SYSTEM DEVELOPMENT METHODOLOGIES: UNANSWERED QUESTIONS AND THE RESEARCH-PRACTICE GAP

Judy L. Wynekoop
The University of Texas, San Antonio

Nancy L. Russo
Northern Illinois University

Follow this and additional works at: http://aisel.aisnet.org/icis1993

Recommended Citation
http://aisel.aisnet.org/icis1993/15

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 1993 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
SYSTEM DEVELOPMENT METHODOLOGIES: UNANSWERED QUESTIONS AND THE RESEARCH-PRACTICE GAP

Judy L. Wynekoop  
Division of Accounting and Information Systems  
The University of Texas, San Antonio

Nancy L. Russo  
Operations Management and Information Systems  
Northern Illinois University

ABSTRACT

There is much we don’t know about information system development and the use of systems development methodologies. Research to date has focused on the development of new methodologies. There has been little evaluation of these methodologies or examination of the selection, development, adaptation, or use of methodologies in practice. This paper discusses this lack of knowledge, its ramifications, and key research questions.

1. INTRODUCTION

Although many diverse system development methodologies exist, there is not universal agreement that existing methodologies are useful in today’s environment (Baskerville, Travis and Truex 1992; Lyytinen 1989), nor is there agreement that they were ever useful (Lyytinen 1987b). Research has focused on the development of new methodologies and frameworks for the selection, evaluation and use of methodologies, not on their evaluation or use in practice.

This narrow focus and lack of “real world” research has critical ramifications. Current research may overlook changes occurring in organizations and produce findings or suggestions that are irrelevant to practice. By failing to evaluate current practices and needs, researchers may develop methodologies that are not only irrelevant, but flawed (Lyytinen 1987b).

This paper discusses existing research on systems development methodologies and challenges readers to pursue a new research paradigm to understand how systems are developed in today’s organizations, how well methodologies work, and what practitioners need today and in the future, before developing more guidelines or methodologies.

2. BACKGROUND

Over a decade ago, IFIP WG 8.1 initiated CRIS (Comparative Review of Information Systems Design Methodologies) because “it seemed inappropriate for the Working Group to develop further...approaches [to developing information systems] without first undertaking some kind of systematic study of the existing state of the art” (Olle 1982, p. 1). However, more than a decade later, little more is known about the “existing state of the art.” In this section, methodologies are briefly discussed and defined.

2.1 The History and Purpose Methodologies

Development methodologies originated to improve the management and control of the software development process, structure the process, reduce its complexity, and standardize the development process and product by specifying activities done and their relationship. Methodologies may be commercial, sold or recognized outside a single organization; “homegrown,” developed and used within one organization; or academic, developed and used within a research context.

There are hundreds of published methodologies and most likely thousands in use, since many information systems (IS) organizations develop their own. These methodologies, many developed in academic institutions and never used in practice, are “more or less similar” (Bubenko 1986, p. 289). Methodologies have been relatively static since their inception, generally based on the “waterfall” process model. Recently, these linear, process-oriented methodologies have been supplemented with methodologies based on evolutionary development and data-oriented and object-
oriented paradigms. The bottom line in practice and research is that, despite frenzied activity developing methodologies, there remains no generally accepted theory of IS development, nor has there been a systematic investigation of evidence of the deficiency of methodologies in use (Bubenko 1986; Lyytinen 1987b).

2.2 What is a Methodology?

The divergent opinions of what constitutes a methodology is problematic when discussing methodology research. Although a complete discussion is beyond the scope of this paper, in this section some of the viewpoints are discussed and “methodology” as it is used in this paper is defined.

First there is the methodology versus methodology debate. Some argue that, since methodology means a “science of methods,” the term has no place in information systems (Baskerville, Travis and Truex 1992; Schach 1993). There are also those who use “method” and “methodology” interchangeably (Colter 1984; Connors 1992), those who consider “methods” to encompass “methodologies” (Davis 1982; Hackathorn and Karimi 1988), and those considering “methodologies” to encompass “methods” (Hirschheim 1985). Because it appears that “methodology” is still the most widely used term, it is used in this paper.

Second, what is a methodology? Definitions run from the generic, “a set of guidelines that prescribe a behavior in order to think and act in a situation” (Nielsen 1989, p. 82), to the precise, “specific, step-by-step strategies for completing one or more phases of the systems development life cycle...[imposing] tools and standards on the SDLC” (Whitten, Bentley and Barlow 1989, p. 111), and everything in-between. “Methodology” has been used to refer to a single phase in the development process (Schach 1993), to “all aspects from initial problem identification...to the design of alternative solutions” (Bantleman and Jones 1984, p. 214), and to IS planning, analysis and design (Olle et al. 1988). Many discussions of methodologies omit definitions. Given the divergent opinions of what a constitutes a methodology, it seems essential to define the term in any publication.

Analysis of the dominant definitions used in methodology research identified three distinct entities which have been called methodologies. We offer the following definitions, largely based on IEEE (1987), Lyytinen (1987b), and Schach (1993), to clarify terminology as we use it:

- **Methodology**: A systematic approach to conducting at least one complete phase (e.g., design; testing) of software production, consisting of a set of guidelines, activities, techniques and tools, based on a particular philosophy of system development and the target system.

- **Technique**: specific steps for conducting a portion of a phase of the software development process.

- **Software Process Model**: a representation of the sequences of stages (e.g., requirements analysis, specification, planning, design, implementation, integration, maintenance, and retirement) through which a software product evolves.

“Methodology” is used by some to mean methodology as defined above, but by others to mean a software process model or a technique. For instance, is Structured Analysis and Systems Specification a methodology (Nielsen 1989; Olle et al. 1988) or is it a technique (Mendes 1980)? Is prototyping a methodology (Burch 1992; Burns and Dennis 1985; El Louadi, Pollalis and Teng 1991), a technique (Jayartna 1988; Olle et al. 1988), or a software process model (Schach 1993)? Is CASE a methodology (McLaughlin 1993)?

According to our definitions, Information Engineering, Structured Analysis and System Specification, SADT, PSL/PSA, object-oriented design, and Jackson System Design are methodologies. Prototyping, the waterfall model, the spiral model, and incremental development are process models. JAD and data flow diagramming are techniques. CASE is a tool.

At best, inconsistent terminology in the literature is confusing. The confusion is not surprising: methodologies and process models are intertwined (Floyd 1986). Methodologies are defined within frameworks provided by software process models. For example, Information Engineering, a methodology, defines techniques, deliverables and activities within the framework provided by the waterfall process model.

This is not merely semantics, but has implications for methodology research. For example, surveys that compare the use or effectiveness of “prototyping” to the SDLC, or “structured approaches” to prototyping (e.g., Mahmood 1987; Necco, Gordon and Tsai 1987) are problematic, since multiple methodologies may be used within each development process. Comparisons of the same process model may be invalid if all organizations do not use the same methodologies. For example, Company A’s SDLC might include very different methodologies and techniques than Company B’s, thus producing different results.

Although we follow the definitions above, due to the interrelationship of process models and methodologies and their interchangeability in the literature, both are included in this discussion. However, the term “system development method” (SDM) will be used to incorporate both methodologies and process models and “methodology” to refer to methodology only. Research in both areas is vital, but researchers (and readers) must identify the phenomena
under study and whether comparisons are valid. For simplicity, this discussion focuses on methodologies as they are used in the analysis and design phases of a software process.

3. THE PROBLEM

Olaisen’s (1991) categorization of our understanding of information systems is instructive when applied to our knowledge of methodologies. Knowledge can be classified into four categories:

(1) What we know;
(2) What we know that we don’t know;
(3) What we don’t know that we know; and
(4) What we don’t know that we don’t know.

Most published methodology research assumes our knowledge about IS development falls into the first two categories: we either understand something or know what it is that we must find out. However, researchers and practitioners may know more about developing systems than we realize. Unfortunately, it is more likely that there is much about IS development and methodology use that we don’t realize that we don’t know. Unfortunately, this can bias our assumptions and actions and may cause us to misinterpret situations and results (Olaisen 1991).

There are two critical areas where our research community has assumed we “know” or, if we don’t know, that it’s unimportant to know. In the literature, it is implicitly assumed that: (1) methodologies (public or homegrown) are used, and are useful and effective; and (2) the frameworks that have been developed to evaluate and select methodologies are useful.

These assumptions have not been validated and must be examined. We must understand how systems are developed, evaluate currently used and proposed methodologies, and find out what practitioners need before we propose new ways to develop information systems. Although we may not need to understand every detail of current practice, we should have at least a general knowledge of what is done, how well it works, and what is needed.

In the following section, a preliminary list of research issues is discussed, with the disclaimer that only those issues in Olaisen’s first two categories (what we know that we know and what we know that we don’t know) can be included. Interpretive research is needed to uncover issues about which we are still unaware.

4. RESEARCH AGENDA

Because of the inadequate base of empirical evidence, numerous questions about methodologies remain. Table 1 summarizes the most important research issues we have identified to date. In this section, each research issue is discussed in terms of what is and is not known.

4.1 Are Methodologies Used? If Not, Why Not?

This appears to be a simple question. However, the little we know comes from a small base of survey results and practitioners’ narratives describing events in their organization. For instance, we know that New York Life has used a structured homegrown methodology incorporating data modeling since 1984, and that CASE tools were acquired to support it since developers considered it to be tedious (Zagorsky 1990). We do not know why this particular methodology was used, how well it worked, or how it was developed.

In 1986, Edward Yourdon wrote that only 10% of North American information systems organizations used structured techniques (his terminology) in a “disciplined fashion.” On the other hand, a year later, survey results were published showing that 69% of the sample used structured “approaches” to developing information systems (Necce, Gordon, and Tsai, 1987). Which is right? Since we don’t know the extent to which survey respondents used structured methodologies, it is impossible to know.

Perhaps the focus of our question should not be whether or not methodologies are used, but: “To what extent are methodologies used, and how are they used?” When an IS manager surveyed says a firm uses a particular methodology, we should interpret that to mean, “We have the meth-

Table 1. Methodologies: Unanswered Questions

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Are methodologies used? If not, why not?</td>
</tr>
<tr>
<td>2.</td>
<td>How are homegrown methodologies developed?</td>
</tr>
<tr>
<td>3.</td>
<td>How are methodologies selected?</td>
</tr>
<tr>
<td>4.</td>
<td>Do methodologies work?</td>
</tr>
<tr>
<td>5.</td>
<td>When are specific methodologies successful?</td>
</tr>
<tr>
<td>6.</td>
<td>Are methodologies obsolete?</td>
</tr>
</tbody>
</table>

183
odology and I assume it is being used and applied correctly and completely." It should not be assumed that because an organization has a methodology all projects follow it (Pressman 1982; Thayer, Pyster and Wood 1981), nor should it be assumed that a survey respondent knows how systems are developed in his organization. For example, in a recent survey, almost one-fourth of the 112 IS managers responding described their methodology as "unknown" (Dekleva 1992).

Most surveys to date provide a general idea of methodologies in use, since they do not distinguish process models from methodologies (Dekleva 1992; Necco, Gordon, and Tsai 1987). In one, the confusion of methodologies, tools and techniques by respondents was noted by the authors (Beck and Perkins 1983).

An indication of the reasons methodologies may not be used is seen in a field study of the design process used in large systems, although the study's focus was not methodologies (Curtis, Krasner and Iscoe 1988). Many software engineering practices were found to fall apart when applied to large systems with deadline pressures. Interviewees indicated that conditions surrounding their projects prevented ideal development practices from being used. More studies of this type are needed to help identify why methodologies are not used or how they are used.

At best, surveys provide a snapshot of methodology use — a general idea of what methodologies are in use. To understand the motivations for using particular methodologies in particular contexts and the historical and contextual influences on the use or nonuse of methodologies, longitudinal field research, particularly interpretive studies, is necessary.

Corollaries to this question are: How are systems developed when no methodology is used, and what happens when methodologies are not used? Is a methodological development successful? Only a database of results from numerous field studies (both qualitative and quantitative) will answer these questions. Without answers to these questions, there is no basis for creating new development technologies, such as methodologies, techniques, process models or CASE tools.

4.2 How are Homegrown Methodologies Developed?

We know that some organizations develop their own methodologies or adapt commercial methodologies for internal use. In a recent survey of over one hundred organizations, 65% of the organizations had developed their methodology in-house rather than purchase a commercial one (Russo and Klomparens 1993). However, we know little about the nature of "homegrown" methodologies or how they are developed.

The adaptation of methodologies to fit a particular situation appears to be common. In a survey of IS managers, half (49%) agreed that there should be a single methodology used on all projects. However, 89% of these respondents believed that formal methodologies should be adapted on a project-by-project basis (Russo and Klomparens 1993). We do not know how such adaptation is done, how frequently it is done, whether there are any controls over the changes, and how well the adapted methodologies work.

Again, although surveys might provide some insight into the development of homegrown methodologies, longitudinal, qualitative studies will provide an understanding of the reasons for, process of, and influences on, the development of homegrown methodologies. Answers to these questions are important to shape new system development technologies to improve the development process and resulting software products.

4.3 How are Methodologies Selected?

Assuming methodologies are used, how do practitioners select them? Most publications on methodology selection have been frameworks or guidelines to help the reader understand (Fitzgerald, Stokes and Wood 1985; Olle et al. 1988), classify (Hackathorn and Karimi 1988; Wood-Harper and Fitzgerald 1982), or select (e.g., Connors 1992; El Louadi, Pollais and Teng 1991; Nielsen 1989; Wood et al. 1988) methodologies. No empirical validation of the frameworks could be found.

Only one empirical study addressing how methodologies are selected or adapted was found: a field study of eight Finnish companies to identify how CASE tools and methodologies are selected and adapted to one another (Smolan- der, Tahvanainen and Lytynen 1987). Although the authors found little methodology adaptation, that which they found was done by trial and error and was prompted by the purchase of a CASE tool incompatible with existing practices or the use of an inadequate, non-standard methodology that required improvement. The data also indicated that methodologies are not used until automated tools supporting them are available.

Surveys have been used to identify the types of SDMs in use. In a 1987 study, about two-thirds of a 97 firm sample used traditional IS development approaches, the SDLC, and structured methodologies, while less than half used prototyping; and organizations often used more than one SDM
(Necco, Gordon and Tsai 1987). In a more recent survey, over three-quarters (76%) of a 133 firm sample reported using an SDLC-based approach to system development (Russo and Klomparens 1993).

Practitioners are faced with numerous methodologies. Since a cause of development failure is the use of inappropriate or inadequate methodologies, it is important that practitioners choose the correct methodologies or techniques (Lyytinen 1987b; Nielsen 1989). Yet, although taxonomies and selection guidelines have proliferated, they have not been empirically validated, nor do we know how methodologies are successfully selected in practice. Although surveys and interviews may provide preliminary answers, qualitative research is needed to understand the motivations of, and influences on, the selection process.

4.4 Do Methodologies Work?

Deficiencies in methodologies have been cited as a cause of IS failure (Lyytinen 1987a, 1987b). Assuming that methodologies are used, are they effective? What do they accomplish? The answers to these questions lie in research to evaluate methodologies. Unfortunately, with the exception of Avison and Wood-Harper’s (1991) evaluation of Multi-view, we could find no systematic evaluations of methodologies in use under realistic conditions.

Recent field studies and surveys have only begun to evaluate the effectiveness of methodologies. A study of 65 maintenance projects from one firm indicated that the use of structured methodologies for maintenance increased the time spent on analysis and design activities, adding 2.17 hours per function point to the project (Banker, Datar and Kemerer 1991). A survey of 122 IS organizations found no relationship between the use of “modern” software development methods (defined in the study as structured methods, information engineering, prototyping, or CASE tools versus the traditional SDLC) in original development and the amount of time spent maintaining systems, although use did decrease the amount of time spent correcting errors (Dekleva 1992).

The CRIS workshops began to evaluate methodologies. The second workshop (Olle, Sol and Tully 1983) consisted largely of “paper” comparisons and evaluations of methodologies. In CRIS I (Olle, Sol and Verrijn-Stuart 1982) and CRIS III (Olle, Sol and Verrijn-Stuart 1986), a limited number of methodologies were evaluated by using them on a standard case problem. Unfortunately, few true evaluations in practice were done. An exception was Floyd (1986), who evaluated four methodologies used by college students on case study problems. CRIS was an outstanding start, but an academic exercise if not followed by evaluations of methodologies on real problems in real contexts.

A study of eight Finnish firms found that when methodologies were used, they were considered helpful. Their main advantage was considered to be a better understanding of design options and problems, not improved productivity or software quality. However, the methodologies were not enforced in most firms (Smolander, Tahvainen and Lyytinen 1987). A recent survey found that only 45% of a sample of over 100 IS managers were satisfied with their methodology (Russo and Klomparens 1993).

Published practitioner evaluations of their methodologies are infrequent and their contribution is limited. For example, although we know that Exxon used a methodology based on Jackson’s Program Design methodology, and that productivity did increase as a result of its use (Menard 1980), we don’t know how this methodology would work in different contexts or how the methodology evolved.

The evaluation of object-oriented methodologies has only started. For example, two object-oriented analysis and design methodologies were used by researchers on the same problem and the resulting designs measured for complexity (Sharble and Cohen 1993). Large-scale studies of real systems will provide more information.

This question raises a difficult issue: How do we define, much less measure, the “success” of a methodology? Is it developer, MIS Director, or user satisfaction with the process and/or product; design complexity; or software maintainability? Is it all these things? Although researchers may develop their own measures, we should first look to practice: surely someone has evaluated their methodology.

Evidence is emerging that any methodology is beneficial only in the context of an organization with a pre-existing ability to produce quality software (Humphrey 1989; Loy 1993). Therefore, methodologies must be evaluated in the context in which they are used. Factors other than the methodology or object system, such as application and domain knowledge of the developer, in addition to the software process capabilities of the organization, must also be considered (Curtis, Krasner and Iscoe 1988; Vitalari 1985). Without a large base of systematic evaluations of various methodologies in different organizations, we can draw no concrete conclusions. Again, although surveys and laboratory experiments provide initial information, field experiments and qualitative research will be needed to truly understand the issue. Field experiments may be used to compare the development of the same system using two different methodologies. Interpretive studies will be needed.
to relate the use of methodologies and how they are used to their "success" in specific contexts. If we do not know how systems are successfully developed, we have no basis to improve the development process.

4.5 When are Specific Methodologies Successful?

Most researchers acknowledge that all methodologies are not equally applicable in all situations (Episkopou and Wood-Harper 1986; Jayartna 1988; Kumar and Welke 1992; Nielsen 1989; Olle et al. 1988), and methodologies have been characterized by their strengths and weaknesses (Avison and Fitzgerald 1988). However, there is little empirical insight into why some methodologies might be better than others in certain situations.

Laboratory studies comparing prototyping to the SDLC or evaluating parts of SDMs have been conducted (Alavi 1984; Boland 1978; Boehm, Gray and Seewaldt 1984). One field survey confirming their results was found (Necco, Gordon and Tsai 1987), while one found users had no real preference (Mahmood 1987). In such surveys, however, it isn't clear how many methodologies were actually studied, since numerous methodologies may have been represented in the samples. However, recent case studies conducted in Europe indicate the limited generalizability of laboratory results. Findings include: users cannot articulate requirements or contribute to the design process as much as designers hope; user satisfaction with prototyping might be lower than lab studies show, since their expectations of early prototypes are too high; and extensive systems analysis must be done before a prototype is developed (Kieback et al. 1992; Pape and Thoresen 1992).

Researchers have only begun to understand the kinds of methodologies appropriate for certain kinds of systems, contexts or goals. While additional surveys may be informative, the application of specific methodologies to specific problems must be examined in the field to understand the contexts and uses contributing to the success or failure of a methodology.

4.6 Are Methodologies Obsolete?

Not only has the nature of information systems changed dramatically within the last decade, but so too have organizations, industries and the role of information systems — and the changes continue. Methodologies for the twenty-first century have been envisioned as addressing not only new technological forms, but also as including rich semantics and support for identifying and exploiting information technology for strategic advantage (Lyytinen 1989).

Lyytinen (1989) described first generation methodologies as addressing code quality (e.g., structured programming) and second generation methodologies as focusing on the analysis and design of well-structured application domains (e.g., structured methodologies, ISAC). Unfortunately, we still don't know how first and second generation methodologies were used or how well they have worked. Therefore, we cannot know what third generation methodologies should be, or if they are needed at all.

On the one hand, evidence is mounting that the design process is not understood, nor is it rational (Curris, Krasner and Iscoe 1988; Hirschheim and Newman 1991; Turner 1987). We therefore must question the validity of applying methodologies to structure the design process. On the other hand, Lyytinen (1988) concluded that systems analysts view the design process as "rationalistic" and want order and control.

Without a sound empirical base, we cannot conclusively state that existing methodologies are appropriate or inappropriate for today's systems or tomorrow's. Since we have not evaluated the effectiveness of existing methodologies or how they are used, we do not know if existing methodologies are or are not appropriate for today's systems or for "post-modern" organizations (Baskerville, Travis and Trux 1992). The evaluation of existing development practices and methodology usage is clearly necessary to identify any strengths and shortcomings of existing practices before prescriptions for the future can be made.

5. SUMMARY

The dearth of research of methodologies used to develop real systems in real contexts is obvious from the preceding discussion. To date, published work has largely been conceptual (e.g., frameworks) and applied research (e.g., proposing of a new methodology), punctuated by a few laboratory studies, field surveys, practice descriptions, and case studies. In short, methodology knowledge is based on conceptual writings and studies of small systems in unrealistic contexts, augmented by surveys that often compare unspecified methodologies (e.g., comparing prototyping to SDLC use). Although this is a start, research must now focus on the use of methodologies to develop large systems in realistic contexts.

We do not know if, or when, there are critical differences among methodologies under real conditions. We do not understand how methodologies are selected or adapted or how they should be selected — or if they should be selected at all. We don't know how methodologies are used or how effective they are. The questions raised in Table 1
Table 2. Examples of Research Questions and Methods

<table>
<thead>
<tr>
<th>Are methodologies used? Why not?</th>
<th>Case Study</th>
<th>Field Study</th>
<th>Field Survey</th>
<th>Field/lab Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>How are methodologies developed/selected/adapted?</td>
<td>How is the decision made to use/not use a methodology?</td>
<td>What are the most important factors when deciding to use/not use a methodology?</td>
<td>How many organizations use methodologies?</td>
<td>Probably not very helpful here, although experiments addressing other questions would provide some insight.</td>
</tr>
<tr>
<td>Do methodologies work?</td>
<td>How are methodologies developed/selected/adapted?</td>
<td>How important is application type or staff experience in selecting or adapting a methodology?</td>
<td>What are the most important factors in selecting a methodology?</td>
<td>What are the trade-offs in selecting vs. developing or adapting a methodology (eg protocol analysis of case solved by groups of IS pros).</td>
</tr>
<tr>
<td>When are specific methodologies successful?</td>
<td>How do contextual, project-related, and people-related variables affect methodology success?</td>
<td>How successful was Methodology X (according to a given metric) when used on recent projects?</td>
<td>What evaluation metrics do organizations use?</td>
<td>Under controlled conditions, how successful is Methodology X?</td>
</tr>
<tr>
<td>Are methodologies obsolete?</td>
<td>How do contextual factors affect methodology success?</td>
<td>How useful is Methodology X when used to develop realtime systems?</td>
<td>What are managers' opinions about the effectiveness of Methodology X on specified systems?</td>
<td>Which of 2 methodologies used on the same system by two teams, produces the best system, according to a specific metric?</td>
</tr>
<tr>
<td></td>
<td>How well do alternative development approaches work in certain contexts?</td>
<td>How well does Alternative Y work on realtime systems?</td>
<td>What alternatives are used?</td>
<td>Does a team using Methodology X produce a better system than one using an alternative on the same problem?</td>
</tr>
</tbody>
</table>

must be answered. Clearly, the past research paradigm of prescriptive writing and applied research, with limited laboratory and field research on small or unrealistic samples, has been inadequate. Not only are more field studies needed, but qualitative research is crucial. Examples of future research questions are summarized in Table 2. Questions and categories in Table 2 are combined to conserve space. Action research and practice descriptions, while acknowledged to be helpful in answering the questions in Table 1, were also omitted from Table 2 due to space considerations.

Future research may provide a basis for the methodologies of the next century or may show that methodologies — as we know them today — are not beneficial. However, it will provide the basis for future, useful, methodology research.

6. REFERENCES


Burns, R. N., and Dennis, A. R. “Selecting the Appropriate Application Development Methodology.” Data Base, Fall 1985, pp. 19-23.


Loy, P. "The Method Won't Save You (But it can Help)." Software Engineering Notes, Volume 18, 1993, pp. 30-34.


Moran, R. "The Case Against CASE." Information Week, February 17, 1992, pp. 29-32.


7. ENDNOTES

1. Prototyping is problematic. If it is conceptualized as “evolutionary development” — that is, the prototype eventually becomes the working system — then it is a software process model. However, it may also be used as a technique. For example, if user interfaces are prototyped, then prototyping is a requirements or specification technique.