

Name: Kaung Min Khant

Roll No: 3KaTha – 6

Module No: CS – 3109

Subject: Computer Aided Software Engineering

Assignment No: I

1. Define levels of CASE technologies and why do we need CASE tools for software development process.

Three different level of CASE technologies can be identified:

- (1) Production-process support technology This includes support for process activities such as specification, design, implementation, testing, and so on. These were among the earliest and consequently are the most mature CASE products.
- (2) Process management technology This includes tools to support process modelling and process management. These tools will call on production-process support tools to support specific process activities. A few products in this area are available but it is still the subject of considerable research.
- (3) Meta-CASE technology Meta-CASE tools are generators which are used to create production-process and process management support tools. Some meta-CASE tools are available but they are not easy to use and have not been widely adopted.

Historically, the most significant productivity increases in manufacturing or building processes have come about when human skills are augmented by powerful tools. One man and a bulldozer can probably shift more earth in a day than 50 men working with hand tools. Similarly, the productivity of engineering designers is improved when they are supported by CAD systems which take over tedious drawing chores and which check the design for errors and omissions. Automated tool supports for software engineers should therefore lead to improvements in software productivity. Therefore, we need CASE tools for software development process.

2. Describe three categories of CASE system. Why the first generations of CASE products has not led to the high level of productivity of improvements?

CASE systems should be classified in three categories:

- (1) Tools support individual process tasks such as checking the consistency of a design, compiling a program, comparing test results and so on. Tools may be general-purpose, stand-alone tools or may be grouped into workbenches.
- (2) Workbenches support process phases or activities such as specification, design and so on. They normally consist of a set of tools with some greater or lesser degree of integration.

(3) Environments support all or at least a substantial part of the software process. They will normally include several different workbenches which are integrated in some way.

The first generations of CASE products have not led to the high level of productivity improvements for the following reasons.

- (1) For large system development, the fundamental problems were accurately identified by Brooks. They are the problems of managing complexity both in the product being developed and in the development process. Brooks suggested that CASE technology can provide some support but cannot address their essential problem of complexity to any significant extent.
- (2) Current CASE products represent ‘islands of automation’ where various process activities are supported to a greater or lesser extent. Integration between these different products is limited. This limits the applicability of the technology.
- (3) Adopters of CASE technology sometimes underestimated the training and process adaptation costs which are essential for the effective introduction of CASE. They often skimmed on these costs with the consequence that the CASE technology was under-utilized.

3. Discuss different types of functional classifications of CASE tools with examples for each type.

Tool Types	Examples
Management tools	PERT tools, estimation tools
Editing tools	Text editors, diagram editors, word processors
Configuration management tools	Version management systems, change management systems
Prototyping tools	Very high-level languages, user interface generators
Method support tools	Design editors, data dictionaries, code generators
Language processing tools	Compilers, interpreters
Program analysis tools	Cross-reference generators, static analysers, dynamic analysers
Testing tools	Test data generators, file comparators
Debugging tools	Interactive debugging systems
Documentation tools	Page layout programs, image editors
Re-engineering tools	Cross-reference systems, program

restructuring systems