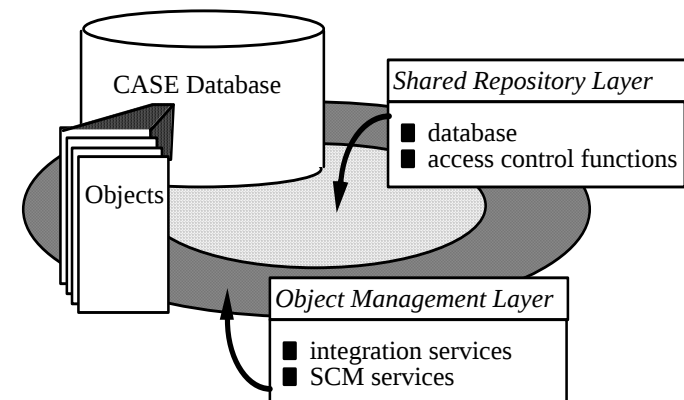
Integrated CASE environments

- The benefits of integrating CASE tools include:
 - ◆ Smooth transfer of information (models, programs, documents and data) from one tool to another;
 - ◆ A reduction in the effort required to perform activities such as software configuration arrangement, quality assurance, and document production;
 - ◆ An increase in project control that is achieved through better planning, monitoring, and communication;
 - ◆ Improved coordination among staff members who are working on a large software project.

M8034 @ Peter Lo 2006

15

Data Integration: CASE Repository



M8034 @ Peter Lo 2006

16

Potential Risk Areas of CASE

- Loss of creativity on the part of the software developer.
 - ◆ There is some argument that since software development is an occupation that requires skilled artistry, CASE- being tools that often follow strict rules for software development- may possibly stifle creativity.
- CASE is still relatively new.
 - ◆ Not all CASE tools are hence well integrated with each other or the software development process itself.

Factors in Consideration of CASE Tools

- Existing Company Standards and Methods
- Existing Computers and Future Computer Procurement
- The Class of Applications to be Developed
- Cost of the Package
- Ability to Customize Package

Existing Company Standards and Methods

- The CASE tool selected should support existing development methodologies in use, and should not introduce new standards or methods of development.

Existing Computers and Future Computer Procurement

- Unless the organization is willing to support the purchase of any additional equipment required to use the intended CASE tool, the CASE tool selected should ideally run on existing computers.

The Class of Applications to be Developed

- The CASE package selected supports the type of application that the organization normally develops.

Cost of the Package

- CASE tools do not come cheaply.
- Although the actual cost of CASE would depend on the particular CASE tool being acquired and the associated support platform, it has been estimated that organizations should expect to spend as much as US\$18,000 per person for a reasonable CASE package, with expenditure as high as US\$40,000 per person when support costs are factored in over 5 years.

Ability to Customize Package

- A CASE package that allows for custom modification to suit specific notation or symbols used in the current development methodology adopted would ideal.

A Taxonomy of CASE Tools

