The good CASE

Gunnar Nilsson

CAP GEMINI

Cap Gemini BRA AB, Stockholm, Sweden

ABSTRACT

The good CASE has a result-oriented life-cycle, proper methods and supporting tools

CASE is not just a tool it is about new working principles. It is based on engineering principles and computer support. It includes graphical presentation and user cooperation.
In Cap Gemini we believe that the reason for CASE is to be found in todays and yesterdays problems in systems development.
Today our desktop computers have a power that would have been unimaginable just a few years ago.In software there is no similar trend.
There is an intense need for a clear improvement in price/performance concerning development of software.

This will put the following requirements on systems development:

- ❒ Increased productivity in systems development
- ❒ Increased quality in systems development
- ❒ Increased effectivity in systems development

And it requires a new way of working. How do we find the good CASE - the CASE that supports a good way of working - for systems development in the nineties!
Considering the rapid development in the CASE area it is today difficult to chose the right way.
Our first advice to a company considering CASE is. What is the problem? Is it about methods or tools? Does the problem concern a lack of methods for analysis or is it lack of programming productivity?

The CASE tools kan improve the effectivity in the working principles. But if there are no defined working principles the first thing to do is to establish that. CASE in itself does not provide defined working principles.

- ❒ Go for a good working principles.
- ❒ Remember that the development of CASE has but started.
- ❒ Look at the total environment.
- ❒ In Cap Gemini we will go for integrated working principles.

THE GOOD CASE

A result-oriented life-cycle, proper methods and supporting tools.

The good CASE

CASE is not just a tool. CASE is much more. CASE is about new working principles. It is based on engineering principles and computer support. It includes graphical presentation and user cooperation. This is the basis for this presentation.

Background

Why is CASE such a current topic? In Cap Gemini we believe that the reason for CASE is to be found in todays and yesterdays problems in systems development. Quite simply our inheritance.

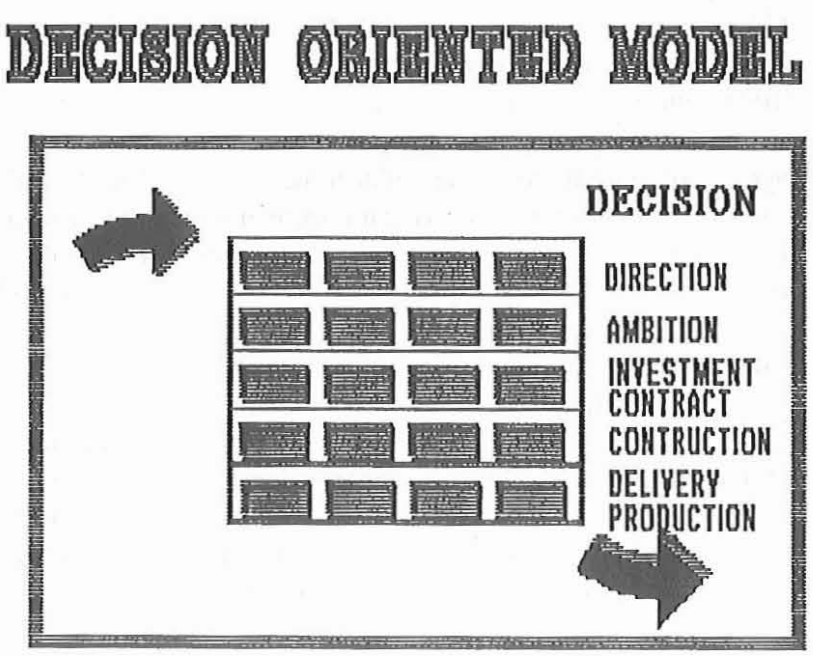
A few principal points:

The development of the price/performance ratio in hardware and software. On our own desks we can see the results of an enormous development in hardware. Today our desktop computers have a power that would have been unimaginable just a few years ago. If we compare the development of power and performance with the decreasing prices the development is even more fantastic. Today our money buys much more performance than what they did five or ten years ago.

In software there is no similar trend. Of course we are more efficient and productive but this trend is not at all as strong. In many cases it is hardly noticeable. The 4GLs are not commonly used and in many cases systems development is done today as it was ten years ago. There is an intense need for a clear improvement in price/performance concerning development of software.

– The development towards systems that support the business operations. We are more and more building systems that will improve the competitiveness of our customers. It is no longer rationalizations and reduced costs but gains in competitiveness that we are looking for. All systems are unique. This puts even greater demands on systems development. A delayed release could mean that the expected profit for the company never shows up - the competitor might already be ahead. We as systems developers must be able to guarantee quality in timely deliveries, cost and functionality. How can we do this? We must have competent staff and good working principles supported by good tools.

– Systems Maintenance is an ever increasing problem for many companies. An ever growing part of the resources that we have for systems development is occupied with maintaining older systems. This is all right if the maintenance produces good value for its money. But too often maintenance is seen only as a cost and with no payoff.
Cap Gemini has for some time developed a concept that focuses on profitability/usability as a steering mechanism for systems maintenance. This starting point is necessary but not sufficient. We must also create possibilities for reducing the differences between development and main-



tenance. The structure of a system must not deteriorate for every corrective action taken. We should try to reach an, in principal, unlimited technical lifecycle for the systems that we build. A good way of working and CASE tools are a prerequisite.

Requirements

All in all this will put the following requirements on systems development:

- ❒ Increased productivity in systems development
- ❒ Increased quality in systems development
- ❒ Increased effectivity in systems development

This requires a new way of working. How do we find the good CASE - the CASE that supports a good way of working - for systems development in the nineties! That is what we will spend the rest of this presentation talking about.

Working principles – for whom?

Who needs CASE? Who needs new working principles? Before we go into detail about what new working principles would look like we should stop to think about these issues.
Cap Gemini will during 89 make a major investment in a new way of working supported by CASE. The questions put forward earlier in this paper were the ones that we had to start with trying to answer.
Presently Cap Gemini BRA has 400 DP consultants. The new way of working is meant to be used firstly by our own staff but not only. Our 400 staff are working together with our customers staff. They are also invited to use our new way of working.
The new way of working is not only meant for the DP-Professionals.!!!

With a trend towards systems that support the business the group of people that are really knowledgeable about the company business operations is also a group that will benefit from our new way of working.

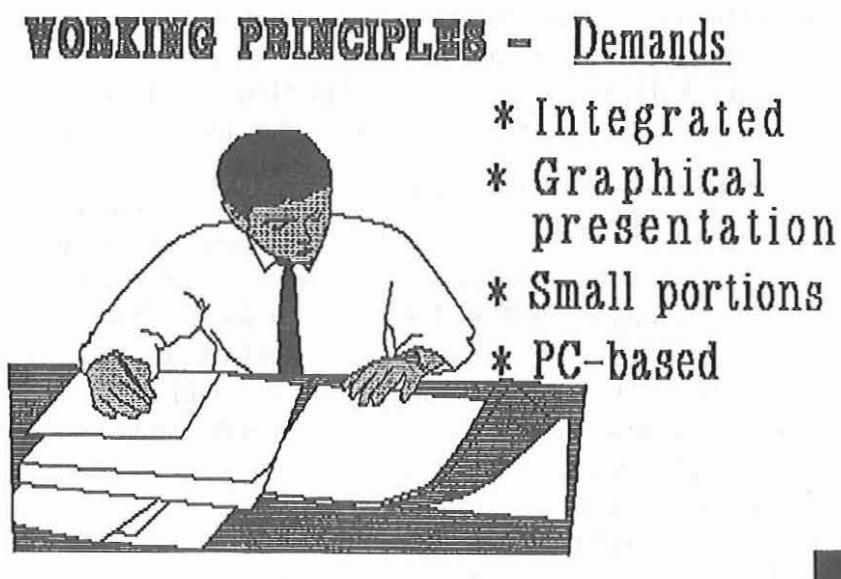
New working principles – why?

When we are working in our customers projects there are always a demand for new tools and methods. New tools and methods are needed to perform the work well, tools are needed to perform the work quickly and efficiently. A model for systems development is seldom asked for. Still a model for the systems development process is an absolute neccessity. It is the backbone that holds together methods and tools

Today there is no methods or tools that covers all aspects of systems development. What we can learn from the fact that methods and tools are in focus is that a model must not have a life of its own. The model must be a natural structure for what are the most important issues, methods and tools.

Requirements on a new way of working

We were just addressing that!
The way of working must be integrated. The development model, the methods and the tools must be united - even if they are separate and to some extent exchangeable.



– A graphic presentation interface.
We have learned from the first wave of CASE tools - a picture tells more than a thousand words. If pictures are good in a tool they are of course also good in descriptions of model and methods. We must create a way of working that is based on graphics not just tools.

A form to present the way of working in small portions.

Gone are the days of the thick binders. A new way of working must be delivered in small portions. A new way of working must be packaged for several purposes, depending of areas of use and demands from different users of it. This means a flexible model for development - we will come back to that later.

- PC-based.

The way of working must be supported by a PC. Almost all of us have PCs on our desks and we like to work on a PC. That way we feel more effective. Working principles for a PC - what is that if not a CASE tool?

Working principles – with computer support

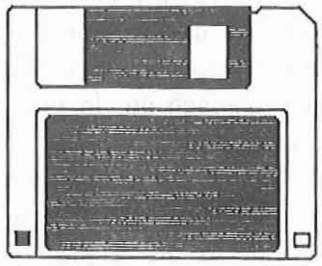
Coming back to my introduction - to Cap Gemini a CASE tool is not everything. Consequently it is not enough with just a CASE tool on PC, to say that the working principles are PC-based. That is why we have put Cap Geminis new development model on PC- with the help of presentation tools or ... as an integrated part of a CASE tool.

The working principles – the model- based on results

As mentioned earlier a well defined model for systems development is the basis for integrating methods and tools.

Cap Geminis model is based on results: We see systems development as a set of results that are to be produced. A result could in this context be a program description, a functional description, a data model etc.

The basis for our model is the description of these results.
What are the quality criteria for each result? What is the purpose of the result? Which methods and tools can be used to produce the result?



The DEVELOPMENT MODEL – available on PC

The results can be seen from different angles. Cap Geminis result model describes how the results interrelate in result chains and result areas. Let me give you an example. The resulting data base depends on the datya model in the result area of Storage.

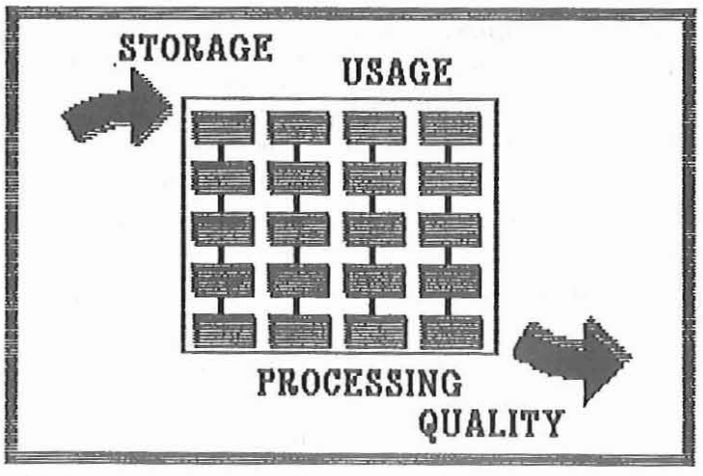
Cap Geminis decision model describes the results grouped in decision areas.
Earlier development models were often activity oriented. One of the drawbacks of that is that they are less flexible.
Now we look at the results rather than the activities. To reach a certain result the activities can vary- according to project or according to the method or tool used. Therefor the activities are handled in the project plan and not in the development model. To focus on results also makes quality assurance easier. (reviews etc).

The working principles – the methods

Some other important aspects of the methods that Cap Gemini uses:

- IRM
- Easy to learn
- Engaging
- Aiming for automatisation

The RESULT ORIENTED MODEL



- ❒ Reduces the difference between development and maintenance

In conclusion we like to call this way of working Information Engineering. This is not a method in its own right but rather requirements on the methods that we want to use.

The seminar is an important part in our use of methods. It fits well into project work because it provides: concentraton, quick results, a forum for creative debates, a natural opening for computer support, fast documentation of basic business operations and a common view for all those involved.

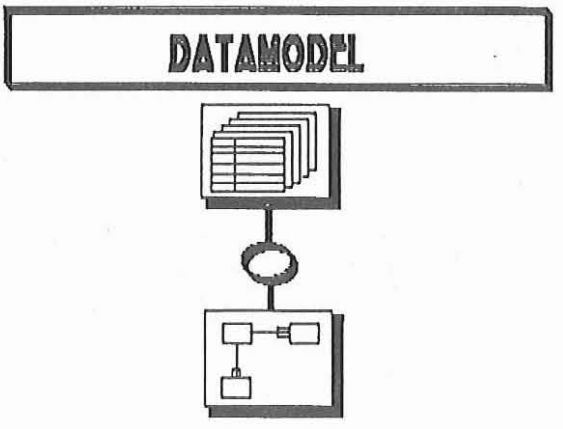
In Cap Geminis way of working we use the seminar in several circumstances, for instance for project start up, data modelling, function modelling, goal setting, project calculation, prototyping etc.
Project work is characterized by seminars and breathing spaces. In the same way the number of people engaged in project work varies from a large group during the seminar and smaller task forces in between. In this way of working a larger group can be better activated. The quality of the work improves and the organisation is more devoted to the work.

Is there enough experience built into the method of the CASE tool? Does the CASE tool cover all areas of methodology?

Unfortunately the total tool does not yet exist and it is doubtful if it ever will. Perhaps it is not even to be wished for.

We presently lack computer support in the following areas (there are however certain differences between suppliers):

- ❒ Security
- ❒ Quality Assurance
- ❒ Routine sketching
- ❒ Implementation
- ❒ Showing certain technical structures
- ❒ Distributed systems/data bases
- ❒ Prototyping (Prototyping in a Windows system environment)



Formal methods apart, but using methods is also to use earlier experience. Is it possible to incorporate our own experience into the CASE tool?

Of course it is possible. In Cap Gemini we have chosen to incorporate our methods into the tool. Through the orientation towards results in our model we can describe our methods directly in the tool, with an example throughout the model, with standards etc.

The working principles - tools

4GL, CASE, I-CASE

How could our business attain a faster development of our productivity in developing software?

The solution is often discussed in terms of 4GL, CASE and I-CASE. Are there any contradictions in these concepts? Could they be integrated?

4GL is firstly a tool for improving programming productivity. 4GLs does not mean that the quality of the systems that we develop improves. When programming productivity is the main problem 4GLs can be an answer.The problem with 4GLs is their lack of standardization, limited areas of application and lack of performance under top load.

CASE adresses the problems of software development from another angle, that is analysis. The main issue is to produce the RIGHT system. Not just to produce a system and fast. CASE has helped to open our eyes to a graphical way to presentation. With CASE it is, for the first time, possible to work standardized and methodological in large projects down to the last specification. With CASE redrawing is no major obstacle. With CASE it is no longer difficult to integrate several types of diagrams.

I-CASE (or Integrated CASE) is a further development of CASE. An integrated CASE tool creates

possibilities for combining the advantages of CASE and 4GLs. Effective analysis and effective programming.

However, I-CASE is just one more step on the way. There is still a long way to go before we have the total tool. Let us think for while what such a tool could be made up of:

functions as development tools:

- ❒ Above all a complete dictionary (repository)
- ❒ Simulation and prototyping facilities
- ❒ Description of logic and program- and data base generations
  (Methods tailored to the companys needs)
- ❒ Word processing
- ❒ Presentation tools

Functions for management

- ❒ Integrated with project management tools
- ❒ Integrated with systems management tools

End user functions

- ❒ Report generators
- ❒ Query languages

What can CASE give us today?

We are only in the beginning of a major development phase. But we can already today note several advantages with using CASE tools.

CASE for increased productivity - quality

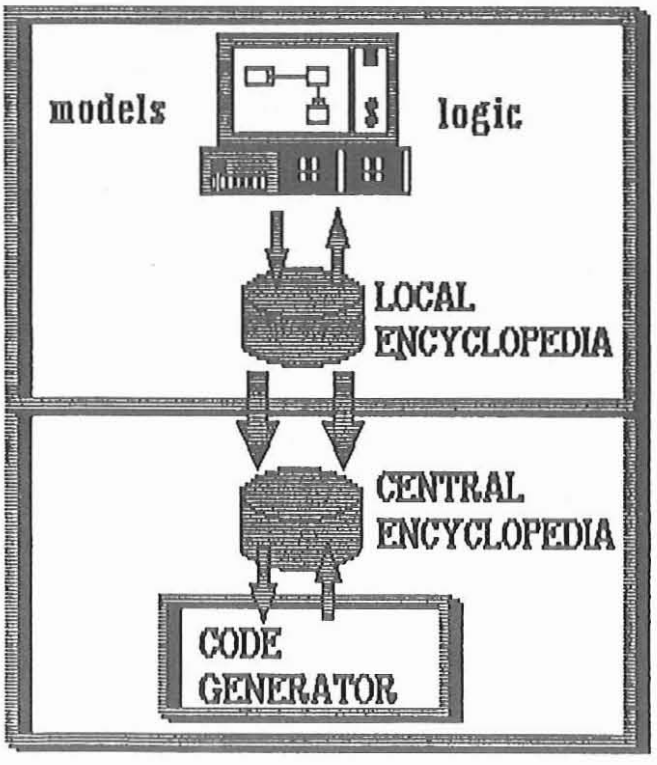
CASE can improve our productivity. We do not have to redraw diagrams. We can automatically generate data bases and screens. The tool can check for consistency in away that we could not have done on our own.

Sometimes the improvement in productivity is not to be seen in hard figures. The explanation for that is that the quality improves even more. Our customers have for the first time been able to make a complete analysis. This will show in better quality now and in the future, during maintenance.

CASE for better systems maintenance.

Maybe the biggest payoffs will occur in systems maintenance:

- ❒ By reducing the differencies between development and maintenance. The system structure does not deteriorate when programs are changed since everything is regenerated from the changed specifications.
- ❒ The number of deliveries during a systems development process are reduced. Now we can reach our results successively. The end user has greater possibilities of controlling the whole chain of events. This should give us better systems and better maintenance.
- ❒ In analyzing more thoroughly we will find more errors in an earlier stage and there will be less errors to contain in testing and in production. An error detected during maintenance costs many times more to correct than an error detected during analysis.

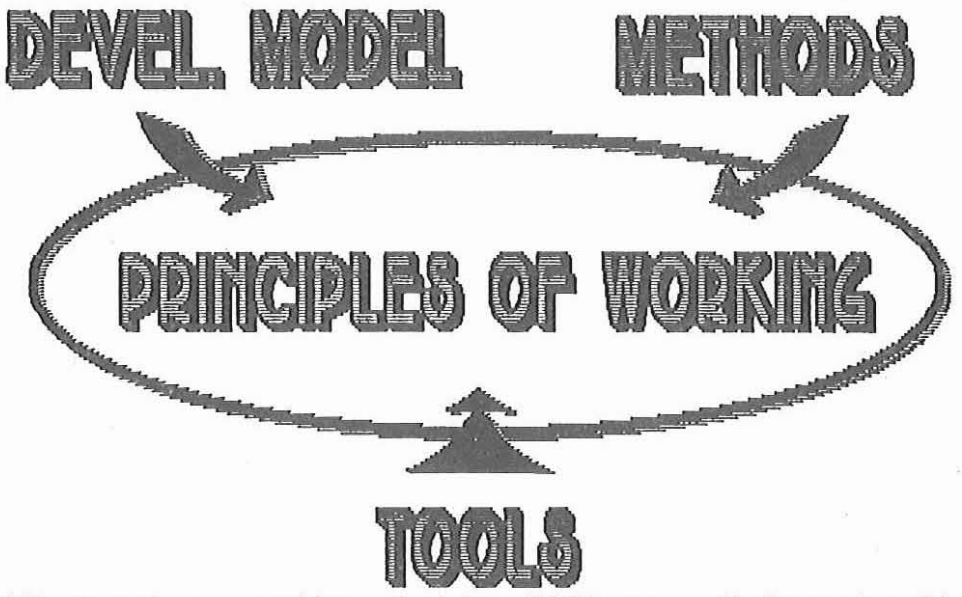


CASE

I-CASE

What should we do today?

Considering the rapid development in the CASE



area it is today difficult to chose the right way. The fact that the most influential hardware and software suppliers still have nlot shown their strategies and plans does make it easier. How should a company act? Is it better to wait for something even better? Or should one try to be in the front line? Is there other alternatives?

Our first advice to a company considering CASE is. What is the problem? Is it about methods or tools? Does the problem concern a lack of methods for analysis or is it lack of programming productivity?

The CASE tools kan improve the effectivity in the working principles. But if there are no defined working principles the first thing to do is to establish that. CASE in itself does not provide defined working principles.

Lack of effectivity and quality in the analysis is mostly due to problems with methods. There is a lack of methods, methods exists but are not used or methods exists only in the minds of experienced analysts. A CASE tool could work as a catalyzer. A vehicle for implementation of good methods and to for implementing good working principles. CASE can also fail - if it is not founded on a good way of working.

Lack of programming productivity does not neccessarily indicate a need for a complete I-CASE tool. Cap Gemini is right now performing tests where we compare quality and productivity when using a PC work station as opposed to conventional programming.
It is a question of putting the conventional programming environment on a PC in order to gain advantages.

If the task comprises the whole chain from analysis to programming I-CASE seems to be the choice. But that means not considering the costs of it. On top of that I-CASE is a rather untried concept. Who wants to pay for the development that still has to be done? Considering the large investment this group of products is the most sensitive to development. Will the products available today still exist tomorrow? Integrated CASE on PC, what role will it play in the future?

Many questions? Fewer answers? But...

- Go for good working principles. Try to find the strengths and the weaknesses of todays principles of working. Go for the profitable solutions. Making the largest investment does not have to mean reaping the largest profit.
- Remember that the development of CASE has but started. There will be ever more new products and ever more new releases. Waiting is probably not a good strategi but neither is going for one single solution.
- Look at the total environment. What dictionary does the company have? How does it work? Is the dictionary an asset or a burden? Do not let CASE be a bastard in the technical environment!
- What effects will IBMs proposal for a new CASE dictionary have?

In Cap Gemini we will go for integrated working principles:

- ❒ MODELL
- ❒ METHOD
- ❒ TOOL

And do not forget that the working principles is to be used by the individual in the development projects!