

## Scandinavian Journal of Information Systems

---

Volume 4 | Issue 1

Article 6

---

1992

# THREE BARRIERS FOR CONTINUING USE OF COMPUTER-BASED TOOLS IN INFORMATION SYSTEMS DEVELOPMENT: A grounded theory approach

Jan Pries-Heje

*The IT University of Copenhagen*, [jph@itu.dk](mailto:jph@itu.dk)

Julius Thomsens Plads

[JuliusThomsensPlads@emailaddressnotknown](mailto:JuliusThomsensPlads@emailaddressnotknown)

Follow this and additional works at: <http://aisel.aisnet.org/sjis>

---

### Recommended Citation

Pries-Heje, Jan and Plads, Julius Thomsens (1992) "THREE BARRIERS FOR CONTINUING USE OF COMPUTER-BASED TOOLS IN INFORMATION SYSTEMS DEVELOPMENT: A grounded theory approach," *Scandinavian Journal of Information Systems*: Vol. 4 : Iss. 1 , Article 6.

Available at: <http://aisel.aisnet.org/sjis/vol4/iss1/6>

This material is brought to you by the Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Scandinavian Journal of Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

**THREE BARRIERS FOR CONTINUING USE OF  
COMPUTER-BASED TOOLS IN  
INFORMATION SYSTEMS DEVELOPMENT**  
**A grounded theory approach**

JAN PRIES-HEJE

Institute of Computer and Systems Sciences at Copenhagen Business School  
Julius Thomsens Plads 10, DK-1925 Frederiksberg C, Denmark

**Abstract**

Qualitative information systems research such as field studies and interviews generates large numbers of unstructured and complex data. This paper describes the use of a grounded theory approach to the analysis and structuring of interviews from nine Danish firms concerning the use of computer-based tools in information systems development. The following grounded theory was discovered: Before a software developer starts using a computer-based tool continuously three barriers have to be overcome. Firstly the information on the tool, whether verbal or in writing must be evaluated positively. Secondly the proper situation for putting the tool into use must have arrived. And thirdly, when the tool has been put into use, it must fit when used to remain in continuing use. Throughout the paper emphasis is placed on the ability of grounded theory to facilitate the movement from initial unstructured interview transcripts through coding-memos and concepts to a theory and its implied hierarchy of categories.

*Keywords:* Computer-based tools, barriers for use, grounded theory approach, interview study.

## 1 Introduction

Every day new computer-based tools for software designers are created and marketed, but there is little research available to enlighten the development of computer-based tools and to tell why and how a tool is put to use. Neither do we understand the conditions under which a tool can be used again and again instead of gathering dust in a corner.

This paper presents an empirical investigation of why and how tools are put into use. The presentation is based on interviews conducted in several projects in 9 companies. The interviews were analyzed using the grounded theory approach, the result being a theory saying that there are three barriers to be surmounted before a tool is in continuing service, that is positive evaluation, proper situation and appropriateness when used.

The organization of the paper is as follows: Section 2 describes the research design and the methodological considerations behind it. Section 3 gives an account of the applied grounded theory approach for Data Analysis. The resulting grounded theory in the shape of a model containing three barriers for the continuing use of computer-based tools is presented and discussed in Section 4. Finally in Section 5 the grounded theory approach for information systems research is discussed together with related work.

## 2 Research Design

One can study the use of tools for software development at many levels in organizations. Bill Curtis has, for example, put forth a so-called layered behavioural model of software development (Curtis 1988), from which it appears that software development can be studied at five levels: Business milieu, company, project, team or individual. I decided to limit my project to examining the project and team levels.

The next problem was to establish contact with some relevant firms. About twenty firms were selected with a view to obtaining as differentiated a representation as possible. Nearly half of them agreed to participate in the project, with the proviso, however, that their time was a very scarce resource, and that they were not particularly glad to let me observe (parts of) the software development process, primarily for fear that my presence would influence the process in an unwanted direction. I chose, therefore, interviewing as my research method.

Because a study that uses only one method is more vulnerable to errors linked to that particular method I decided to combine different data collection methods in the same interview. In this way I hoped to be able to obtain some of the advantages of what (Patton 1990, p. 186) calls Methodological Triangulation, that is the use of multiple methods to study a single problem. In the concrete the interviews were divided into three parts. In the first part I used an interview guide approach (see an extract in Appendix 1) to investigate problems in software development and experiences with methods and tools. The second part of the

interviews was an open-ended question: Which methods/tools would you want to have and use? And in the third part of the interviews I collected reactions to and quantitative evaluations of a checklist of existing methods and a wide range of tools, see Appendix 2 and 3.

Pseudonym	Firm Type	Project type, size and status
Alfa	Electronic firm	Control of electronics. 60 persons. Implementation phase.
Bravo	Software house	Software for communication. 3 persons. Implementation phase.
Charlie	Software house	Industrial control. 10 persons. Maintenance phase.
Delta	Software house	Software Development tool. 30 persons. Design & implementation phase.
Echo	Co-operation among several software houses	Software development tool. 20 persons. Implementation phase.
Foxtrot	Machinery and electronic manu- facturer	Process control software. 15 persons. Maintenance phase.
Golf	Hardware manufacturer	Several projects.
Hotel	Consultant	Several projects.
India	University research	User interface management. 3 persons. Analysis & design phase.

Figure 1: *Participating firms and projects*

In Figure 1 is shown the nine projects and firms I conducted 19 interviews in, with an average duration of over 2 hours. Each interview was recorded, transcribed and validated by the interviewee. In the nine firms I tried to get in touch with an ongoing project to get data on the project which were as new and unprocessed as possible. In most cases I succeeded in interviewing, say, the leader and one or two team members of an ongoing project, but in some cases I had to make do with interviewing 1–2 persons with great experience from earlier projects.

### 3 Using the Grounded Theory Approach for Data Analysis

As a result of the interviews in the nine projects and firms I had several hundred pages of validated interview reports and the following problem: How can I make

sense of all this material? After having considered several methods I decided to apply a grounded theory approach, originally developed by two sociologists in the book *The discovery of grounded theory* (Glaser & Strauss 1967), because “Its systematic techniques and procedures of analysis enable the researcher to develop a substantive theory that meets the criteria for doing ‘good’ science: significance, theory-observation compatibility, generalizability, reproducibility, precision, rigor, and verification” (Strauss & Corbin 1990, p. 31).

Analysis in a grounded theory approach is composed of three groups of coding procedures, called open, axial and selective coding. Below these are explained with some examples from the coding and categorization of my interview data.

### 3.1 Open Coding

“Open coding is the process of breaking down, examining, comparing, conceptualizing, and categorizing data” (Strauss & Corbin 1990, p. 61). My procedure for coding each of my interview reports has been to write one or more so-called concepts for each section which bore the least relation to computer-based tools, by asking: What is this? What does it mean? What does this represent? I have continuously taken care that concepts which were closely related got the same name. In Figure 2 the coding of three sections taken from three different interviews is shown. The first outcome of the Open Coding was a classification of all the identified concepts in about 100 different groups, the so-called categories.

### 3.2 Axial Coding

“Open coding fractures the data and allows one to identify some categories, their properties, and dimensional locations. Axial coding puts those data back together in new ways by making connections between a category and its sub-categories.” (Strauss & Corbin 1990, p. 97). By looking for connections between the 100 categories from the Open Coding I first arrived at 12 categories.

### 3.3 Selective Coding

Selective coding is “the process of selecting the core category, systematically relating it to other categories, validating those relationships, and filling in categories that need further refinement and development” (Strauss & Corbin 1990, p. 116).

As an essential limitation, recommended by Strauss and Corbin, to avoid loss of clarity and precision and to achieve tight integration of categories, there can be only one core category to which all other categories should be related. In my case this led to many unsuccessful attempts to define a core category, but at last I succeeded in finding one central phenomenon to which all the categories could be related: Barriers for continuing use. Then all the pieces fell into place, and I could formulate a consistent theory.

<p>Solves problem - with re-use</p>	<p>Echo, comment on a CASE-tool: "In actual fact we could use that kind of tool because we have a fantastic problem with reusing parts. A repository where you could place and recover things would be an aid."</p>
<p>Automating manual process + requirements for the same</p>	<p>Alfa, comment on a shared database including document annotation: "It reminds me of hypertext. I think it would be smart but it presupposes on-line documents and electronic communication all along. It would require that one could sketch, point and frame like on paper."</p>
<p>1) Automating manual process - requires that tool is better</p>	<p>Hotel, comment on a meetingroom with individual workstations and a large shared screen: "Usually I do this with a blackboard. And then you can take a polaroid-picture of the blackboard afterwards. In order to be better than a blackboard and a rapporteur one must be able to achieve the same spontaneity as with a blackboard and at the same time facilitating the rapporteur-part of the work. Maybe this tool can help if the bottleneck is to propagate ideas. But if the bottleneck is to get the ideas then the tool does not help. Concerning noting it will probably not help either. Supposedly it will deadlock you into a fixed type of noting, making it impossible to state a new type of noting when needed."</p>
<p>2) Is the right problem being solved?</p>	

Figure 2: *Open Coding of three sections. The result of the coding is concepts, that is conceptual labels placed on utterances or events*

### 3.4 Back and Forth Between the Coding Procedures

The many attempts to find a core category proceeded by alternation among the three coding procedures. First I came up with an idea for a core category, then I went back to the 12 categories from the first axial coding to see whether I could link all categories to the core category in question. And when I had found a promising core category I went back to the open coding procedure to validate the core category and its linking to all the other categories. Going back and forth several times was how I at last arrived at the core category of barriers and related it to five categories:

1. WHY\_POS, Reasons for positive reaction to a tool
2. WHY\_NEG, Reasons for negative reaction to a tool
3. WHY\_USE, Reasons for putting a tool to use
4. WHY\_NOT, Reasons for not putting a tool to use
5. STOP\_USE, Reasons for discontinuing use of tool

In Figure 3 I have, as an example, shown the category WHY\_POS with some of the concepts classified by that category.

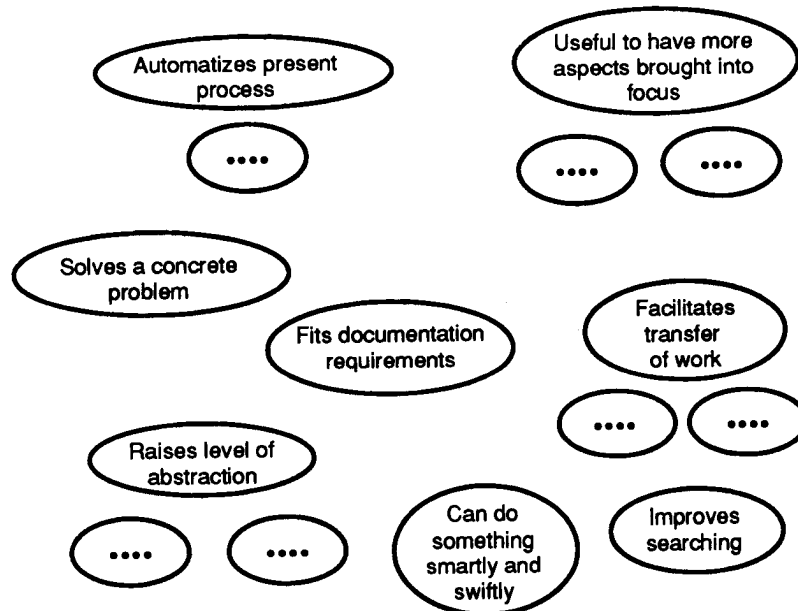


Figure 3: *Some concepts classified by the category WHY\_POS*

I found, however, in the final iteration through the three coding procedures that the concepts under WHY\_POS could be combined to one key characteristic for the category and four sub-categories:

Key characteristic: Positive reaction to a tool is always linked to perceived improved problem-solving, and maybe to:

- Sub-category 1. Automating of manual procedure
- Sub-category 2. Extension of perspective
- Sub-category 3. Raised level of abstraction
- Sub-category 4. Independence of particular persons

The relation between the concepts and the categories are illustrated in Figure 4. The crossing out of some categories means that it is found that the concept appears to be covered by the classification in the key characteristic, while the concepts grouped in sub-categories appears to imply something that adds to the key characteristic though not being an independent phenomena.

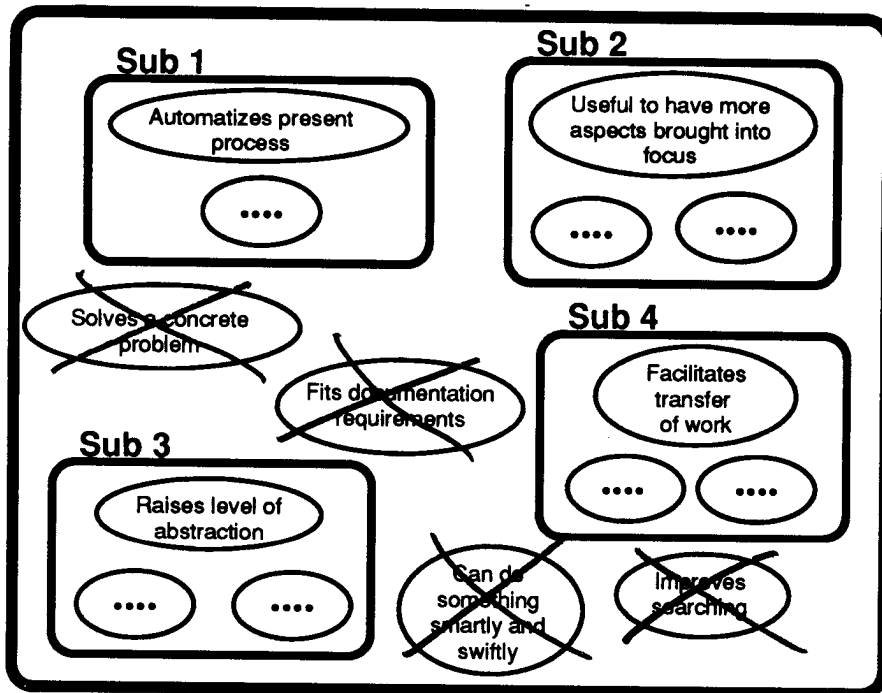


Figure 4: *WHY\_POS* with four sub-categories

#### 4 A Theory for Continuing Use of Computer-Based Tools

On the basis of the analysis presented above a grounded theory was formulated:

*Before a software developer starts using a computer-based tool three barriers have to be overcome. Firstly the information on the tool,*



whether verbal or in writing must be evaluated positively. Secondly the proper situation for putting the tool into use must have arrived. And thirdly, when the tool has been put into use, it must fit when used to remain in continuing use. In situations in which a tool is evaluated positively, but the advantages the software developer can obtain are very long-term, the tool will not be put into use at the proper situation, unless dictated by management.

Below the details of the theory are examined in the shape of a presentation of related categories and sub-categories. In this examination I have made use of quotations from interviews, partly to make the theory accessible, and partly to show the solid grounding of the theory in data.

The hierarchy of categories implied in the theory is also shown in Figure 5.

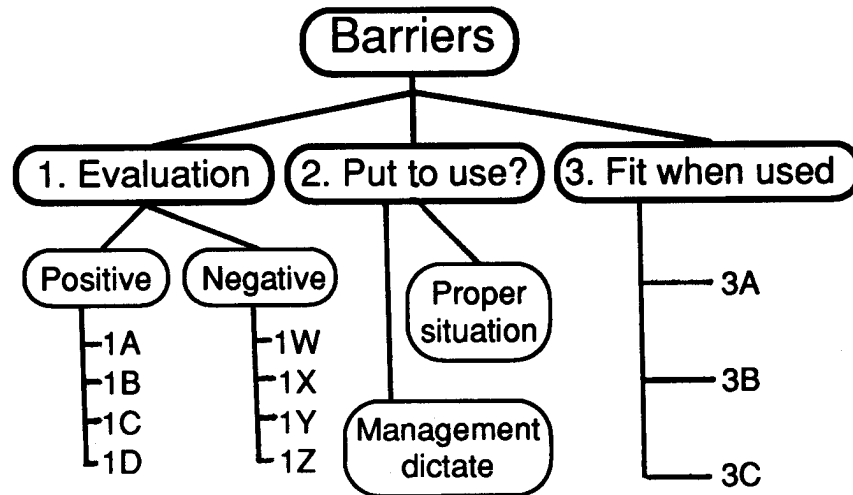


Figure 5: The category-hierarchy implied in the grounded theory

#### 4.1 Barrier 1: Evaluation of a Tool

For the information on a tool to be evaluated positively it is always a prime requirement that the software developer believes that the tool can contribute to a better problem solution for him or her, either related to a quite concrete problem or in one of the following ways:

**1A.** *By automating a manual procedure which contributes to solving a problem for the software developer.*

Golf, comment on an electronic calendar: "That would be extremely useful. It is terribly difficult to fix the time for meetings."

Foxtrot, comment on CASE-tool: "... we have projects where the demands for documentation are rather hard, and it would of course be convenient if the tool filled the requirements..."

**1B.** *By extending the software developer's perspective.*

Bravo, comment on Decision Support System: "I think that various tools of that type would be useful. seeing a thing in different lights is a good way of learning."

**1C.** *By raising the level of abstraction in problem-solving.*

Alfa, speaking about which tools he wants: "... I would like some tools doing the same as tools for analysis can do today but including the possibility of compiling all the way. I believe that higher productivity is only feasible if one can remove some phases from the development process instead of dividing the development into more phases, each with a specialized tool attached."

**1D.** *By making problem-solving independent of particular persons.*

Charlie, comment on a shared document database: "... it would also be nice to have design documents and annotations in a central place, so when somebody is to take over a case it would be easy to come straight to the point ... It would be of great value."

A negative evaluation of a tool can also be due to four categories of reasons:

**1W.** *That the tool is seen as superfluous, i.e. it is not related to the solution of a relevant problem.*

Hotel, comment on e-mail: "I do not think it is essential. It is of much greater importance that people realize that they should merely dispatch important mail. If there is so much internal junk-mail in a project they need rules for sorting and processing, then the project is the problem one should attack."

**1X.** *That the software developer doesn't believe in the concept behind the tool.*

Alfa, comment on Decision Support System: "I have no confidence in rational decision models. Occasionally I have tried to weight criteria but it did not get me anywhere, I simply got my own opinion. Ultimately I do not believe that the figures denote anything. At least my nature is such that I will manipulate the figures until the outcome is what I want to see."

**1Y.** *That the software developer thinks that the concept is OK, but the working process in which the tool is used does not suit the firm or project in question.*

Delta, comment on a meeting-room equipped with individual workstations and a large common screen: "It looks fascinating, but I do not think it will fit the way we work."

**1Z.** *That the software developer believes in the concept behind the tool and in the manual working process, but not in the automation of the working process.*

Bravo, comment on a Decision Support System: "... it would be really nice to have a reasonably fixed work procedure, some good questions and some good check points. But a computer-based tool would not constitute any improvement."

Alfa, comment on an electronic calendar: "I don't believe in it. In the first place it would be difficult to administer. Secondly I think that in many cases people would be too shy to write all their appointments in the calendar."

## 4.2 Barrier 2: The Proper Situation or Dictate from Management

Even if the information on a tool is evaluated positively, it does not follow that the tool will be put into use right away. The proper situation has to be reached, i.e. the problem to which the positive evaluation was related is felt to be urgent.

Golf, talking about a successful tool: "We have had a major success with a test tool ... it was a tool that did not have to be sold internally. When it became available people started using it voluntarily and with pleasure."

Besides, it must be a tool from which some advantages can be derived quickly. In the case of tools which get a very positive evaluation, but has exclusively long-term advantages, it will not be put into use by the software developer (on his own initiative) even if the time is proper. That kind of tool requires a dictate from management.

Delta, comment on a tool: "Certainly it would be a smart system, but I do not know how far you can bring up people to use it. Probably it is a tool which in some organizations would demand a dictate from above before one started to use it."

Golf, comment on a problem-structuring tool: "The problem with a new tool is that on the face of it one cannot see the advantage because here and now, in the project one is working on, one might as well use the old method. What the tool gives you is long-term advantages, however it is hard to anticipate a very long time. And we have to fix this before tomorrow! ... without doubt we could get a lot of pleasure out of it but I doubt we could make out the introduction."

## 4.3 Barrier 3: Does the Tool Fit When Used?

If a tool is to remain in continuing use the primary requirement is that the perceived improvement in problem-solving which caused the positive evaluation (Barrier 1), is obtained wholly or in part. Besides it is needed:

**3A.** *That the tool does not impede problem-solving unnecessarily.*

Echo, talking about experiences with a project management tool which they tried to use, but which did not fit the problem which needed solution: "... (The tools) are not made for managing software projects. They are made for digging ditches. If two men are going to dig a ditch then the tool partly demands that the two men start at exactly the same time, and partly demands that the two men work entirely simultaneously until they have finished the ditch. Things do not take place like this in software development projects."

**3B.** *That the tool is technically suitable.*

Golf, facing a portability problem, i.e. to get the desired tool for the operating system used as their development platform: "... that is why we regret to have to look for another test tool, not quite as good as the first."

**3C.** *That the tool fits with other tools in continuing use.*

Alfa, speaking about tools in general: “I have the impression that we and others have taken quite a few initiatives using different tools. One often hears about tools of which it is alleged that one can improve one’s performance by a factor 10. But it appears to be incredibly difficult to integrate the tools into the organization. I believe it is because the tools are not integrated, i.e. one cannot use the same tool throughout the whole development course . . . on the whole the use of tools tends to settle around the compiler and the debugger. Those are the only tools people hold on to.”

## 5 Discussion

Perhaps the major contribution of this article is to show that grounded theory can function as a tool for a researcher in structuring and conceptualizing a very large number of qualitative data. Compared to a more intuitive approach grounded theory has the following advantages:

1. The analysis of data is well-documented so it is possible to trace the derivation of any category back to concepts, coding-memos and the actual transcripts of interviews.
2. Besides it will be possible to use the grounded theory approach as a solid foundation for subsequent survey work, thus avoiding quantitative evidence of intuitively developed concepts that are imperfect.
3. The grounded theory method allows the data to play a prominent role in determining appropriate categories.

### 5.1 Validity of the 3-Barrier Grounded Theory

As part of going back and forth between the coding procedures (see Section 3.4) one validates the discovered theory by asking: Is this statement valid in a broad sense for each interviewee? And if not relevant categories are changed or enhanced.

When discovering and determining the categories in this study several incidents from the data provided the material for defining a category. If many incidents illustrate the category it is called well-saturated (Calloway & Ariav 1990). In the theory on the three barriers to continuing use the first- and the last-mentioned barriers are related to incidents in more than half of the 19 interviews or to incidents in more than half of the 9 projects. Therefore my opinion is that these aspects of the theory can be called well-saturated.

On the other hand, the 2nd barrier is only supported by data from three out of the nine projects. So even if all data point in the same direction, it is my view that further research is needed before this part of the theory can be called

well-saturated. In addition this view is supported by a number of interesting implications of the theory, which ought to be investigated further, such as: What happens when a tool is seen to have a mixture of short-term and long-term advantages. Will a dictate from management then be necessary? And will a dictate from management prescribing the use of a tool in itself create so much resistance against the use of the tool that it will never be put into continuing use by the software developers?

Another limitation to the applicability of the theory is partly that it is based on data from software product development projects, partly that the nine projects which entered into my research must be described as technically oriented projects, i.e. non-administrative. Consequently, although it may seem tempting to generalize the theory of the three barriers further studies are needed before this can be ventured.

## 5.2 Related Work

One would expect to find closely related work in information systems research and in research on introducing standard application packages and CASE tools. But only to a certain extent that is the case as we shall see:

Markus (1983) claims that three basic theories underly many MIS- implementation rules and prescriptions. 1. Resistance because of factors internal to person or group. 2. Resistance because of factors inherent in application. 3. Resistance because of an interaction between characteristics related respectively to the people and to the system. She argues for the 3rd and gives some recommendations for implementation, e.g. "Fix organizational problem before introducing system," which are very focused on in-house development of computer systems. Compared to the 3-barrier (grounded) theory, focused on commercial software products, only the overall conclusion, that resistance originates in the interaction between application and person/group, is to a certain extent similar.

In a recent survey of CASE research by Aaen & Sørensen (1991) it is pointed out that "despite the grave problems of software development and the great expectations for CASE, this new technology seems to be spreading at a fairly low pace." Aaen & Sørensen do not go into a discussion of why the pace is slow. Instead they adopt an innovation diffusion model and attribute the slow pace to the S-curve of the diffusion process, with CASE being in the early adoption phase. So the only mention of something similar to the 3-barrier theory is the following (Aaen & Sørensen 1991, p. 19): "Last, with respect to the introduction process in general, problems with accepting the tool and other organizational issues may emerge in each of the four introduction activities."

There is very little research on standard software. One of the few is Anveskog *et al.* (1984) on the introduction of standard application packages into an organization. Unfortunately one cannot compare the 3-barrier theory to Anveskog *et al.* (1984) because they only look at introduction of very large strategic applications, so large that one can afford to go through a 15-step method including the

Pries-Heje and Plads: THREE BARRIERS FOR CONTINUING USE OF COMPUTER-BASED TOOLS IN INFO preparation of market surveys during 11 of the 15 steps (Anveskog *et al.* 1984, p. 163).

More related work can be found in marketing and innovation theory:

A key question in marketing is how customers decide what to buy, if anything (Urban & Star 1991). A marketing manager can reach an understanding of the customers' decision behaviour by using an individual information processing model, which in its simplest form comprises three activities:

1. Input (e.g. product information).
2. Decision Making (e.g. decision rules, and which information is remembered).
3. Output (e.g. buying and using).

Compared to this the 3-barrier theory contributes with the specific focus on improved problem solving as the reason for positive evaluation of a computer-based tool (Barrier 1 and 3). Furthermore, the dynamic dimension, that is the "proper situation" and the perceived horizon of the advantages (Barrier 2), are not to be found within the scope of the individual information processing model.

Innovation literature is dominated by the adoption and diffusion perspectives on innovations, which are based on the premise that all innovations are good for the consumer, that is a "pro-innovation bias" (Rogers 1983). S. Ram (1987, p. 208) suggests that one should give attention to the process of innovation resistance: "Adoption begins only after the initial resistance offered by the consumers is overcome . . . If the resistance is too high, the innovation dies and there is no adoption." The scope in this innovation resistance research is very similar to that of the 3-barrier theory. But the results are somewhat different. E.g. (Sheth & Ram 1987) identifies five customer barriers to innovation:

1. Usage (incompability with existing work flows, practices, habits).
2. Value (Low performance-price ratio).
3. Risk (economic loss, danger or uncertainty).
4. Tradition (norms and culture).
5. Image (of the product).

Only categories 1Y and 1X, reasons for negative evaluation, can be conceived as equal to Sheth's 5 barriers, respectively '1. Usage' and '4. Tradition.'

## Acknowledgements

The author greatly benefited from a visiting scholarship at the School of Management of the State University of New York (SUNY) in Binghamton from October 1990 to June 1991, financed by the Danish Research Academy. Especially I would

like to thank Richard Baskerville for encouraging me to use the grounded theory approach, and Heinz K. Klein for improving the presentation of ideas. I would also like to thank Søren Lauesen and Søren M. Borch, both of the Copenhagen Business School, for their many comments and thought-provoking conversations. And last but not least I would also like to thank the three anonymous reviewers for several constructive comments.

## References

- Anveskog, L., A. Nilsson & I. Nord, (1984). *Verksamhetsutveckling, Att välje standard-system (In Swedish)*. Studentlitteratur, Lund.
- Bly, S. A., (1988). A Use of Drawing Surfaces in Different Collaborative Settings. In (Tatar 1988,p. 250–256).
- Calloway, L. J. & G. Ariav, (1990). Developing and using a qualitative methodology to study relationships among designers and tools. In Nissen, H., H.K. Klein & R. Hirschheim, editors. *Proceedings from the IFIP TC 8 WG 8.2 Working Conference on The Information Systems Research Arena of the 90's (ISRA-90)*. Lund, Sweden. Pages 415–433.
- Conklin, J. & M. L. Begeman, (1988). gIBIS: A Hypertext Tool for Exploratory Policy Discussion. In (Tatar 1988, p. 140–152).
- Cooper, R. G., J. Arleth & H. C. Juhl. *Vurdering og valg af nye produktider (In Danish)*. Teknologisk Institut, Driftsteknik. Tåstrup, Denmark.
- Curtis, B., H. Krasner & N. Iscoe, (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11):1268–1287.
- Glaser, B. & A. Strauss (1967). *The discovery of grounded theory*. Aldine, Chicago.
- Grudin, J., (1988). Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. In (Tatar 1988, p. 85–93).
- Kohl, B., B. Meyer, L. Ginnerup & D. Kjaersgaard, (1989). *Et værktøjskatalog (In Danish)*. Samfundslitteratur, Frederiksberg, Denmark.
- Kraemer, K. L. & J. L. King, (1988). Computer-Based Systems for Cooperative Work and Group Decision Making. *ACM Computing Surveys*, 20(2):115–146.
- Leland, M. D. P., R. S. Fish & R. E. Kraut, (1988). Collaborative document production using Quilt. In (Tatar 1988, p. 206–215).
- Malone, Th. W., K. R. Grant, K. Lai, R. Rao & D. Rosenblitt, (1987). Semi-structured Messages Are Surprisingly Useful for Computer-Supported Coordination. *ACM Transactions on Office Information Systems*, 5(2):115–131.
- Mantei, M., (1988). Capturing the Capture Concepts: A Case study in the Design of Computer-Supported Meeting Environments. In (Tatar 1988, p. 257–270).
- Markus, M. L., (1983). Power, Politics, and MIS Implementation. *Communications of the ACM*, 26(6):430–444.
- Martin, J. & C. McClure, (1988). *Structured Techniques: The Basis for CASE*. Prentice-Hall, Englewood Cliffs, New Jersey. Revised Edition.
- McClure, C., (1989a). *CASE is Software Automation*. Prentice-Hall, Englewood Cliffs, New Jersey.
- McClure, C., (1989b). *Tutorial Notes from the European CASE Conference*. Blenheim Online, London.
- McCracken, G., (1988). *The Long Interview*. Sage University Paper Series on Qualitative Research methods, Vol. 13. Sage, Beverly Hills, California.

- Patton, M. Q., (1990). *Qualitative Evaluation and Research Methods*. Sage Publications, Beverly Hills, California. Second Edition.
- Ram, S., (1987). A Model of Innovation Resistance. *Advances in Consumer Research*, (14):208–212.
- Rogers, E. M., (1983). *Diffusion of Innovations*. The Free Press, New York.
- Sheth, J. N. & S. Ram, (1987). *Bringing innovation to market*. John Wiley & Sons, Inc., New York.
- Stefik, M., G. Foster, D. G. Bobrow, K. Kahn, S. Lanning & L. Suchman, (1987). Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, 30(1):32–47.
- Strauss, A. & J. Corbin, (1990). *Basics of Qualitative Research. Grounded Theory Procedures and Techniques*. Sage Publications, Beverly Hills, California.
- Sørensen, C. & I. Aaen, (1991). A CASE of great expectations. *Scandinavian Journal of Information Systems*, 3:3–23.
- Tatar, D., editor, (1988). *Proceedings of the 2nd Conference on Computer-supported Cooperative Work*. ACM, New York.
- Urban, G. L. & S. H. Star, (1991). *Advanced Marketing Strategy. Phenomena, Analysis, Decisions*. Prentice-Hall, Englewood Cliffs, New Jersey.
- Winograd, T. & F. Flores, (1986). *Understanding Computers and Cognition, A New Foundation for Design*. Ablex, Norwood, New Jersey.

## Appendix 1: Extract of Interview Guide

*The project itself:* Can you give a brief description of what the project aim at? For how long has the project been going? How is the project organized?

*Methods and tools:* Can you tell me about the method and the procedures in the project? Do you use specific tools—why? Do you make any kind of prototypes? Have you tried using mathematical based methods or tools like the Vienna Development Method or Z?

*Communication:* What is your impression of the culture in the project? How do you manage documents in the project? Do you have a configuration control system or a version control system? How do you manage loose memos, that is memos which do not fit into the project at the moment?

*New methods and tools:* Are there any methods or tools you have dreamt of?

## Appendix 2: Checklist of 20 Existing Methods and Tools

In the third part of the interviews I collected reactions to a checklist of 20 existing methods and tools. Concretely I asked the interviewees about the following,

“Below a number of tools and methods are mentioned that you could have had at your disposal of the project. You are now requested to



evaluate the effect on productivity and quality on a scale from one (no effect) to five (tremendous effect) by putting a circle around the figure that you consider the best expression of the effect.”

Here is an extract of the checklist:

1. Posting simple electronic mail to other members of the project. For instance:

To:	XXX
From:	YYY
Subject:	ZZZ
Free text:	≡ ≡

-1	0	1	2	3
Adverse	No	Small	Moderate	Large

2. Advanced electronic mail including the capability of posting semistructured messages that means messages of identifiable types, with each type containing a known set of fields, but with some of the fields containing unstructured text or other information. For instance a seminar announcement:

SEMINAR ANNOUNCEMENT	
To :	ALL
From :	JAN
Date :	JANUARY 30 , 9:00 am
Subject :	STATUS MEETING
Place :	GREEN MEETINGROOM
Text :	Compulsory meeting! Now we got to have control of bug fixing. I will forward a suggestion Friday.

Fixed Fields

Free unstructured text

Figure 6: Extract of the checklist

### Appendix 3: Framing of the Checklist

The first version of the checklist was drawn up in connection with interviews in the first project, the university research project India. Based on experience from India the second and final version of the checklist was made. Below a brief description and the source of each of the 20 described methods and tools in the final checklist are stated.

1. Possibility of posting simple electronic mail to other members of the project. Included as basis for evaluating the other sorts of electronic mail (2. to 5.).
2. Advanced electronic mail including the possibility of posting semistructured messages, that means messages of identifiable types, with each type containing a known set of fields, but with some un-structured text fields too. Source: First half of Information Lens (Malone *et al.* 1987).
3. Advanced electronic mail including the possibility of constructing rules for processing messages. Source: 2nd half of Information Lens (Malone *et al.* 1987).
4. Advanced electronic mail including the possibility of managing electronic conversations for action. Source: The Coordinator (Winograd & Flores 1986).
5. Advanced electronic mail including an automatic meeting scheduling feature. The system implies that everyone in the project maintain a personal electronic calendar. Inspired by (Grudin 1988).
6. Video conference facilities for two or more persons geographically separated. The video link can either be used to show a frontal face view of a person or it can focus on a chalkboard. Inspired by (Bly 1988).
7. Video conference facilities for two or more groups geographically separated. For instance a group can be composed of 3 to 8 active members. The camera and the microphone are automatically aimed at the person speaking, but a chairperson can interrupt. Source: (Kraemer & King 1988).
8. A document database shared by all participators in the project. But naturally you can have private documents in the database too. Included as basis for No. 9 and 10.
9. A computer with a large screen that everyone in the meeting room can see. There is access to the shared document database from the computer. Included as basis for No. 11 and 12.
10. A shared document database including the possibility of annotating a document or another annotation. Source: Part of the Quilt system (Leland *et al.* 1988).
11. A meeting room equipped with personal workstations in a network complemented by a common large screen. The participants can either create text and sketches individually. Or the group can create texts and sketches in common. Inspired by the Capture Lab (Mantei 1988).
12. In the same meeting room as above, with individual workstations and a common large screen, you have available a tool to organize ideas. Source: The Cognoter system (Stefik *et al.* 1987).

13. A system helping to decide whether you ought to continue the product development by asking about 50 critical questions, that is questions which are known to have significance for the success or failure of a product. Source: The Newprod or Danprod system (Cooper *et al.* 1986).
14. A system supporting selection between alternatives. At the beginning all the alternatives and selection criteria are posed. Henceforth the alternatives are evaluated and ranked. Source: The Priorities system (Kohl *et al.* 1989).
15. A CASE-tool including the possibility of drawing diagrams such as data-models and dataflows, where the consistency automatically are being checked. Inspired by (McClure 1989a, 1989b).
16. A CASE-tool including the possibility of generating a database automatically from a datamodel. Inspired by (McClure 1989a, 1989b).
17. A CASE-tool where all diagrams, programs, screen designs, data etc. are stored at one place viz. the repository. Inspired by (McClure 1989a, 1989b).
18. A CASE-tool including the possibility that modifications somewhere automatically result in changes in all other relevant places, or at least the CASE-tool points out to the systems developer all relevant places where changes have to be made. Inspired by (McClure 1989a, 1989b).
19. A CASE-tool including the possibility of automating the test of a program entirely or partially, by using formal mathematical methods for the requirements specification. Inspired by (Martin & McClure 1988).
20. A system that helps managing three essential elements in the design process: Issues, Positions and Arguments. Source: The gIBIS system (Conklin & Begeman 1988).