

How to Combine Tools and Methods in Practice — a field study

Kari Smolander, Veli-Pekka Tahvanainen, Kalle Lyytinen

Project **Syti**
University of Jyväskylä
Dept. of Computer Science
Seminaarinkatu 15
SF-40100 JYVÄSKYLÄ
FINLAND
email: syti@jytko.jyu.fi,
syti@finjyu.bitnet

Abstract

In spring 1989 we surveyed the experiences of some Finnish companies in methodology modelling (*metamodelling*) and adaptation of tools and methodologies to each other (*methodology adaptation*). The companies represented software production, banking, wood and metal industry, and wholesale trade. The study was carried out as a field study where we interviewed method developers, systems analysts and their supervisors. The goal of the survey was to find out whether there was need for metamodelling or methodology adaptation in general and how this need had been satisfied. The study shows that a little experience had been gained in adapting data dictionaries to methodologies but no such attempts had been made with CASE tools. One reason for this was that few methodological guidelines were extensively employed and supported in organisations. In general no systematic approach had been followed in the adaptation. We also explored possible causes for encountered difficulties and conditions for successful metamodelling and methodology adaptation. A crucial success factor was the adaptability and ease of use of the tool. In general the causes for the success are similar to those of information systems development in general. These include sufficient resources and management support, concrete benefits for those who do the actual work and friendly user interfaces.

1. Introduction

As the field of CASE matures more research is needed in focusing on problems in using CASE technologies. So far, this has been little researched area, though the literature on this topic is continually growing (see *e.g.* Le Quesne 1988, McDaniel 1989, I/S Analyzer 1989, Kemerer 1989, Orlikowski 1988)

Most of the studies have focused on general attitudes and opinions of CASE technology, productivity changes, or changes in systems analysts' skills and work patterns. So far there are no empirical studies on how matching of tools to methods (or vice versa) is done and what the outcomes of such attempts are. In the theoretical literature, there are, however, several solution strategies for the problem of matching CASE technology with the existing development practices and tools (*cf.* Bubenko 1988). What are the outcomes of and contingencies in the use of various strategies in practice is still largely unexplored.

The goal of this paper is to report the results of a field study where we examined how metamodelling and methodology adaptation had been tried out in some large Finnish IS departments. The major focus of the study was to gather the experiences of the organisations of methodology adaptation and metamodelling. We also attempted to shed light on how much activities are needed in the adaptation and how CASE technologies are changing this need.

The paper will proceed as follows. First we discuss the general setting of the study, explain the research methodology and define some key concepts applied in the field study. In sections 3 and 4 we describe how methodologies and software tools are being used and how they are valued. In section 5 we take a closer look at methodology adaptation, discuss reasons for it, and describe the methods used in the process and the process differences in studied organisations. Section 6 illuminates the necessary requirements for a successful adaptation based on our observations. Lastly, we briefly summarise the results of the field study and compare them with related research outcomes.

2. Research methodology

The impact of CASE technologies is little researched topic. In largely unexamined fields of study, it is usual to start with case-studies to gain understanding and to generate a valid research hypothesis (Benbasat *et al.* 1987). Therefore we saw it appropriate to begin our research with a qualitative field study. During this study we interviewed seventeen method specialists and IS managers in eight Finnish companies. In four organisations we interviewed one person, in three organisations three persons, and in one organisation four persons.

The companies were (in Finnish scale) large and they had highly developed and well-staffed IS departments. All companies were interested in CASE technologies and almost all had some experience of it. We divided the organisations into three major groups: "Software production" (two companies), "Banking" (three companies) and "Trade and industry" (three companies). In one software house we visited two fairly independent departments so there are actually nine *quasi*-organisations in the study.

The interviews took place in the late January and early February 1989. The interviews were carried out by two interviewers which helped to keep a relaxed and open atmosphere during the conversations as the other interviewer could concentrate on the conversation and the other on keeping track of the dialogue. No tape-recorder was used. The interviews were semistructured and they followed the interview format exhibited in the appendix. The results of the interviews were discussed with the representatives of the companies after the interviews so they fairly well reflect the opinions expressed during the interviews.

We also followed the following strategy when we interviewed more than one representative in one firm. The general questions about the organisation (see the appendix) were asked from the group of interviewees whereas the specific questions about metamodelling and methodology adaptation were asked personally. This turned out to be a good practice, as we thereby received varied opinions and different examples from the interviewees.

The results of our study should not be generalised too strongly because of the relatively small number of organisations included. In this sense the research could be viewed as *hypothesis generation* research. However, as the organisations were usually relatively large and had well-developed IS departments, they probably were more mature to adopt new technologies such as CASE tools or data dictionaries. Therefore, the outcomes of our study may apply to organisations in general in the longer run.

With these caveats, we believe that the study uncovers some interesting factors that affect the success or failure of CASE tools, data dictionaries and methodology adaptation approaches.

3. Conceptual preliminaries

As the empirical study focussed on metamodelling and methodology adaptation it is necessary to define these terms (and a host of other terms) in the way they were employed throughout the study.

Methodologies are conceptually distinct from methods which are generally defined as:

Explicit prescriptions for achieving an activity or set of activities required by the life cycle

model used to develop a software product. (Charette 1986 p. 64.)

Well-known examples of methods are Structured Analysis (De Marco 1978), ER-modelling (Chen 1976) and project management methods. A method embodies a conceptual structure which defines what the method is about. The conceptual structures of the most widely known methods include such concepts as *activity*, *entity*, *relationship* etc. The process of modelling the what aspect of methods deals with describing their conceptual structure in some notation. This involves using another conceptual structure (and method) to describe another method *i.e.* “going outside” the method domain proper (shifting to “metalevel”).

A methodology, in turn, is defined in the context of this study as an ordered collection of methods as suggested by a framework *e.g.* a life cycle model¹.

By *metamodelling* we mean in this paper a process of describing (modelling) a methodology *i.e.*, its methods and their interdependencies in order to implement the methodology in a CASE tool, data dictionary or the like.

A *tool* is defined here as an instrument for the execution of some procedure. More specifically, a tool incorporates software with which a IS design task can be done. We are mainly concerned in the following with CASE tools as well as to some extent with data dictionaries.

Methodology adaptation can be defined as the adjustment of a given IS design methodology and an IS design tool (*e.g.* a data dictionary or a CASE tool) to each other. (See also Lyttinen, Smolander, Tahvanainen 1989.)

This happens in *de novo* situations where they have not been matched with each other before. In such a case some modification has to be made in order to get the most out of the methodology and the tool. A tool can be adapted *e.g.* by changing its inherent data model. A methodology can be “tuned” for example by changing the syntax and/or semantics of its description language.

4. Methodical practices in organisations

Many companies visited had no methodology at all (as defined above). Instead, many companies had a sort of development *framework*, dividing the IS development into logical

¹ In general there is very much confusion about the use of terms such as “method”, “methodology” and “technique”.

and/or temporal phases (a phasing structure). Usually these frameworks did not suggest any particular methods but left the phases vaguely defined and loosely adaptable. In this sense, all organisations had developed (or transferred) some systematic practices, denoted here as “methodical practices” how the systems work should proceed.

In general, the attitude towards methods varied largely. In some companies they were defined and documented, but their use was voluntary. In some others method use was in principle obligatory and to some extent controlled.

The only widely used method was a species from the conceptual modelling fauna which usually included a locally modified ER-notation. Conceptual modelling was mentioned in six of the nine organisations covered. This was mostly used in the database design.

Some formalised phase models had been developed in connection with the data dictionaries. The most common of them was some variation of a Finnish design methodology TISMA which is actually a loose framework.

No other method nor methodology was widely used. The following were mentioned once during interviews: MBI, JSP, SA/SD and (an adaptation of) ISAC. Thus, only few organisations applied methods to model processes (as opposed to modelling data structures).

Despite the small number of methods in use, the methods were nevertheless considered useful. The general attitude was that the prevailing practice of letting every analyst to do things in his/her own way was yet more harmful and should therefore be abolished. The interviewees expressed that all organisations were striving a strategy towards a broader coverage of methods in the development work. As most organisations aimed at the use of CASE tools, it was also widely accepted, that a wider understanding and use of methods was needed.

Tables 1a and 1b summarise opinions about the benefits and drawbacks of the methods. In table 1a one can see that methods were not considered to improve software quality. Rather, their value flows from better common understanding of design options and problems. A majority of the interviewees mentioned the standardisation to improve joint understanding. On the other hand, only four interviewees mentioned better quality as an outcome of the method use. Surprisingly, no interviewee mentioned increased productivity though better understanding could be expected to result in better productivity. Information systems that meet the requirements can be expected to be produced more effectively and with less errors.

The biggest mentioned drawbacks of the methods were their rigidity and complexity (table 1b). The most frequent drawback was the inflexibility of methods: they compell the

Benefits of methods use	
+ Enhanced standardisation of documents and systems work and therefore easier communication in working groups.	16
+ Methods make systems work easier and faster.	7
+ Better quality of produced applications.	4
+ Methods structure the systems work and make project management more easy.	4
+ Enhanced maintainability of applications.	4
+ Less dependency from key persons.	3
+ Large databases are easier to construct.	2
+ Testing is more easy.	1
+ Naming problems are easier.	1

Table 1a. Noted benefits of methods use

Drawbacks of methods use	
- A strict procedure is enforced, that is unsuitable for some purposes.	7
- Methods mean more work and more bureaucracy and slow down the actual development work.	6
- Methods are often complicated and difficult to learn. Training for them takes time and costs money.	5
- Work load in the first phases of systems work increases. The benefits are seen only later.	3
- The maintenance of the descriptions is tedious.	1
- Methods are not mature yet.	1

Table 1b. Mentioned drawbacks of methods use

analysts to follow a rigid design procedure that is not necessarily fit to all situations. The next common drawback is controversial: in table 1a we have seven people claiming that methods ease systems development. In table 1b we see six interviewees claiming just the opposite: methods mean more work and slow down the development. One explanation for this controversial situation is the following: whether methods are seen to “fit” to a design situation depends largely on the analysts’ personal background, his/her education, attitudes, earlier experience etc. However, this variation was not explained during this study and is a research issue to be tackled in the future.

5. Tool support in organisations

In general the organisations hesitated to introduce CASE tools because the outcomes were still unknown and the investments considerable. Nowhere, however, the attitude towards CASE technologies was negative or the interviewees had no expectations at all. In addition, the interviewees expected that the use of CASE tools would grow rapidly in the next years. This was especially true in those organisations where a new methodology was being planned to be used side by side with the tool.

Every organisation had discussed the use of CASE tools and most had tried out some tool(s). In the majority of the organisations the use was still experimental. In no organisation was a CASE tool supporting the whole systems development life-cycle. In most cases the CASE tools applied were “front end” tools that supported analysis and design tasks. In one company the tool was also supporting information systems planning tasks.

In some companies a suitable methodology – tool combination had been found and a wider application was in the beginning. Only few companies had taken CASE tools in “production” yet, and even in those companies where the tool or tools had been put in the production line the experiences were preliminary.

In production, or just moving into introductory phase were the following tools: Excelerator®, IEF™, IEW®, Managerview and Teamwork. In addition at least Deft, CASE2000 and IDMS/Architect had been explored for use. The tool variation in organisations was considerable. Only IEW (three sites) and Excelerator (two sites) were used in more than one organisation. Oddly enough, the organisations seemed to have few methodological reasons for choosing these tools. Only in one place was IEW used in concordance with the Information Engineering methodology™ (Martin 1987).

Data dictionaries were less common. Only four organisations had a central repository. Worth noticing is that all the banks used a centralised data dictionary. The reason for this could be that banks have large centralised applications that can be managed with data dictionaries. The banks also had centralised IS departments in which the benefits of using data dictionaries are quite obvious. Data dictionaries in use were: Datamanager (2), IDD (1), and DDDS (1). Usually the data dictionaries covered a larger share of the system design tasks, but the emphasis with these tools was on technical design and programming tasks.

The usage pattern of data dictionaries seemed to evolve in two directions: first, they were to be introduced in new organisations (there were at least two sites that had plans for purchasing one). Second, new, more active tools were to be acquired in order to better support the maintenance of applications. Many interviewees considered very important that the data dictionaries should acquire their data automatically without requiring an extra task of

feeding data into the tool. One opinion emphasised contingencies in using data dictionaries: *i.e.* there should exist an option of not using the data dictionary. In some situations data dictionaries make the work only more rigid and do not allow for “bending the rules”.

Tables 2a and 2b summarise the recorded opinions about the benefits and drawbacks of the tools. In contrast to methods, tools were mostly seen to enhance quality. In addition, tools were expected to increase productivity, and ease maintainability. However, the major drawback of current tools was their low quality. Many respondents argued that tools only marginally support their primary functions and they are not yet a mature technology.

Benefits of tool use	
+ Better general manageability: less errors and enhanced quality.	11
+ Better productivity as descriptions come out easier.	8
+ Documents always up to date: easier maintenance and better changeability.	6
+ Naming easier.	4
+ Better quality documentation.	4
+ Documents and descriptions always in the same format.	2
+ Projects better managed.	2
+ The use of applications easier.	1

Table 2a. Tool use benefits mentioned

Drawbacks of tool use	
- Tool is of poor quality.	5
- Tool is costly, resource and time consuming.	4
- Tool requires much learning. May become too high a threshold for new users.	3
- Tool use means additional work.	3
- Maintenance of descriptions is tedious.	2
- Methods must be strictly followed. Great danger of making a mistake.	1
- Expectations often too high.	1

Table 2b. Tool use drawbacks mentioned

If we compare the benefits of methods and tools we can observe that increased quality was seen to flow almost exclusively from the use of tools. In principle, one could expect quality to increase through a disciplined use of a methodology which produces better results. This reveals an underlying belief in the interviewees' thinking: methods and methodologies are not used unless there are tools to support them. Another interesting *raison d'etre* is found in their thoughts of increased productivity: wider productivity issues (a working system for the right tasks in time) are not so much considered as limited efficiency *i.e.* the cost-efficient use of project resources. However, a disciplined use of methodologies could be expected to affect positively on the former whereas it might even work against the latter goal.

6. Methodology adaptation in the organisations

6.1. Reasons for methodology adaptation

Organisations usually face the adaptation problem when a new tool is purchased or when their methodology is changed. The new tool may not support the methods used, or the old methodology is found insufficient in some sense. An adaptation to prevailing methods had been attempted with all the the data dictionaries. On the other hand, no adaptation had been attempted with any CASE tool.

Table 3 tabulates the reported reasons for methodology adaptation. The most common reason for methodology adaptation is that a new tool is purchased. It is also common that the prevailing methodology is found insufficient or there are no widely used, standard methods in use. In some cases the tools may have technical shortcomings requiring adaptation *e.g.* when the interfaces are inappropriate.

Reasons for methodology adaptation	
• The new tool does not support the earlier methods or its description base schema is not fit for the descriptions.	8
• The previous methodology does not work well. A new methodology is needed.	6
• The interfaces or some other features of the tool do not fit.	3
• No previous coherent methodology.	3
• Both tools and methods are made in-house.	1
• The type of applications produced has changed.	1

Table 3. Reasons for methodology adaptation

6.2. Adaptation strategies

In the time of the interviews some methodology adaptation had been attempted in six companies. In some of them the work was still unfinished. In four cases out of the six the data dictionary had been adapted to the methods used, in the remaining two a methodology was modified to fit for CASE tools.

No formal metamodelling methods or strict procedures had been followed: the adaptation had been done more or less *ad hoc* using a trial and error procedure. In some cases there had been some attempts to describe the methods by themselves. This is quite natural for the conceptual modelling methods.

The most common strategy was to adopt the methods supported by the tool or pick up a tool that supported the methodology used. A special case was to construct a new methodology (mainly from existing parts) for the CASE tool. The organisations were reluctant to give up old methods if they were in widespread use. However, on many sites no standard methodologies were used, so adopting a new methodology was not a greater effort than introducing CASE tools and associated training.

From the interviews we can generalise five adaptation strategies².

- (1) *No adaptation*: the tool is used “as is” and only those parts that fit the prevailing methods are applied. This is a common strategy with CASE tools and also followed with data dictionaries when nothing else succeeds. (4 cases, banking, software production.)
- (2) *Purchase both the methodology and the tool and see that they fit for each other*. This was the most common way to adopt CASE tools. If the old methods are unsatisfactory or no methods are widely used, this may also be the most efficient strategy. (3 cases, trade and industry, software production.)
- (3) *Adapt the purchased tool to own methods*. This had been attempted with all data dictionaries. No CASE tools had been adapted in this way. (4 cases, banking, software production.)
- (4) *Standardise the ways in which to develop the descriptions according the prevailing methods with a CASE tool*. In other words, create translation rules for

²Several strategies had been used in some organisations with different tools or due to the failure of one strategy. Hence the counts do not seem to sum up correctly.

notations. This may be feasible if it is not possible to use one CASE tool in the organisation. Supposedly this strategy is appropriate for software houses that need to work with the customer's methods and tools. (One case, software production.)

- (5) *Build a tool, a methodology or both in-house.* The building of an own tool requires a large organisation for development and is feasible only in environments where standard tools are unsuitable. An own methodology may be developed more easily to support a CASE tool if its own built-in methodology is found inadequate. (2 cases, trade and industry, software production)³

It is to some extent arguable, how much "true" methodology adaptation had been done in all cases. Some borderline situations are the development of a methodology for CASE tools (strategy 5 above) and the adaptation of an existing methodology to the organisation (e.g. some local adaptations of a life-cycle model and observed local variations of ER-modelling). None of these examples fits very well for the concept of *methodology adaptation*.

Surprisingly, no organisation had adapted an existing methodology to a tool. Some steps in this direction had been made in some organisations: in one organisation more exact descriptions had had to be developed with the method due to the introduction of a tool. In one organisation a project was being started at the time the survey was done that aimed partially at this end.

6.3. Success of adaptation strategies

The interviewees had great difficulties in evaluating the success of the strategies they had followed at the time of the study. Strategy 1 seemed to work quite well in most cases, as it relies largely on the existing methodological base. However, it is problematic if the company does not have any sound methodological base or if it wants to renew and extend it.

At the time wider experience of adaptation was available only of the second and third strategy. With the third strategy the adaptation had either been unsuccessful or succeeded

³Bubenko (Bubenko 1988) has developed a similar type of classification for the introduction of CASE tools into the organisation. He does not focus on the methodology adaptation proper, and therefore there are obvious differences from our classification. In his classification two of the basic strategies are purchasing the tool and the methodology or making an own tool. These correspond to our strategies 2 and 5 above. Two other basic strategies mentioned in his article are the integration of several tools and the purchase of a customised CASE tool from some outside vendor. These could also be applied to the situation we were interested in. We did not, however, observe these strategies in our field study.

only partially. Though less experience was available of the second strategy the opinions were more positive as the interviewees reported of no larger disappointments. Strategies four and five were both in the preliminary phase and therefore their evaluation was not possible.

6.4. Differences in the adaptation strategies due to the industry sector

We further analysed whether the industry sector had any impact on the preferred adaptation strategy. We could not find any significant differences between banking and software production. Trade and industry differed greatly from the other two because they had not attempted to do any adaptation proper. On the other hand they had usually followed strategies 2 and 5.

This observation was somewhat surprising because we expected the software production group to do adaptation more readily than the two other groups, because of the primary role of systems development productivity in their competitiveness and their better skills in using tools. However, this hypothesis was not confirmed. It seems that the client and product oriented work to some extent **inhibits** the software houses to try adaptation. The software production companies seemed, however, to be better in adopting new methods than companies in other industry sectors.

These differences between banking and software houses and the trade and industry sectors can be explained by two factors. First, in banking and software production the IS departments are usually large (the average number of personnel in the studied companies was over 500 people) and the pressure to standardise development practices therefore greater *i.e.* these companies usually had developed and tried out some methodological principles. Second, these companies had usually specialised method development departments and therefore they had obtained more skills and knowledge to attempt methodology adaptation.

In banks and in software production we observed nearly all five strategies having been adopted. However, the industry sector did not seem to affect the success. For example in banks every organisation had done adaptation in one form or another. But the solutions and outcomes differed from one bank to another: one complete failure, one partial success and a project under way when the study was made.

6.5. How the tools and methods in use affect the methodology adaptation

We also explored how the methodology and tool used in the adaptation process affected the type and success of the strategy followed.

First, the failed adaptations (strategy 3) all concerned data dictionaries. According to the interviewees these failures were mostly caused by the shortcomings of the respective tools. The available data dictionaries were too inflexible to be adapted in any real sense. With CASE tools similar arguments could not be observed, because they had not been adapted but instead used “as is”.

Second, our study does not suggest any particular requirements for the methodologies to be adapted, since there was no data available of cases in which an existing methodology would have been adapted to some tool. In one project this was being tried out with a data dictionary at the time the interviews were made. Another borderline situation of method is the strategy 4 above, where the CASE tool use is standardised even though the tool is not used according to its native methods. These cases were still in the planning stage so there was no data available on their success.

Third, the available hardware platform has a crucial effect on the adaptation strategy. There is a dearth of data dictionaries for mainframes other than IBM and compatibles. Therefore, organisations had had to live up with programs that were not flexible enough for successful adaptation or even for normal use. The CASE tools used were excessively PC-based. One organisation had rejected a tool partially because it required more exotic (MacIntosh) hardware.

7. Requirements for successful methodology adaptation

As noticed above the success of the adaptation attempts varied considerably. Major difficulties and problems in methodology adaptation are depicted in table 4.

Difficulties and problems in methodology adaptation	
• The functions of the tool (user interface, reports) or its descriptive power lacking or too limited.	9
• The benefits for application builders negligible, the use of the tool is seen as an unnecessary burden.	3
• Technical difficulties, tool malfunctions.	3
• The adaptation project took too long.	1
• The adapted methodology inexact, not fitting for the tool.	1
• No support from the tool vendor.	1

Table 4. Difficulties and problems in the adaptation

From table 4 we can see, that the most common difficulties were shortcomings in or plain malfunctions of the tool. The next significant problem group was the lack of immediate benefits for the users. Many interviewees pointed out that there must be some benefit for every user — the adopted tool is not used unless it gives some concrete advantage.

A significant factor not explicitly mentioned in all cases is the lack of management support. A difficult task of methodology adaptation often involves major changes in how systems development is carried out. This can not be accomplished if sufficient resources are not available⁴. Often, however, the resources were insufficient which makes the analysts feel that their work is not seen as important. In these cases it is also easy for the analyst to blame the tool as the main culprit, even if the shortcomings could have been removed if more resources were given to the project. The lack of management support was also explicitly mentioned as a reason for unsuccessful adaptation. Also awkward and rigid user interfaces caused the adaptation to fail. Not surprisingly, all these reasons are widely known in the general literature concerning failures of information systems (Lyytinen & Hirschheim 1987).

We also asked about the requirements for the successful adaptation. The results of this question are summarised in table 5. The two most frequent requirements were: management support and training and user support (20 observations in together). Other necessary requirements are the ease of use of the tool and tangible benefits gained from its use (7 and 5 observations). Somewhat surprising is, how little importance is given to the functionality of the tools in comparison to table 4, where the shortcomings of the tool were seen as the major factor for failed adaptations. An explanation is, that the interviewees thought of CASE tools when they were asked for criteria for successful adaptation. All experiences of adaptation, in turn, they had about data dictionaries.

Requirements for methods were quite common (12 observations): they must be known both in theory and in practice and they must be both flexible and precisely defined.

8. Summary

The widespread use of tools was just beginning in large IS departments — especially in the case of CASE tools. A reason for this lies partially in the available tools and partially in the inertia of the organisations and in their reluctance to invest on large scale in the (still) costly CASE technology. More powerful and flexible CASE tools and CASE shells with better integration to other tools (repositories, document generators and 4GLs) will

⁴ For similar arguments regarding the use and usefulness of CASE see (Wilson 1989), where some crucial preconditions for the use of CASE are outlined.

Requirements for a successful adaptation	
• Management support is necessary: enough resources must be provided, company policies must support the adaptation.	10
• User support: enough training for new working practices, support must be there when needed, because it is difficult to estimate correctly the required work in projects that use new tools and methods, there should be some help for the task.	10
• The tools need to be easy to use and function as automatically as possible. There must be interfaces to development and documentation tools.	7
• Thorough knowledge of the methods and systems work is necessary.	6
• Methods must be flexible, but precise.	6
• The users have to accept the tools. Their use must give real benefits for everyone involved. Feedback from users must be taken into account.	5
• The tool has to be customisable.	4
• The tool has to be well understood.	2
• The aims must not be set too high.	1
• Methods discipline must be taken care of.	1

Table 5. Requirements for the success of the adaptation

heal some of these problems. Also when the price/performance ratio improves organisations will be more willing to large scale investments.

Our study shows that organisations express a rich array of ways to adopt CASE technologies and the adaptation process is often painful. Several failures were reported, but many organisations also had carried out the adaptation successfully.

However, there seems to be no general strategy that is good in all situations. Several factors, such as the size and the type of IS department, application portfolio, evolution of methodological practices and skills in the organisation, management commitment and support, expectations of CASE technology, available hardware platforms, and evolution of the CASE market itself affect the shaping and choosing of the strategy.

Organisations did not generally choose the strategy based on careful analysis and consideration. Instead, the strategies seemed to be more or less invented on the fly. Therefore organisations were quite flexible in changing their strategies if major obstacles should occur. We did not encounter any systematic methodological procedures followed in metamodelling and adaptation.

We could generalise five generic strategy options that organisations are likely to follow in introducing the CASE tools. Some of them were more common with data dictionaries which were also included into the study.

The study reveals that the success of the methodology adaptation is crucially dependent on the amount of resources available. If the work is seen as important and is given enough time and work, it is more likely to succeed. The introduction of CASE and the accompanying method discipline is a major change in the IS development which needs to be taken into account through proper training, cultural change and the like. The quality of the tools is also an important success factor: their reporting capabilities (graphic pictures as well as textual reports) and user interface must be advanced and customisable. The use of the tools must give some direct benefits to everyone involved, they should contain many automated functions and have a functional interface to programming tools, code generators and the like.

Most of the success factors we encountered are fairly well known from implementation studies of “ordinary” IS applications. This suggests that methodology adaptation and the introduction of design tools needs to be considered as a species of software development. Its success or failure depends to a large extent on the same factors as developing a “conventional” information system.

References

- . , “Building more flexible systems,” *I/S Analyzer* **27**(10) pp. 1-12 (October 1989).
- . Benbasat, Izak, Goldstein, David K., and Mead, Melissa, “The Case Research Strategy in Studies of Information Systems,” *MIS Quarterly* **11**(3) pp. 369-386 (September 1987).
- . Bubenko, jr., Janis A., *Selecting a strategy for computer-aided software engineering (CASE)*, SYSLAB University of Stockholm, Stockholm (June 1988).
- . Charette, Robert N., *Software Engineering Environments: Concepts and Technology*, Intertext/McGraw-Hill, New York (1986).
- . Chen, Peter Pin-Shan, “The entity-relationship model - toward a unified view of data,” *ACM Transactions on Database Systems* **1**(1) pp. 9-36 (March 1976).

- . De Marco, Tom, *Structured Analysis and System Specification*, Yourdon Press, New York (1978).
- . Kemerer, C.F., "An Agenda for Research in the Managerial Evaluation of Computer-Aided Software Engineering (CASE) Tool Impacts," in *Procs. of the 22nd Hawaii International Conference on System Sciences*, (January 1989).
- . LeQuesne, P. N., "Individual and Organisational Factors and the Design of IPSEs," *The Computer Journal* **31**(5) pp. 391-397 (1988).
- . Lyytinen, Kalle, Smolander, Kari, and Tahvanainen, Veli-Pekka, "Modelling CASE environments in Systems Development," in *Procs. of CASE89 The first Nordic Conference on Advanced Systems Engineering*, , Stockholm (1989).
- . Lyytinen, K. and Hirschheim, R. A., "Information System Failures: A Survey and Classification of Empirical Literature," *Oxford Surveys in Information Technology* **4** pp. 257-309 (1987).
- . Martin, James, *Information Engineering*, Savant Technical Report 1987.
- . McDaniel, Paul D., "Using PSL/PSA to Model Information System Planning for The United States Department of the Army Headquarters," in *Procs. of CASE89 The first Nordic Conference on Advanced Systems Engineering*, , Stockholm (1989).
- . Orlikowski, W. J., "CASE Tools and the IS Workplace: Some Findings from Empirical Research," in *Procs. of the 1988 ACM SIGCPR Conference on the Management of Information Systems Personnel*, (April 7-8, 1988).
- . Wilson, D. N., "CASE: guidelines for success," *Information and Software technology* **31**(7) pp. 346-350 (September 1989).

APPENDIX: Interview format

1. Background

These questions inquire the relationships between methodology adaptation and the size of the organisation, the industry sector, the educational background of its personnel etc.

It is possible to answer to these questions in advance. If some printed material is available, it will be very useful. We recommend that you collect the necessary information in advance.

1.1. The company

What is the industry sector the company is operating in?

The most important types of customers/most important customers?

1.2. The systems work in the company

By systems work we mean here the production of applications as a whole (at least from systems design to maintenance), not just system design or programming. By methods we mean predefined ways of producing, classifying, and describing design information, not merely drawing techniques or phase models.

The hardware and systems software in use?

The volume of IS work (number of personnel, percentage of company turnover?)

The education of the IS personnel?

The organisation of the IS work?

- As part of the company
- Internal organisation (units, their working areas)
- How is methods development organised?

The design methods in use? In-house changes in them?

How widely are methods used (percentage of personnel and projects)?

Is methods use mandatory? How is it supervised?

The benefits/drawbacks of methods use/not using methods?

The ratio of development/maintenance work?

How many projects are currently running?

Are there any CASE tools or data dictionaries in use/proposed for use? Which?

How are these linked to code generation?

If a data dictionary is used, how many objects/object types are currently defined in it? Number of inherent types/added synonyms/aliases?

What benefits/drawbacks have the tools had? Why?

Is coding made in-house or by some subcontractor?

How are the design documents passed over to programmers?

Some examples of current applications? How are applications managed?

How large are the applications (many years/lines of source code)? (<1, 1 to 5, 5 to 10, >10 many years; <10000, 10 000 to 50 000, 50 000 to 100 000, >100 000 lines)

What types of applications are in use? (Batch oriented, real time, interactive, other)

Does the type of application make difference in the methods used in design? How?

Are IS architectures made?

- By whom?
- With which methods?

1.3. The interviewee

The position in the organisation (title, work area)?

Educational background?

Experience in work?

Phone/email for checkings?

2. Methodology adaptation

Here we try to describe methodology adaptation in the company or the need for it. We are especially interested in the benefits and drawbacks of the situation and their causes.

By methodology adaptation we mean the matching of a IS design method and a tool (CASE tool, data dictionary). Methodology description means here that the method to be adapted is described with some (possibly some other) method and description technique.

2.1. The need for adaptation

Has the company experienced need for methodology adaptation?

When and why?

Has some adaptation been done?

If not and there would have been need to, why not? How else is the problem solved?

What other (if any) solutions were considered for the problem?

2.2. The making of the adaptation

What tool(s) and which method(s) was (were) involved?

Who did the work, who was responsible for it?

How was the adaptation done (tool, method, ad hoc...)?

How was the method described (E-R model, other)?

Did the adaptation succeed (not at all/badly/moderately/well/perfectly)?

How much resources did the work take?

Difficulties encountered?

Were changes in the method or in the tool needed? In which? What kinds of changes?

2.3. Benefits and drawbacks

What benefits and drawbacks followed from the adaptation/abandoning the adaptation? (In comparison to the other possibility)?

In which situations do the benefits/drawbacks occur?

What are the benefits/drawbacks due to (method, adaptation, compability of methods, tools, changes made, personnel, other)?

The conditions for a successful adaptation (training, prevailing knowledge, tools, methods, management support)

2.4. Other remarks

Other remarks on the topic?

Comments on the interview?