An Empirical Evaluation of System Development Methodologies

By: Prashant C. Palvia and John T. Nosek

Palvia, P. and Nosek, J.T. "An Empirical Evaluation of System Development Methodologies." Information Resources Management Journal, Vol. 3, No.3, Summer, 1990, pp.23-32.

Abstract:
Many methodologies and techniques may be used in the development of information systems. Two widely used methodologies are: system development life cycle, and proto-typing. In this article, we have extensively evaluated the two methodologies based on field data collected from information systems professionals in business and industry. Specifically, the two methodologies are evaluated for their applicability during various life cycle phases, by types of systems, and by the amount of problem structure. Various attributes associated with the methodologies have also been assessed. Our results support the use of a contingency approach for the selection of a proper methodology for systems development.

Article:
There are numerous methods available and used for developing information systems (IS). These methods may be categorized into three classes (DB Directions, 1985): methodologies, which are organized, systematic approaches for handling the system life cycle or its major parts; techniques, which are means of accomplishing specific tasks in the system life cycle; and tools, which are software packages to support one or more techniques. Characteristics, usage, and applicability of the techniques and methodologies have been reported in the literature (Colter, 1984; Couger, Colter, and Knapp, 1982; Doke and Myers, 1986; Gingberg and Ariav, 1986; Necco, Gordon, and Tsai, 1987: Palvia and Nosek, 1989; Vessey and Weber, 1986; Wells and Naumann, 1985). In this article, we focus on methodologies for information system development.

Several methodologies are reported in the literature (Davis, 1982; Davis and Olson, 1985; Gore and Stubbe, 1983; Gremillion and Pybum, 1983; McDonald, Riddle, and Youngblut, 1986; Naumann, Davis, arid McKeon, 1980; Naumann and Jenkins, 1982; Nosek, 1988; Sumner and Sitek, 1986). The most widely used methodology is the system development life cycle (SDLC), which comprises of a sequence of well-defined linear tasks. The tasks are derived by breaking the system's stages (analysis, design, development, and implementation) into phases, activities, and tasks (Davis and Olson, 1985; Necco, Gordon, and Tsai, 1987). In response to various shortcomings of the SDLC approach as well as an ever-growing backlog of IS applications, alternate methodologies are being utilized. According to Gremillion and Pybum (1983), other approaches for system development are: using commercial application software packages, prototyping, and user-developed systems. Of these, using software packages precludes organizational IS development (although, the package developer must have used some methodology), and user-developed systems are usually ad-hoc, simplistic, narrow in scope, and prone to being of poor quality (Davis, 1982). Therefore, any large-scale comprehensive organizational information systems are developed using one of the two methodologies: system development life cycle and prototyping. The predominant use of SDLC and prototyping is also supported by surveys of information systems professionals and organizations (Palvia and Nosek, 1989). Briefly described, prototyping is a four-stage methodology (Naumann and Jenkins, 1982) consisting of: (a) identifying the user's basic information requirements, (b) developing a working prototype system, (c) implementing and using the prototype system, and (d) revising and enhancing the prototype system. Note that may be a number of iterations of the last two stages.

In this article, we focus on the SDLC and the prototyping methodologies. In practice, the system development life cycle may vary (and does vary) from user to user, and organization to organization in terms of the specific details, but the overall character remains essentially the same. In order to ascertain that an organization is using the SDLC approach, we included only those users that used a formal implementation of the SDLC approach, either obtained from a commercial vendor (e.g. SPECTRUM, PRIDE, SDM, CARA) or an in-house implementation of the SDLC. Many of these implementations are generically referred to as application system development methodologies (ASDM); as such we will also refer to it as ASDM. In our view, ASDM and SDLC are synonyms (the only difference is that ASDM is a formal implementation of the SDLC).

The previously cited references discuss the characteristics, strengths, and weaknesses of the two methodologies mainly on the basis of views, opinions, and experiences. Our purpose is to objectively evaluate the two methodologies based on the actual experiences of IS professionals in business and industry.

Objectives

For the two methodologies, ASDM and prototyping, the following objectives are addressed in this article.

1. Assess their utility and applicability during various phases of the system life cycle. Irrespective of the methodology used, there are always activities related to analysis, design, development, and implementation. It is possible that the two methodologies may have different levels of applicability in the various phases. Besides, the literature describes prototyping both as a methodology and as a technique for various activities during the life cycle phases.

2. Assess the utility and applicability of the methodologies for different system types, namely: operational systems, management information systems, decision support systems and strategic information systems. In this work, strategic systems do not specifically refer to systems for competitive advantage, but to a broader range of systems designed to support the strategic/executive level activities.

3. Assess the utility and applicability of the methodologies for structured and unstructured system problems.

4. Determine the perceived values of the attributes associated with the methodologies. The following attributes/dimensions (based on prior literature, experience and pilot tests) are considered for each method:

- Project control (i.e. how much does a method facilitate project control)
- Cost of Usage
- Ease of Use
- Ease of Learning
- Communicability to End User
- Communicability to Data Processing Personnel
- Flexibility of Design Produced
- Early Discovery of Problems
- Leading to Maintainable Systems
- Quality of Generated Documents

We realize that some of these attributes may be correlated; but we include all of them as they are fairly well understood. Also, we have not included "productivity" directly as an attribute; it is a consequence of the other factors that are included.

Figure 1: Types of Organizations Represented in the Study

Computers, DP, Communications:	5.2 %
Instruments & Electrical:	3.4 %
Printing and Publishing:	6.9 %
Food and Tobacco:	3.4 %
Primary and Fabricated Metal:	3.4 %
Insurance and Banking:	6.9 %
EDP Services:	15.5 %
Education:	3.4 %
Government and Public Utility:	6.9 %
Transportation:	5.2 %
Consultants:	6.9 %
Hotels:	3.4 %
Other:	29.5 %

Research Methodology

The research methodology is based on empirical data collected from MIS professionals working in different types of organizations located in northeastern and midsouth metropolitan areas of the United States. While the research addressed many issues beyond the scope of this article, the methodological steps relevant to this article are:

1. A questionnaire was developed to address the study objectives and related issues. The questionnaire consisted of several sections addressing different issues. One of the sections was devoted to the use and applicability of the various methods under different conditions (as specified under the "objectives"). Another section addressed the attributes associated with the methods. Finally, the questionnaire contained questions about demographic information about the respondents and their organizations.

2. The questionnaire was pilot-tested with MIS graduate students with prior data processing experience, peer faculty members and MIS professionals from industry. As per the feedback from the pilot-testing, the questionnaire was refined and finalized.

3. The questionnaire was administered to data processing professionals in two large metropolitan areas of the United States (northeast and midsouth). The *survey* sample included members of data processing/computer organizations (DPMA and ACM) as well as direct contacts established by the researchers.

4. Following the return of completed questionnaires, the data was analyzed using statistical methods. The results included in this paper are primarily based on descriptive statistics.

Results

Before reporting the "methodology" results, we briefly describe the profile of the respondents as well as the profile of the organizations they represent.

Respondent and Organization Profile

The survey questionnaires were mailed to 300 data processing professionals, out of which 65 usable responses were returned. This constitutes a 22 % rate of return. The individuals responding represented a wide spectrum of organizations. Figure 1 shows a breakdown of types of organizations represented in the sample. Most industries are included in the study. EDP services segment has a slightly higher proportion, as many of the DP professionals work in EDP services. Some other interesting characteristics of these organizations are: a. the data processing department is centralized in 74% cases, decentralized in 22% cases, and a combination approach used in 4% cases; b. for all organizations combined, the reported DP budget as a percent of total budget has an average of 2.4% and a median of 4%; and c. most data processing directors report to the CEO (54%), while 29% report to the vice president of finance and/ or accounting.

Figure 2: Annual Sales of Represented Organizations

Amount	Percent
$100,000-500,000:	5.8 %
$500,000 - 1 million	1.9 %
$1 million- 10 million:	17.3 %
$10 million-100 million:	51.9%
$100 million - 1 billion:	13.5 %
$1 billion - 5 billion:	5.8 %
Over $5 billion:	3.8 %

Figure 3: Data Processing Budget of Represented Organizations

Amount	Percent
$10,00 - 50,000	11.8 %
$50,000 - 100,000	9.8 %
$100,000 - 500,000	13.7 %
$500,000 - 1 million:	33.3 %
$1 million - 2 million:	5.9 %
$2 million - 5 million	11.8 %
$5 million - 10 million:	3.9 %
Over $10 million:	9.8 %

As regards to the size of the organizations surveyed, Figures 2 and 3 summarize the annual sales and annual DP budgets of the organizations. Again a wide range is represented. The median annual sales are in the range of 10 million to 100 million dollars, and the median DP budget is in the range of 500 thousand to 1 million dollars. The number of employees in the organization range from 2 to over 10,000 with a median of 600 employees, and the number of data processing employees range from no additional to several hundred with a median of 12 employees. In essence, the data collected in this research study represents diverse backgrounds of individuals as well as organizations.

Table 1: System Life Cycle Stages

ANALYSIS
- Study present system
- Requirements definition
- Feasibility study

DESIGN
- Preliminary design
- Detailed design

DEVELOPMENT
- Program design
- Programming and coding
- Testing

IMPLEMENTATION
- Installation
- Operation

Use and Applicability of Methods During System Life Cycle Stages

By definition, die ASDM approach is designed to address all stages of the life cycle; while prototyping may have more value in the earlier stages. However, the two methodologies may not have the same degree of applicability or usage in the life cycle stages. Our instrument asked the respondents to indicate the life cycle stages in which a method was being used in their organization; and if it was not being used, whether it was applicable for the stage. There were very few instances where respondents felt that the method was applicable and was not being used. For this reason, we combine the use and applicability results and report it as "use and applicability" of the methods.

Four stages of system development life cycle were considered. Each stage was further divided into sub-phases. The stages and sub-phases are shown in Table 1. We present the results for each stage.

The results are shown in Figure 4. In this figure (and all subsequent figures), the percent-age refers to the percent of respondents that use the method for a given purpose, of all the respondents that use the method (for any purpose). As can be seen, ASDM has high use and applicability during all phases of the system life cycle. This is expected as the methodology is well-established and includes specific activities and milestones to address each phase. On the other hand, prototyping has higher applicability in the analysis and design phases, and less during development and implementation. Somewhat surprising, more practitioners found prototyping useful for design than analysis (64.3% vs. 50%).

Use and Applicability of Methods for Different System Types

As stated earlier, four types of systems have been considered. Operational information systems (OIS) are those whose primary function is to support the transactional and paper work activities of the organization. Management information systems (MIS) additionally provide information (in the form of reports, screens, interactive capability, etc.) for managerial decision making. Decision support systems (DSS), in addition to information, provide active sup-port for decision-making. In our work, strategic information systems (SIS) include systems which meet the strategic needs of top-level management (they are more popularly known as executive information systems).

Figure 4: Methodology Use During System Life Cycle

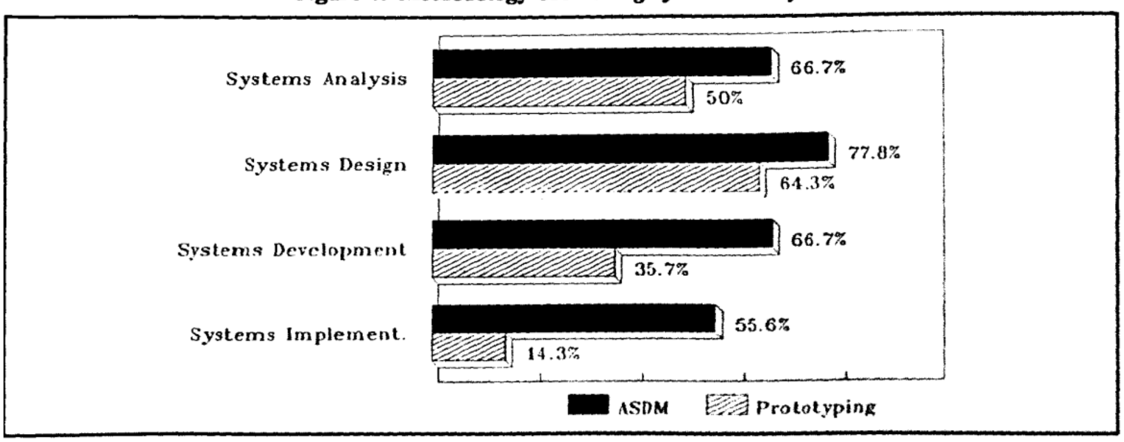


Figure 5: Methodology Use for Different System Types

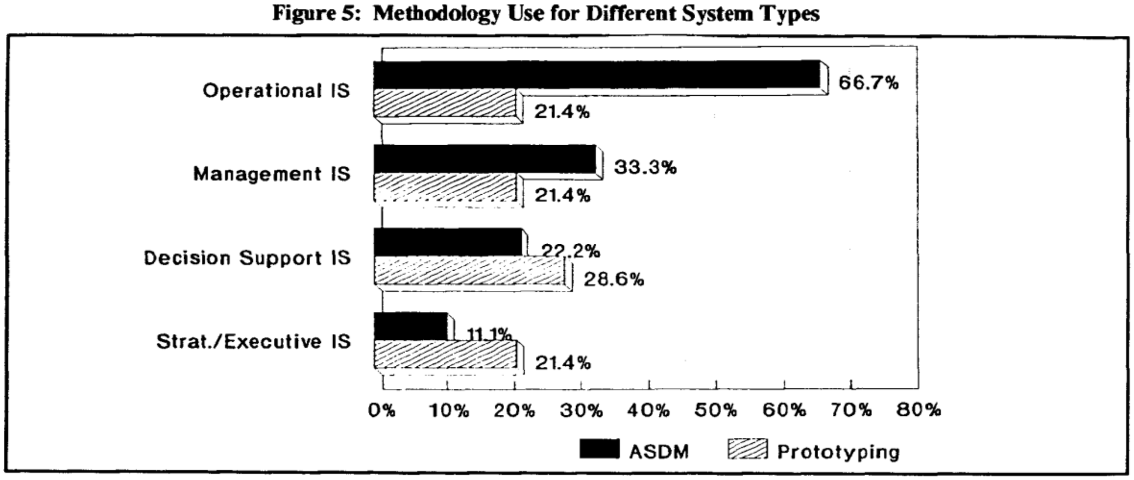
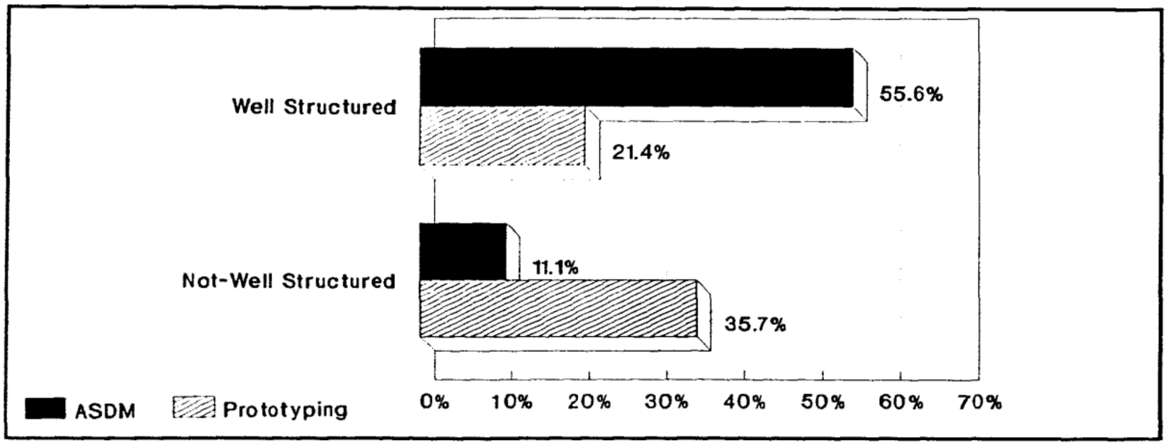


Figure 6: Methodology Use Based on Problem Structure

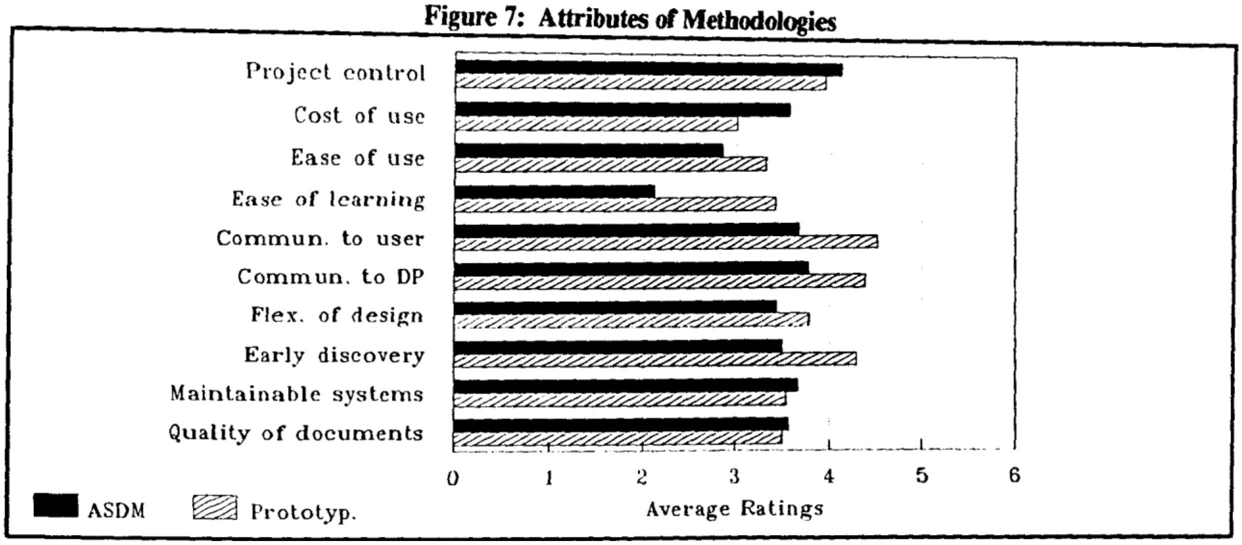


The use and applicability of the ASDM and prototyping methodology are shown in Figure 5. One pattern is readily discernible from these results. The ASDM methodology is most suitable for OIS, and progressively less and less applicable for MIS, DSS, SIS. As for the prototyping methodology, a clear pattern like this is not apparent. The applicability of proto-typing is about the same for all system types, except for DSS where it is slightly superior in its applicability. Also, the results indicate (corroborating the literature) that ASDM is more suited to the development of OIS and MIS, while proto-typing is preferred for DSS and SIS.

Use and Applicability of Methods Based on Problem Structure

The problem structure was considered at two levels: well structured, and not-well structured. The applicability of the two methodologies: ASDM and prototyping, is shown in Figure 6. The results again reinforce proclamations by experts and authors in the field (Gore and Stubbe, 1983; Necco, Gordon, and Tsai, 1987). The ASDM approach is more suitable for structured problems, while prototyping is more suitable for unstructured problems. In fact, this also explains the applicability of the methods for the different types of systems discussed earlier. A major characteristic distinguishing OIS and MIS from DSS and SIS is the degree of problem structure. Operational information systems and management information systems are designed for more structured problems than are decision support systems and strategic/executive information systems. More on this in the next section.

Figure 7: Attributes of Methodologies



Comparison of Problem Structure and System Type

We wished to investigate whether there is a correspondence between well structured and OIS/MIS systems, and not-well structured and DS S/SIS systems. Figures 5 and 6 do indeed show high correspondence for methodology use for well structured problems and OIS/MIS systems, in-fact the use and applicability of the prototyping methodology is an exact match for well structured problems, OIS and MIS. Although there is not as close a correspondence for not-well structured problems and DS S/SIS, the trends are in the right direction.

What this means is that while the respondents may not consciously consider the structure of the problem domain, they certainly understand that DSS/SIS are of the type that requires a more iterative methodology (a requirement for not-well structured problem domains).

Perceived Attribute Values of the Methodologies

For the two methodologies, we asked the respondents to rate them on several attributes (the attributes were listed earlier). We present these results below for each attribute. They are summarized in Figure 7.

1. Project control. Project control refers to how much a technique or methodology contributes to the overall control of the system project. Each method was rated on a 1-5 scale, with 1 being poor control and 5 being good control. On this criterion, ASDM was rated higher than prototyping (average ratings being 4.111 and 3.929).

2. Cost of Use. Again each method was rated on a 1-5 scale, with 1 being low cost of use and 5 being high cost of use. Of the methodologies, prototyping was rated as being less costly to use (average rating of 3.000) than ASDM (average rating of 3.556).

3. Ease of Use. This question was asked to deter-mine the ease of use of the method without regard to the availability of automation tools; the intention was to get a rating based on *the* method alone. The ratings were on a 1-5 scale, with 1 being "hard to use" and 5 being "easy to use". As per these ratings, prototyping is considered easier to use than ASDM (average ratings are 3.308 and 2.856).

4. Ease of Learning. Again these ratings are reported without regard to use of automated tools. On a five-point scale, a rating of 1 is for "hard to learn" and 5 is for "easy to learn". On this criterion, prototyping is much easier to learn than ASDM (ratings are 3.429 and 2.111).

5. Communicability to End User. One of the primary objectives of many methods is to improve the communication and understanding provided to users as well as DP analysts. The "communicability" to users was rated on a 1-5 scale, 1 being poor communicability and 5 being very good communicability. Again, not surprisingly, prototyping is perceived superior than ASDM on this aspect (their average rating are 4.5 and 3.667).

6. Communicability to Data Processing Personnel. On the "communicability to data processing professionals" dimension, again prototyping is rated higher than ASDM although not as much as "communicability to users" is rated higher. The average ratings for prototyping and ASDM were 4.385 and 3.778.

7. Flexibility of Design Produced. "Flexibility of design produced" is a desirable property during development, especially if the problem is unstructured, and the users and designers had little experience with such systems. It is also desirable during maintenance, as maintenance can be a significant resource consuming activity (Davis and Olson, 1985; Gore and Stubbe, 1983). This dimension was rated on a 1-5 scale, with 1 being inflexible design and 5 being flexible design. Prototyping is perceived as more flexible in this regard than ASDM (average ratings of 3.786 and 3.444).

8. Early Discovery of Problems. Information system development consumes extensive resources, both financial and personnel. The literature contains many instances of poorly designed systems, and the ensuing problems (Davis, 1982). The costs of redesigning systems and maintenance are significant. Any methods that can lead to early discovery of problems, should alleviate the situation to some extent. Respondents were asked to rate this characteristics of the methods on a 1-5 scale, 1 being "poor" early discovery and 5 being "good" early discovery. Again, without surprise, prototyping is rated much higher than ASDM (ratings of 4.286 and 3.500).

9. Leading to Maintainable Systems. As was stated earlier, the costs of maintenance are significant. Many estimates (Davis and Olson, 1985; Gore and Stubbe, 1983) have put these costs to be over half the total costs

incurred during the developmental and operational life of a system. It has been argued that structured techniques hold great promise in alleviating maintenance problems. The ratings indicate that ASDM has a slight edge over prototyping in this regard (average ratings are 3.667 and 3.538).

10. Quality of Generated Documents. This characteristic represents the overall quality of the product produced by the method. The method in use produces a product/document for which it was intended for. The product may represent significant portions of the information system, or specific pieces of it. The ratings of the two methodologies, ASDM and prototyping are fairly close on this criterion (3.556 and 3.500).

A final comment on the attributes. There may be a dangerous tendency to treat all of the above attributes equally, while that is certainly not the case. For a given environment and a given systems project, these attributes will assume different levels of importance. Only the systems/ project manager will be able to assess the relative importance of the attributes. Given the relative importance of the attributes, our results will be a good guide in methodology selection.

Conclusions

Based on field data, this article reports on the characteristics of two major development methodologies (system development life cycle methodology, called ASDM here, and proto-typing). Results are also presented on the use/ applicability of the methodologies during various phases of systems development and different types of systems and systems problems.

The major contribution of this paper is its support for the contingency approach for methodology selection (Grernillion and Pyburn, 1983; Naumann, Davis, and McKeen, 1980). From our results, we offer the following model of method selection: a. first make a preliminary selection of methodology based on information system type and problem structure. b. second, finalize the selection based on the needed attributes. Note that the attributes are influenced by organizational considerations, environmental conditions, and project characteristics.

Some related observations, not borne out by the above results, but emerging from the larger study are:

1. Just as methodologies are selected on a contingency basis, the specific techniques should also be made using a contingency model. However, the techniques selection should be made after the methodology has been selected.

2. There is a special observation concerning the use of methodologies. While the two methodologies: ASDM and prototyping, are applicable for the entire life cycle, they have varying degrees of applicability during specific life cycle phases. Our results have demonstrated this, and confirm the MIS literature on prototyping, where prototyping has been described both as a methodology and as a technique to be *used* in specific phases.

3. Developers do not seem to be making major distinctions between operational information systems and management information system, and also between decision support systems and strategic/executive information systems. While the systems may differ in their intended purpose and audience, their development characteristics seem to be similar.

4. Finally, it appears that practitioners still do not explicitly treat problem structure as advocated in the MIS literature; and that they are confused with its meaning when asked in a direct fashion. However, it is obvious that practitioners sense some differences in DSS/SIS/EIS from OIS/ MIS, and that they apply methods to these types of systems as advocated for not-well structured problem domains.

References

Colter, M.A. (1984). A Comparative Examination of Systems Analysis Techniques. *MIS Quarterly. 8(I)*

Connor, D. (1985). *Information System Specification & Design Road Map.* Prentice-Hall.

Gauger, D., Colter, M. and Knapp, R. (i982). *Advanced System Development, Feasibility Techniques. John* Wiley *&* Sons.

Davis, G.B. (1982). Claim: User-Developed Systems *can* be hazardous to your organization. *Proceedings of 17th Annual Hawaii Conference on System Sciences.*

Davis, G.B. and Olson, M.H. (i985). *Management Information Systems: Conceptual Foundation:, Structure, and Development.* New York: McGraw-Hill.

DB Directions 1985. (1986). Kahn, B.K. and March, ST. IILM and the System Life Cycle. Section on Methodologies, Tools and Techniques, by Cline, J.E., Ketabdi, M.A. , Ludt, J., Pelvis, P. and Olle, T.W. Databese Engineering, June, 43-48.

Doke, E.R. and Myers, LA. Jr. (1986). A Matrix Approach to the Analysis of Productivity Aids in the System Development life Cycle." *Proceedings Nat'l DS1 conference*, Hawaii.

Gingberg, MJ. and Ariav, G. (1986). *Methodologies for DSS Analysis and Design: A Contingency Approach to Their Application.* ICIS conference, SIM Diego.

Gore, M. and Stubbe, J. (1983). *Elements of System Analysis.* WCB. 3rd.

Gremillion, LL and Pybum, P. (1983). Breaking the Systems Development Bottleneck. *Harvard Business Review.* March-April.

Guinsams, T. A Study of Application Program Development Techniques." Communications of the ACM, vet 28, no. 5.

Martin,1. (1985). Fourth-Generation Languages, Vol. 1, Prentice-Hall

McDonald, CW., Riddle, W. and Youngblat, C (1986)- Start Methodology Area Summary. Vol II: Preliminary Views on the Software Life Cycle and Methodology Selection. *ACM Software* Eng. Notes. 2(2) April.

Naumann, J.D., Davis, G.B. and Mckeen, J.G. (1980). Determining Information Requirements: A Contingency Method for Selection of a Requirements Assurance Strategy. *The Journal of Systems and Software, vol* i.

Naumann, J.D. and Jenkins, A.M. (1982). Prototyping: The New Paradigm for Systems Development. *MIS Quarterly,* 6(3).

Necco, C.R., Gordon, C.L., and Tsai, N.W. (1987). Systems Analysis and Design: Current Practices. *MIS Quarterly,* December.

Nosek, IT. (1988). Contingency Approaches to IS Development Methodology Selection: Current Practices and Future Directions. *Proceedings of the National DSI Conference,* Las Vegas.

Palvia, P. and Nosek, J.T. (1989). A Comprehensive Assessment of System Life Cycle Techniques and Methodologies. *Proceedings of the National DM Conference,* New Orleans.

Sumner, M. and Sitek, J. (1986). Are Structured Methods for Systems and Analysis and Design Being Used? *Journal of Systems Management.* June.

Vessey, I. and Weber, R. (1986). Structured Tools and Conditional Logic: An Empirical Investigation. *Communications of the ACM.* 29(1).

Wells, C.E. and Naumann, J.D. (1985). An Empirical Comparison of User Understandability: System Flow Charts and Data Flew Diagrams. *Proceedings National DSI conference.* Las Vegas.