Computer Aided Software Engineering Tools (CASE)

Tools of Software Development

2 types of tools used by software engineers:

1. Analytical tools
   - Stepwise refinement
   - Cost-benefit analysis
   - Software metrics
2. CASE tools

CASE:

- CASE stands for Computer Aided Software Engineering which is software that supports one or more software engineering activities within a software development process.
- improving capabilities, **functionality** and quality of software.

CASE TOOLS:

- Software that is used to support software process activities
- Provides software process support by
  - automating some process activities
  - providing information about the software being developed
- Almost all the phases of the software development life cycle are supported by them such as analysis; design, etc.

Example

CASE tools may support the following development steps for developing data base application:

- Creation of data flow and entity models
- Establishing a relationship between requirements and models
- Development of top-level design
- Development of functional and process description
- Development of test cases.

Why CASE tools are developed:

- Firstly Quick Installation.
- Time Saving by reducing coding and testing time.
- Enrich graphical techniques and data flow.
- Optimum use of available information.
- Enhanced analysis and design development.
- Create and manipulate documentation.
- Transfer the information between tools efficiently.
- The speed during the system development increased.

Categories of CASE Tools

- Tools
- Workbenches
- Environments

Categories of CASE Tools

- Tools
  - Support individual process tasks
  - Examples:
    - Checking the consistency of a design
    - Compiling a program
    - Comparing test results

CASE Tools

- Upper-CASE tools (front-end tools)
  - Assist developer during requirements, analysis, and design workflows or activities
- Lower-CASE tools (back-end tools)
  - Assist with implementation, testing, and maintenance workflows or activities
- Integrated CASE tools (I-CASE)
  - provide support for the full life cycle

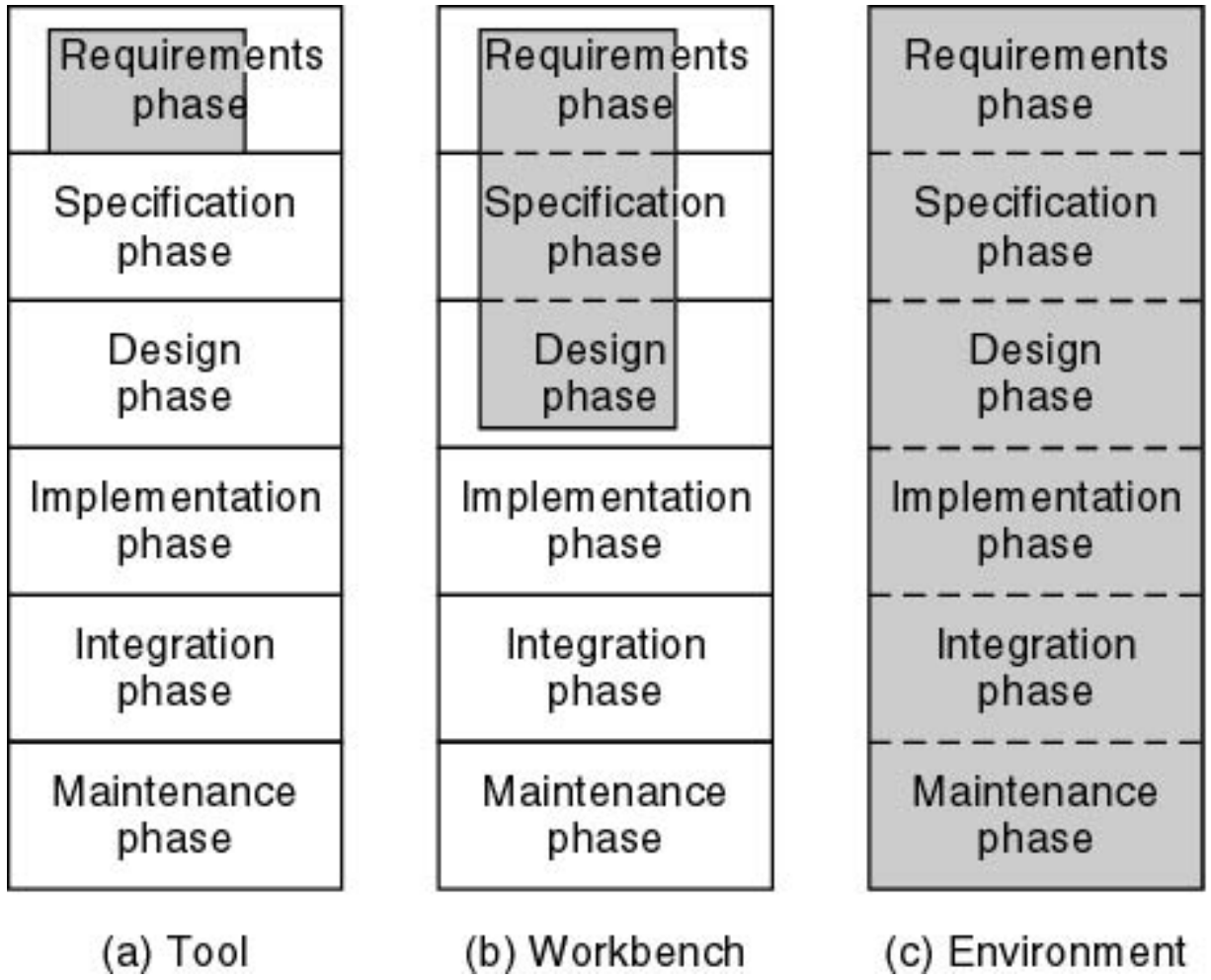
Categories of CASE Tools

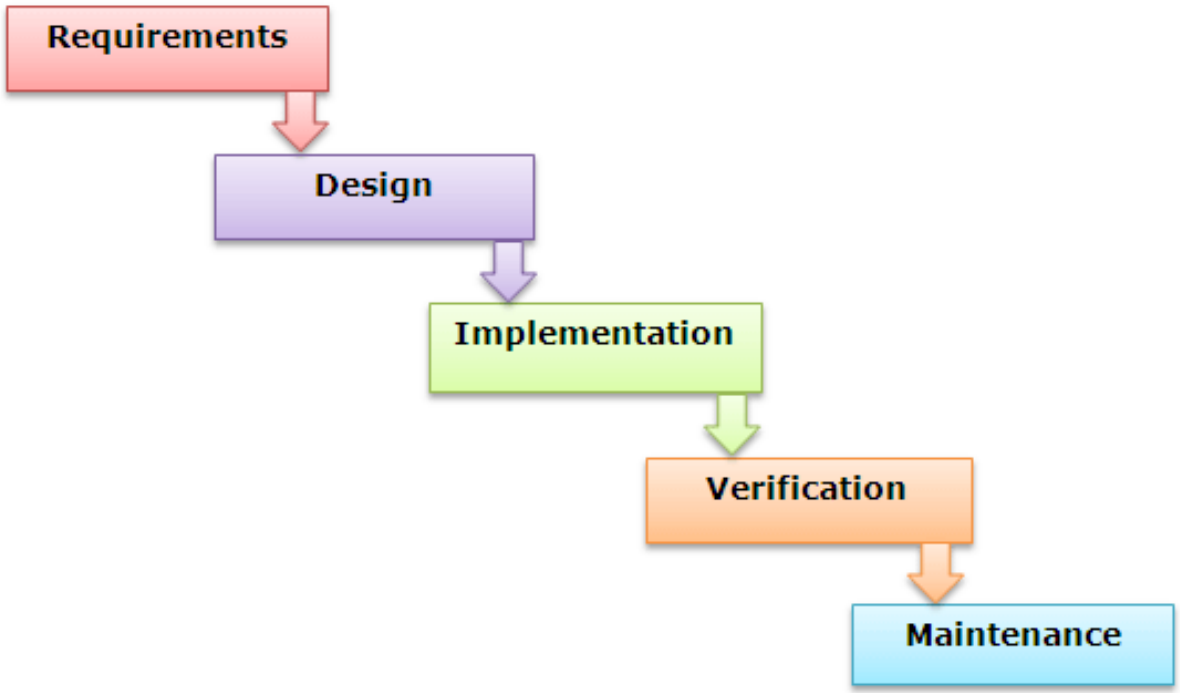
- Workbenches
  - Collection of tools that together support:
    - Process workflows (requirements, design, etc.)
    - One or two activities where an activity is a related collection of tasks
  - Commercial examples:
    - PowerBuilder
    - Software Through Pictures
    - Software Architect

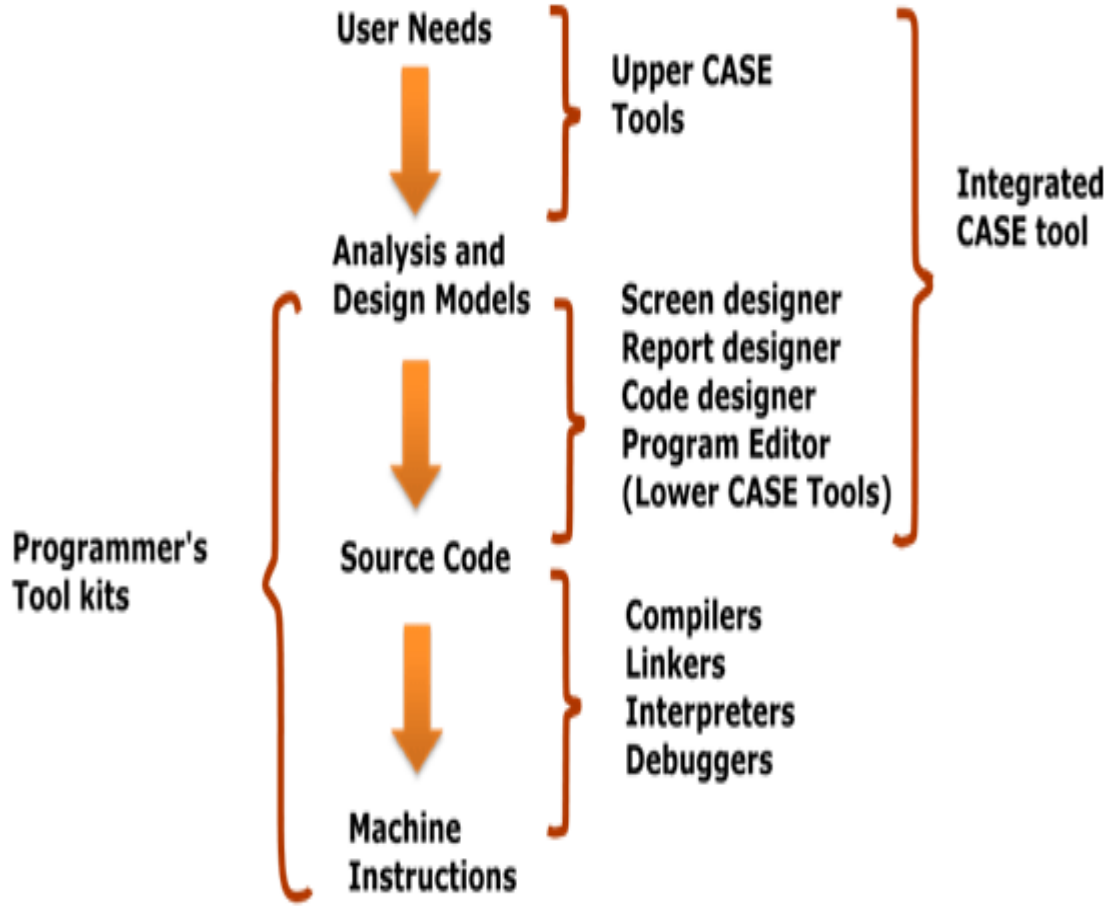
Categories of CASE Tools

- Environments
  - Support the complete software process or, at least, a large portion of the software process
  - Normally include several different workbenches which are integrated in some way

Taxonomy of CASE Tools

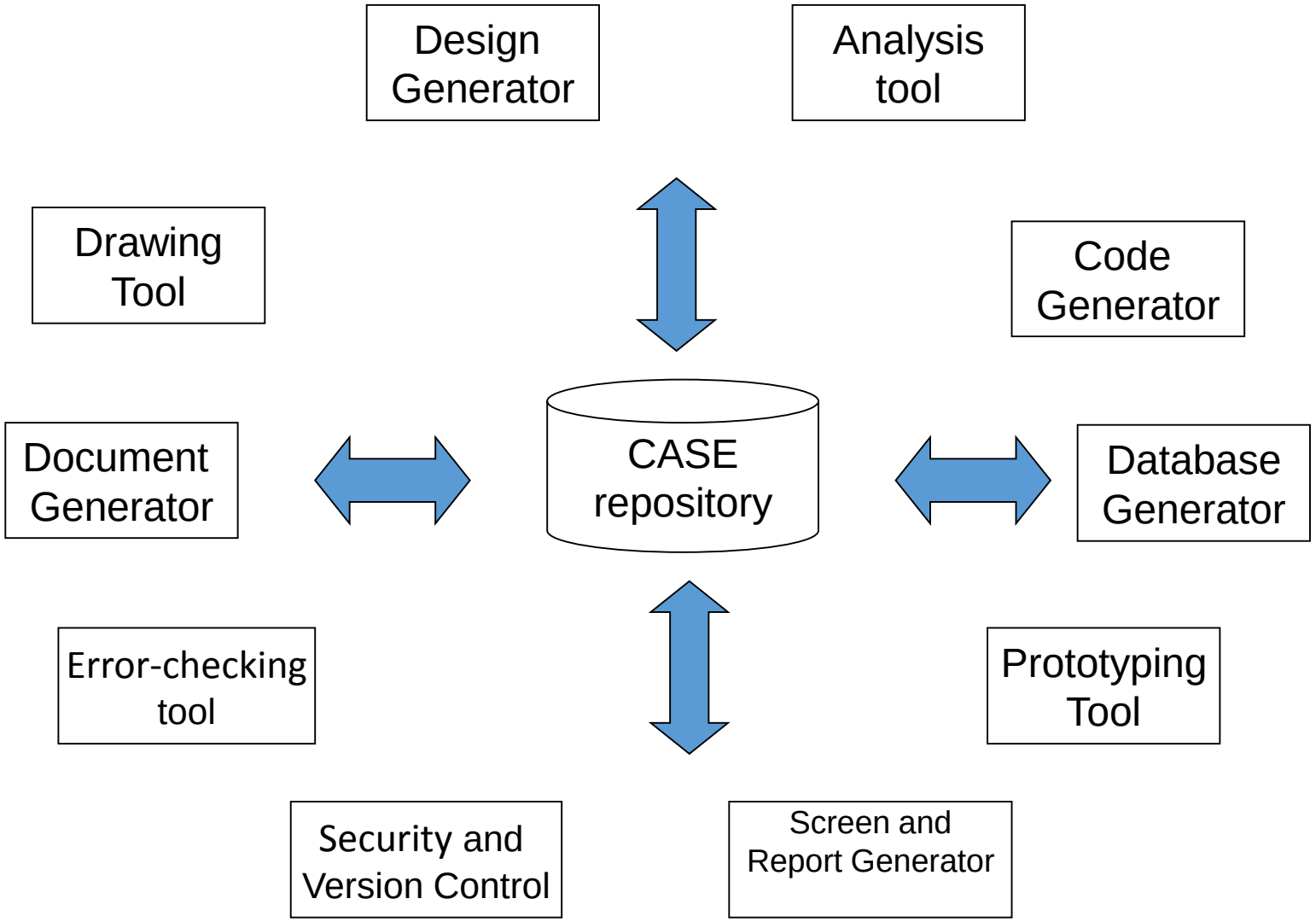






CASE Tool And Application Development

Components of CASE



Components of CASE

- CASE repository
  - Central component of any CASE tool
  - Also known as the information repository or data dictionary

Components of CASE

- CASE repository
  - Centralized database
  - Allows easy sharing of information between tools and SDLC activities
  - Used to store graphical diagrams and prototype forms and reports during analysis and design workflows
  - Provides wealth of information to project manager and allows control over project
  - Facilitates reusability

Components of CASE

- Diagramming tools
  - Allow you to represent a system and its components visually
  - Allows higher level processes to be easily decomposed
  - Can examine processes or data models at high or low level

Components of CASE

- Screen and report generators
  - Used to
    - Create, modify and test prototypes of computer displays and reports
    - Identify which data items to display or collect for each screen or report
  - Some tools have templates

Components of CASE

- Analysis tools
  - Generate reports that help identify possible inconsistencies, redundancies and omissions
  - Generally focus on
    - diagram completeness and consistency
    - data structures and usage

Components of CASE

- CASE documentation generator tools
  - Create standard reports based on contents of repository
  - Need textual descriptions of needs, solutions, trade-offs, diagrams of data and processes, prototype forms and reports, program specifications and user documentation
  - High-quality documentation leads to 80% reduction in system maintenance effort in comparison to average quality documentation

Basically, the CASE tools are used to

- Reduce the cost as they automate many repetitive manual tasks.
- Reduce development time of the project as they support standardization and avoid repetition and reuse.
- Develop better quality complex projects as they provide greater consistency and coordination.
- Create good quality documentation.
- Create systems that are maintainable because of proper control of configuration item that support traceability requirements.

Advantages and Disadvantages of CASE Tools:

Advantages	Disadvantages
Produce system with a longer effective operational life.	Produce initial system that is more expensive to build and maintain.
Produces system that more closely meet user needs and requirements.	Require more extensive and accurate definitions of user needs and requirements.
Produces system with excellent documentation.	May be difficult to customize.
Produces system that needs less systems support.	Require training of maintenance staff.
Produce more flexible system.	May be difficult to use with existing system.

- Thank You