A field examination of system life cycle techniques and methodologies

By: Prashant Palvia and John T. Nosek

Palvia, P., and Nosek, J.T. "A Field Examination of System Life Cycle Techniques and Methodologies." Information & Management, Vol 25, 1993, pp. 73-84.

Abstract:
There is a myriad of system development methodologies, techniques, and tools that have been developed ever since businesses started using computers for information processing. However, there is no real consensus on the aptness or applicability of these methods. There are proponents and critics of each method, and different organizations use different methods. This article identifies many of the system development methods in use today. It, then, empirically assesses these methods on a comprehensive set of attributes. It also makes a contingency analysis of the applicability of the methods based on system life cycle stage, system type, and problem structure. These results should serve as a valuable set of inputs in method selection for system development.
Keywords: System development life cycle; System development methodologies; System development techniques; System development method use; System development method evaluation; System type; Problem structure

Article:
Introduction[1]
Many methodologies, techniques and tools (termed "methods" here) exist and continue to be developed [e.g., see 1,3,9,10,14] for accomplishing various tasks in the system life cycle (SLC). There are strong proponents and critics of each method; however, there is limited research concerning their effectiveness. Studies usually compare a few methods (often two or three) on a limited set of attributes [10,11,13,18,23,24].

Designers, analysts, programmers, end users, and MIS managers are confronted with a wide array of methods and are poorly informed on their utility and desirability. Therefore, a comprehensive approach is essential to understand the attributes and appropriateness of these methods. Elements of such an approach were outlined in the Data Base Directions workshop of 1985 [5]. One comprehensive approach that was pursued was the STAR methodology [16], in the context of the ADA environment for the Department of Defense. Another approach by Doke and Myers [6] focused on the overall productivity aspects of the methods. An important contribution is by Colter [1], where he subjectively evaluated characteristics of various tools and techniques. With respect to DSS methodologies, Ginzberg [7] subjectively evaluated hem and presented a contingency model for their use.

The literature does not present a comprehensive evaluation based on field experience. It is important to know how these methods fare with their users (i.e., IS professionals). Here, we examine a comprehensive set of techniques and methodologies from a field perspective. First, in order to clarify the distinction between the terms, we adopt the following definitions from [5]:

Methodology: is an organized and systematic approach for handling the system life cycle or its parts. It will specify the individual tasks and their sequence.

*Technique:*is a means of accomplishing a task in the system life cycle (SLC). Sometimes, it may become synonymous with the task.

Tool: is a computer software package to support one or more techniques.

We have deliberately limited the scope of this work to techniques and methodologies, as it is a relatively stable set, while tools continue to proliferate rapidly. Therefore, the evaluation of CASE (Computer Aided Software Engineering) tools is outside our scope.

This article identifies the major existing techniques and methodologies for information systems development and reports their use in organizations. Based on data collected from IS professionals, it examines several attributes of the various IS methods and evaluates their applicability under different circumstances.

Research objectives

The following are the objectives of this study:

1. Identify a comprehensive set of available techniques and methodologies (T&M) for IS development.

2. Determine the T&M currently in use and their level of use, by IS professionals.

3. For the T&M in use, determine the perceived values of their attributes. The following attributes/ dimensions (based on prior literature, experience, and pilot tests) have been considered:

- Project control (i.e., is it facilitated?) Cost of use
- Ease of use
- Ease of learning
- Communicability to end user
- Communicability to IS personnel
- Flexibility of design produced
- Early discovery of problems
- Leading to maintainable systems
- Quality of generated documents

Note that some of these attributes may be correlated. Further, we have not used "productivity" as a separate attribute, as it is a consequence of these factors.

4. Assess the utility and applicability of the T&M during various phases of the system life cycle.

5. Assess the utility and applicability of the T&M for different system types, i.e., operational IS, MIS, DSS, and strategic IS. Strategic IC refer to both systems for competitive advantage and a broader range of systems designed to support strategic/ executive level activities. Currently, the latter systems are labeled as executive IS.

6. Assess the utility and applicability of the T&M for structured and unstructured system problems.

Research methodology

The research methodology is based on field data collected from MIS professionals working in different types of organizations in northeastern and midsouth metropolitan areas of the United States. Specifically, the methodology includes:

1. Developing a comprehensive list of available techniques and methodologies through a literature search [2,3,4,8]. This encompassed several generations of methods, but it specifically excluded tools. Minor adjustments were made based on experience and comments from peers. Note that since the conduct of research and until the publication of results, some methods have become obsolete while some new ones have emerged.

2. Developing a questionnaire to address the study objectives. There were three major sections: the first contained questions about the level of use of the T &M; the second on the methods' perceived attribute values;

and the third on the applicability of the methods by life cycle phases, different system types, and degree of problem structure. The questionnaire also included demographics related questions.

3. Pilot-testing the questionnaire with a total of ten respondents that included M.B.A. students, peer faculty members and MIS professionals from industry. The questionnaire was modified and finalized as a result.

4. Mailing the questionnaires to IS professionals in two large metropolitan areas of the United States (northeast and midsouth). The survey sample included those having memberships in two professional organizations (DPMA and ACM).

5. Finally, analyzing the data from the completed questionnaires using a statistical package.

Table 1
Inventory of software methods.

Process Flow Chart
Forms Flow Chart
Systems Flow Chart
HIPO (VTOC Chart)
HIPO (IPO Chart)
Data Flow Diagram
Automated Flowcharting
PSL/PSA

Decision Table
Functional Description
Structure Chart (Yourdon)
Gridcharting
Jackson Chart
Warnier-Orr Diagram
Nassi-Schneiderman Chart
Data Base Dictionary
SODA (Optomization Aid)

Program Flow Diagram
Pseudocode
Structured English
Chapin Chart
Decision Table Processor
Structured Programming
HOSKYNS System

Test Data Generator
Structured Testing
Higher Order Software (HOS) by Hamilton/Zeldin/Martin
Fourth Generation Language, e.g., FOCUS, RAMIS

Structured Analysis as described by DeMarco
ADS System (by NCR)
BIAIT (Bus. Info. Anal. and Integration Technique)
BICS (Business Info. Characterization Study)
SREM (Sys. Req. Engg. Methodology by Mack Alford)

BISAD (by Honeywell)
SOP (IBM's Study Organization Plan)
Automated ADS
Information Algebra
Young/Kent Methodology
Langefors Methodology
Information Engineering by Finkelstein and Martin

Structured Design by Yourdon and Constantine
Structured Systems Design described by Orr
SYSTEMATICS
SADT by SofTech
Jackson System Development by Michael Jackson

PLEXSYS (Workbench Approach)
ISDOS (University of Michigan Approach)
Prototyping
Applic Sys Dev Methodology (e.g. SPECTRUM, PRIDE, CARA)

Structured Walkthrough
Cost Benefit Analysis
Quality Assurance
Independent Software Audit

Results

1. Available techniques and methodologies

A comprehensive list of available techniques and methodologies, as developed from literature search is shown in *Table 1*. The compilation first lists techniques (roughly organized according to system life cycle stages), then methodologies, and finally some general techniques. This list includes some old and primitive methods as well as more advanced ones; the objective was to capture a comprehensive range. This served as the basis for collecting the empirical data. As stated earlier, some of the techniques may have become obsolete and some new ones have appeared by the time of publication of this article. Further, the distinction between a task and a technique may not be clear at times.

The remaining study objectives are addressed by using the field data collected from the IS professionals.

2. Profile of responding individuals and their organizations

The questionnaires were mailed to 300 IS professionals, 65 usable responses were returned, a 21.7% rate of return. The sample represents a wide spectrum of organizations. *Table 2* shows that most industries are included in the sample. The EDP services segment has a slightly higher proportion, as many of the IS professionals work in it. Some other interesting characteristics are:

Table 2
Types of organizations represented in the study.

Computers, DP, Communications	5.2%
Instruments and 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%

Table 3
Annual sales of represented organizations.

Amount	Percent
$100,000–500,000	5.8
$500,000–1 million	1.9
$1 million–10 millions	17.3
$10 million–100 millions	51.9
$100 millions–1 billion	13.5
$1 billion–5 billion	5.8
Over $5 billions	3.8

a. the IS department was centralized for 74% of the respondents, decentralized for 22%, and a combination for the remaining 4%; b. for all organizations, the reported IS budget was 2.4% of the total budget with a median of 4%; and c. most IS directors reported to the CEO (54%), while 29% reported to the vice president of finance and/or accounting.

Table 4
Data processing budget of represented organizations

Amount	Percent
$10,000–50,000	11.8
$50,000–100,000	9.8
$100,000–500,000	13.7
$500,000–1 million	33.3
$1 million–2 millions	5.9
$2 millions–5 millions	11.8
$5 millions–10 millions	3.9
Over $10 millions	9.8

As regards the size of their organizations, *Tables 3* and *4* show the annual sales and IS budgets. A wide range is represented. Median annual sales were in the range of $10 to $100 million, while the median IS budget was in the range of $500,000 to $1 million. The total number of employees ranged from 2 to over 10,000 with a median of 600 employees, and the number of IS employees ranged from none to several hundred with a median of 12.

In essence, the data in this study represents diverse backgrounds of individuals and organizations.

3. Methods in use and their level of use

Prior studies have reported the use of selected methods [e.g., 19,21], but our objective was to determine use across a comprehensive range of methods. Respondents were asked to identify the use (or non-use) of a method under one of six categories. The first two divided use into "widely used" and "occasionally used". The last four divided an unused method into: "considered using", "unused, but understood", "generally aware", and "totally unfamiliar". We show the breakdown of usage patterns in *Table 5*.

Using a 10% cutoff on widely used methods, sixteen are identified as more widely used methods. Using a 15% cutoff on the total of widely and occasionally used methods, twenty three methods are identified. We found, by interviewing selected IS professionals, that many respondents were interpreting process flow chart and forms flow chart as a kind of general flow chart (e.g., systems and program flow chart), and not as described here. We, therefore, exclude them. Also, we combine the VTOC part and the IPO part of the HIPO technique into one (HIPO). This results in twenty methods.

Four of the twenty methods may not be explicitly called techniques; e.g., cost—benefit analysis and software audit are normally regarded as tasks, quality assurance is regarded as a philosophy or emphasis of importance throughout the life cycle, and fourth generation language may better be termed as a programming language.

Table 5
Use of software methods.

Method	Used			Not used
	Widely used	Occasionally used	Total	
Process Flow Chart	23.1	32.3	55.4	44.6
Forms Flow Chart	4.6	30.8	35.4	64.6
Systems Flow Chart	40.0	30.8	70.8	29.2
HIPO (VTOC Chart)	7.7	10.8	18.5	81.5
HIPO (IPO Chart)	9.2	9.2	18.4	81.6
Data Flow Diagram	26.2	27.7	53.9	46.1
Automated Flowcharting	0.0	10.8	10.8	89.2
PSL/PSA	3.1	0.0	3.1	96.9
Decision Table	4.6	24.6	29.2	70.8
Functional Description	38.5	23.1	61.6	38.5
Structure Chart (Yourdon)	3.1	12.3	15.4	84.6
Gridcharting	0.0	3.1	3.1	96.9
Jackson Chart	0.0	1.5	1.5	98.5
Warnier-Orr Diagram	0.0	3.1	3.1	96.9
Nassi-Schneiderman Chart	0.0	2.8	2.8	97.2
Data Base Dictionary	38.5	15.4	53.9	46.1
SODA (Optomization Aid)	1.5	1.5	3.0	97.0
Program Flow Diagram	30.8	32.3	63.1	36.9
Pseudocode	23.1	35.4	58.5	41.5
Structured English	9.2	20.0	29.2	70.8
Chapin Chart	3.1	3.1	6.2	93.8
Decision Table Processor	1.5	1.5	3.0	97.0
Structured Programming	53.8	15.4	69.2	30.8
HOSKYNS System	3.1	3.1	6.2	93.8
Test Data Generator	7.7	9.2	16.9	83.1
Structured Testing	10..8	16.9	27.7	72.3
Higher Order Software (HOS) by Hamilton/Zeldin/Martin	1.5	0.0	1.5	98.5
Fourth Generation Language, e.g. FOCUS, RAMIS	32.6	13.0	45.6	54.4
Structured Analysis as described by DeMarco	4.6	9.2	13.8	86.2
ADS System (by NCR)	0.0	0.0	0.0	100.0
BIAIT (Bus. Info. Anal. and Integration Technique)	0.0	0.0	0.0	100.0
BICS (Business Info. Characterization Study)	0.0	0.0	0.0	100.0
SREM (Sys. Req. Engg. Methodology by Mack Alford)	0.0	0.0	0.0	100.0
BISAD (by Honeywell)	0.0	1.5	1.5	98.5
SOP (IBM's Study Organization Plan)	0.0	0.0	0.0	100.0
Automated ADS	0.0	0.0	0.0	100.0
Information Algebra	0.0	0.0	0.0	100.0
Young/Kent Methodology	0.0	0.0	0.0	100.0
Langefors Methodology	0.0	0.0	0.0	100.0
Information Engineering by Finkelstein and Martin	0.0	0.0	0.0	100.0
Structured Design by Yourdon and Constantine	3.1	9.2	12.3	87.7
Structured Systems Design described by Orr	0.0	1.5	1.5	98.5
SYSTEMATICS	0.0	1.5	1.5	98.5
SADT by SofTech	0.0	0.0	0.0	100.0

Table 5 (continued)

Method	Used			Not used
	Widely used	Occasionally used	Total	
Jackson System Development by Michael Jackson	0.0	1.5	1.5	98.5
PLEXSYS (Workbench Approach)	0.0	0.0	0.0	100.0
ISDOS (University of Michigan Approach)	0.0	0.0	0.0	100.0
Prototyping	15.4	16.9	32.3	67.7
Applic Sys Dev Methodology (e.g. SPECTRUM, PRIDE, CAR	12.3	9.2	21.5	78.5
Structured Walkthrough	12.3	26.2	38.5	61.5
Cost Benefit Analysis	30.8	26.2	57.0	43.0
Quality Assurance	29.2	20.0	49.2	50.8
Independent Software Audit	10.8	15.4	26.2	73.8

Excluding these, we are left with sixteen methods as shown in *Table 6.* Of these, "functional description" to most respondents implied some form of narrative description. The last one in this list (ASDM) is a methodology. Prototyping, while largely known as a methodology [17], can also be considered a technique (during the requirements phase).

Seven methods are used by more than half of the IS professionals. One startling and somewhat disturbing observation is that many methods are used very little. Twenty six methods, or half of those extracted from literature, are used by less than 10% of the respondents; thirteen were not used at all (ADS system, BIAIT, BICS, SREM, SOP, automated ADS, information algebra, Young/ Kent methodology, Langefors methodology, information engineering, SADT, PLEXSYS, and ISDOS).[2] Plausible explanations for the low use are: some methods are outdated, some are too complicated, some were never fully developed and proven, while others are too academically and research oriented. Another surprise was the low use of Warnier—Orr diagrams, and Nassi-Schneiderman charts, in spite of their appearance in most text books on systems analysis and design.

For better insights and understanding, the method evaluation results are grouped under two categories: methodologies (comprised of ASDM and prototyping), and techniques (consisting of the other fourteen methods).

4. Evaluation of methodologies

Use and Applicability During System Life Cycle Stages. By definition, the ASDM approach is designed to address all stages of the life cycle, while prototyping may have more value in the earlier stages. Further, a methodology may not be uniformly applicable during the life cycle stages. Our instrument asked the respondents to indicate where the method was used by them in the life cycle, and if not used, whether it seems applicable in some stage. In only a few instances, was a method considered applicable if it was not being used. Consequently, the "use and applicability" results have been combined.

Four stages of the system development life cycle have been considered, as shown in *Table 7.*

In *Figure 1* (and all subsequent ones), a percentage refers to the ratio of the number of respondents that use the method for a given purpose to the number of all 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 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. Somewhat surprisingly, prototyping is considered more useful for design than analysis.

Table 6
Widely used system life cycle methods.

Method	Percent using
System flow chart	70.8
HIPO chart	18.5
Data flow diagram	53.9
Decision table	29.2
Functional description	61.6
Structure chart	15.4
Data base dictionary	53.9
Program flow diagram	63.1
Pseudocode	58.5
Structured English	29.2
Structured programming	69.2
Test data generator	16.9
Structured testing	27.7
Structured walkthrough	38.5
Prototyping	32.3
ASDM	21.5

Table 7
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 for different system types. The use and applicability of ASDM and prototyping for different types of systems are shown in *Figure 2.* One pattern is readily discernible: ASDM methodology is most suitable for OIS, and progressively less applicable for MIS, DSS, and EIS. For prototyping, no pattern is apparent. Only, the applicability of prototyping in DSS is slightly higher.

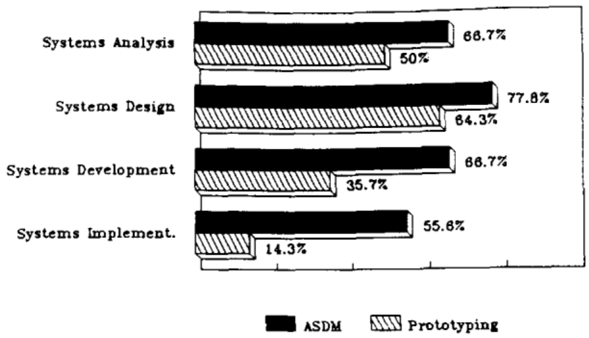

Fig. 1. Methodology use during system life cycle.

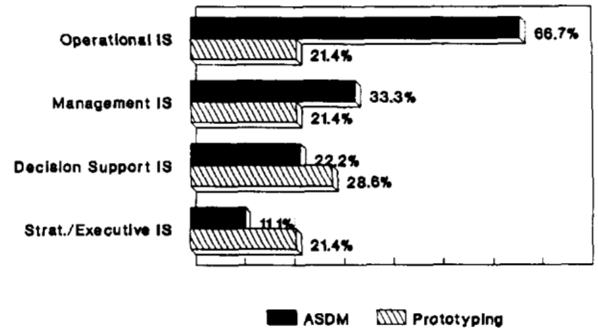

Fig. 2. Methodology use for different system types.

Use and applicability based on problem structure. The problem structure was considered at two levels: well-structured and not-well structured problems. The applicability results are shown in *Figure 3.* The ASDM approach is more suitable for structured problems, while prototyping is better for unstructured ones. A major characteristic distinguishing OIS, MIS, DSS, and EIS is the degree of their problem structure. Generally speaking, OIS and MIS address more structured problem domains, as such the applicability of ASDM is higher for them.

Perceived attribute values. The two methodologies were evaluated by the respondents on the set of attributes listed earlier using a 1-5 Likert scale. The results have been converted so that a low number represents an unfavorable view of the attribute. These results are summarized in *Figure 4.*

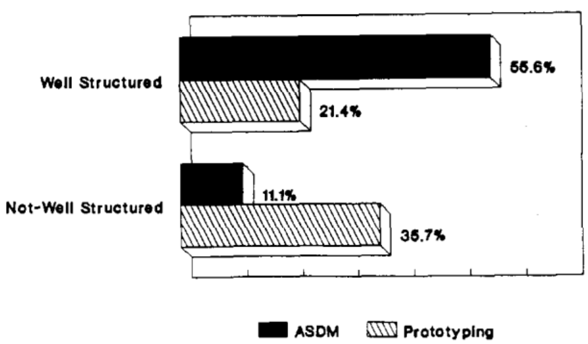

Fig. 3. Methodology use based on problem structure.

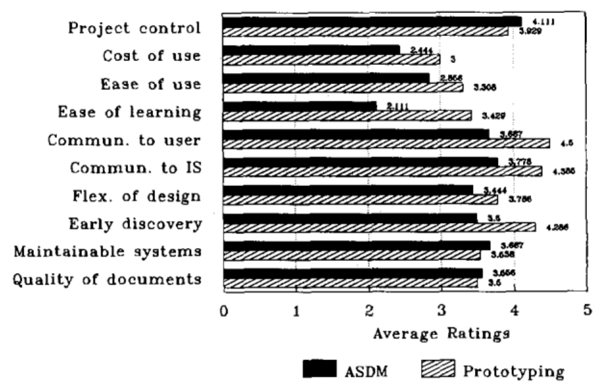

Fig. 4. Attributes of methodologies.

In terms of statistically significant differences in attribute ratings, "ease of learning" and "communicability to end user" are significant (p values of 0.003 and 0.045); and "ease of use" and "early discovery of problems" are marginally significant (p values of 0.121 and 0.113). In all of these, prototyping has more favorable ratings than ASDM. Two possible interpretations of these four attributes are: (i) they truly reflect the most important attributes, or (ii) they represent the factors that are important to the IS professionals, who perceive differences in methodologies only in relation to them.

A final comment on the attributes: for a given environment and a given systems project, they will assume different levels of importance. Only the systems/ project manager will be able to assess their relative importance. Only then can these results serve as a good input measure for methodology selection.

5. Evaluation of techniques

Use and applicability during system life cycle phases. The task orientation of the technique determines which phase(s) of the system life cycle can use it. However, a single technique may be used in several phases, and several may be used for the same phase/task. In order to detail usage characteristics, each respondent was asked to indicate the use/applicability of each technique for each life cycle phase and sub-phases. Once again, the results are reported for the major phases only, and are combined for use and applicability.

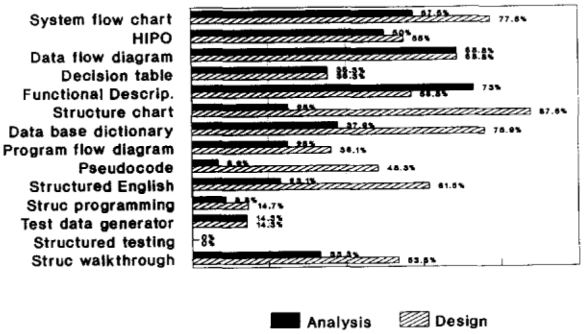

Fig. 5. Technique use in analysis and design.

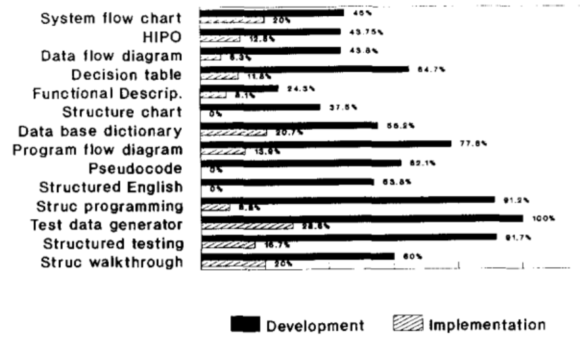

Fig. 6. Technique use in development and implementation.

Figure 5 shows the results of technique use in systems analysis and system design while *Figure 6* shows them for system development and system implementation. Data flow diagram and data dictionary have been advocated as structured techniques to be used in the analysis phase; their heavy use is a reinforcement of the recommendations. HIPO, a semi-structured technique from the IBM Corporation also enjoys heavy use. System flow chart is a traditional technique emphasizing physical and procedural details; its use is heavily favored in the design phase. Structured techniques have apparently finally arrived. Structured walkthroughs are also high in the rankings. A positive sign is the heavy use of testing techniques; apparently systems are being increasingly evaluated before implementation.

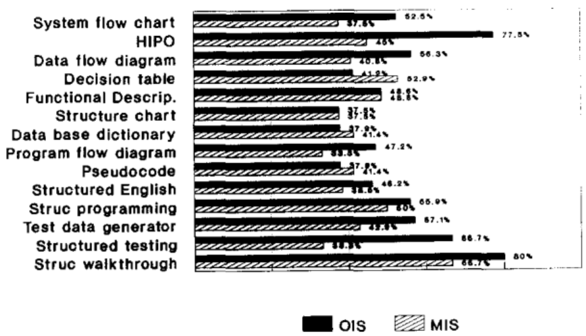

Fig. 7. Technique use in OIS and MIS.

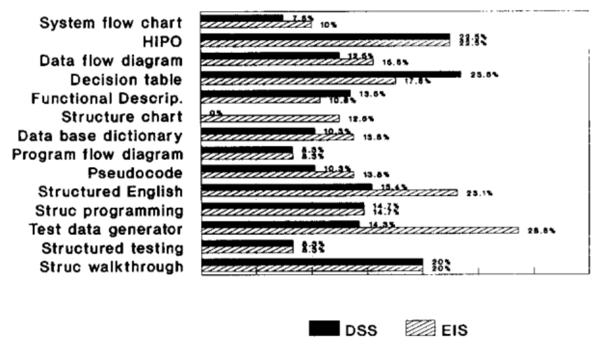

Fig. 8. Technique use in DSS and EIS.

Use and applicability for different system types. Results are grouped for OIS and MIS in *Figure* 7, and DSS and EIS in *Figure 8.* One clear observation is that there are many techniques suitable for OIS and MIS. This should not be very surprising since most of these techniques have originated there. Closer look at the results also suggests that the methods for developing OIS and MIS are about the same (although the specific numbers for are somewhat different). For DSS and EIS, the overall numbers are much lower compared to OIS and MIS, but that is primarily because fewer organization are developing DSS and EIS. Again, there are similarities between DSS and EIS in the use of the techniques.

Use and applicability based on problem structure. Figure 9 shows the use of the techniques for well-structured and not-well-structured problems. As is readily apparent, these techniques are far more applicable for structured problems. For not-well structured problems, the results do not show any clear pattern. This may be attributed to three reasons: first, these techniques may not be applicable for such problems; second, very few developers have experience with developing systems for not-well structured problems; and third, apparently there is still not a good understanding of what is meant by problem structure and it is not consciously considered in system development, despite its emphasis in the last decade [20].

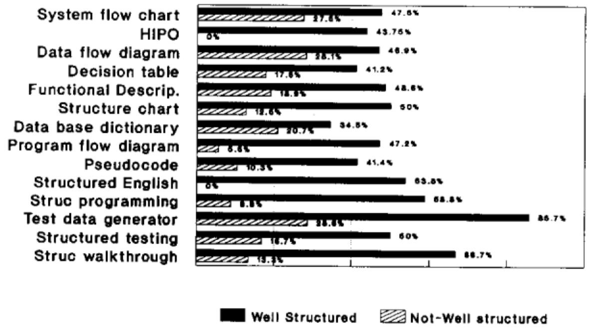

Fig. 9. Technique use by problem structure.

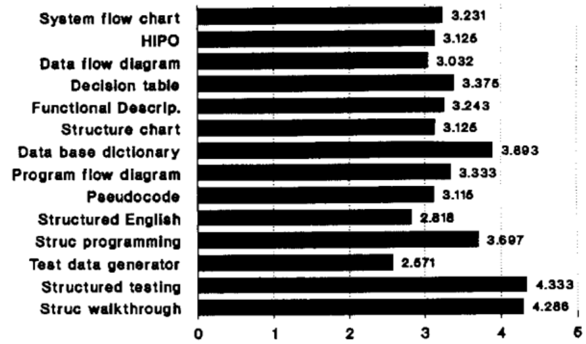

Fig. 10. Project control provided by technique.

Perceived attribute values. Figures 10 to 19 show the average ratings received by the techniques for each of the ten attributes. Higher numbers represent more favorable values of the attributes. While significant differences are of primary value, it is also useful to examine the individual attribute values for each technique.

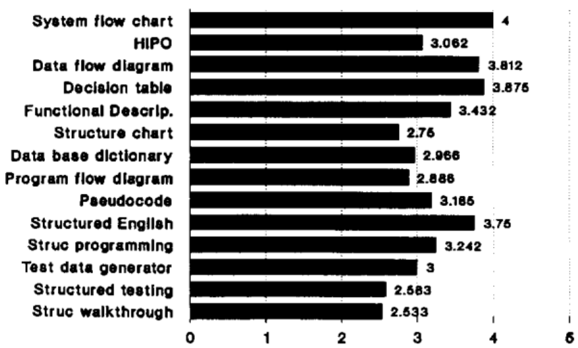

Fig. 11. Cost of use of technique.

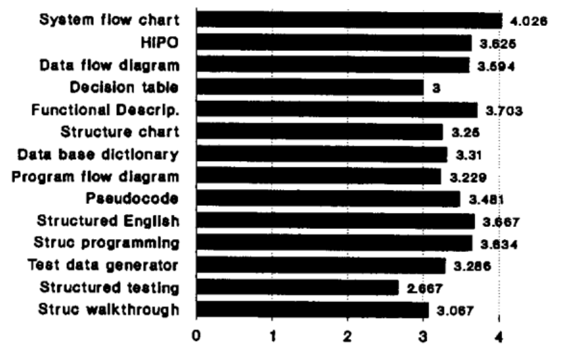

Fig. 12. Ease of use of technique.

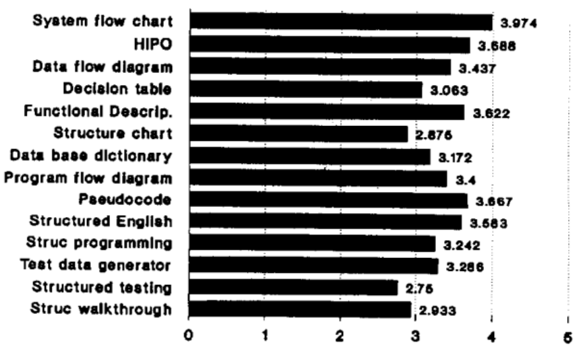

Fig. 13. Ease of learning of technique.

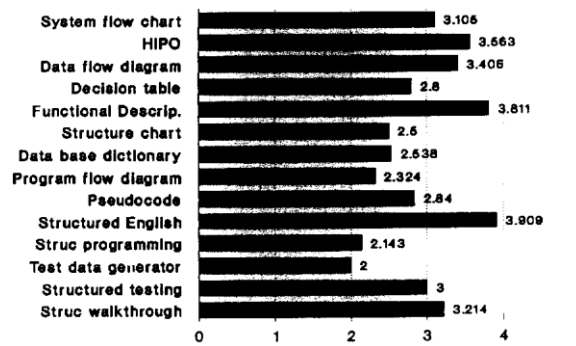

Fig. 14. Technique communicability to end user.

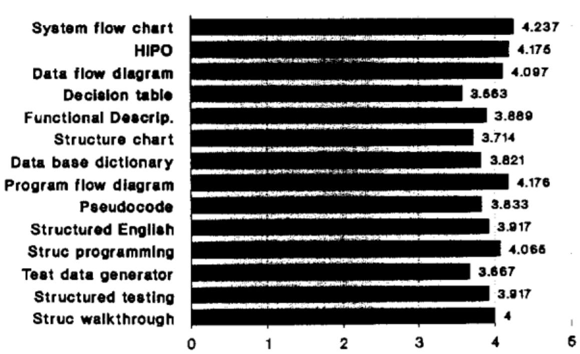

Fig. 15. Technique communicability to IS professional.

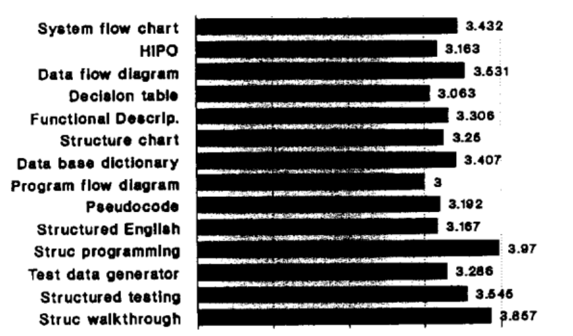

Fig. 16. Flexibility of design produced by technique.

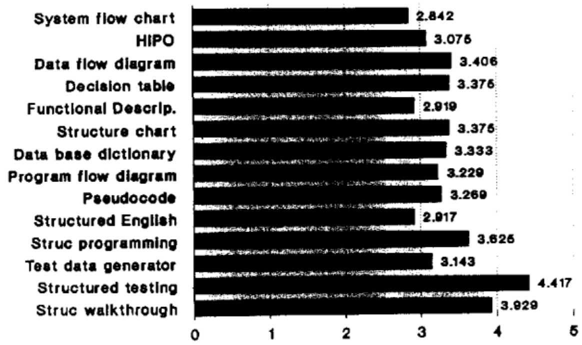

Fig. 17. Early problem discovery due to technique.

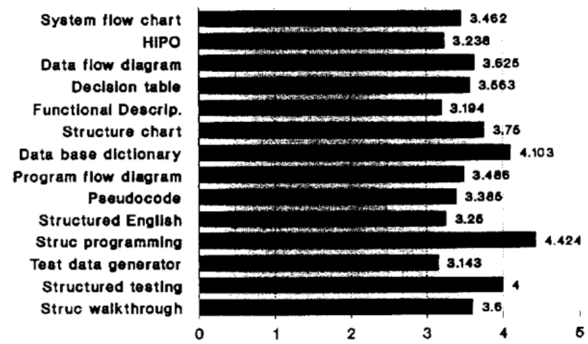

Fig. 18. Future maintainability due to technique.

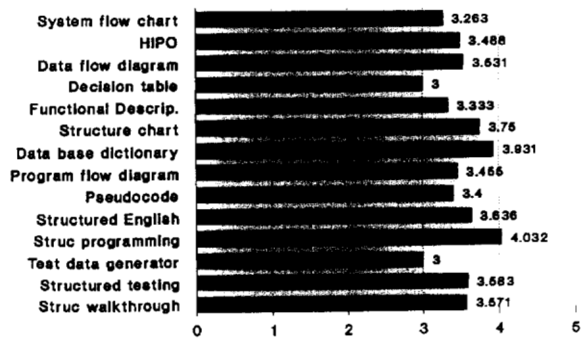

Fig. 19. Product quality generated by technique.

For project control, the structured techniques clearly dominate. In terms of communicability to end user, the highest rated techniques minimize technical jargon. For communicability to the IS individual, the highly rated techniques show procedural and hardware information; thus communicating the details necessary to the IS professional. In general, the ratings of all methods is high for communicability to the IS individual, as these methods were primarily developed by designers for designers.

While all attributes have importance in their own right, several techniques had statistically significant differences in the attribute ratings. Based on analysis of variance (ANOVA) tests, the following attributes were significant: cost of use (p value = 0.002), ease of use (p value = 0.064), communicability to end user (p value = 0.000), leading to maintainable systems (p = 0.012), and quality of generated documents (p = 0.095). If comparisons among techniques are being made, it is these five attributes where comparisons will be meaningful.

Discussion

Model for method selection

The most tangible and practical use of our results is in the selection of methods for IS development: we provide a set of important parameters that needs to be evaluated. Several other "intangible" factors must also be considered in this decision; e.g., prior experience with methods, current expertise, likes and dislikes, currently used methods, etc. The article also lends support for a contingency approach for method selection.

Based on the results, we offer the following normative model: (a). First select the overall methodology based primarily on system type and problem structure. The decision may also be influenced, to a lesser extent, by the needed attributes, (b) At each phase of system development, select appropriate techniques based on a contingency analysis. It does appear, however, that a critical set of four or five techniques may be adequate for most organizations. For example, of the fourteen techniques that we have discussed, the following are identified as systems analysis and design techniques.

A selection from these techniques can be made based on a comparative examination of desirable attributes.

ANALYSIS	*DESIGN*
— System flow chart	— Structure chart
— HIPO	— HIPO
— Data flow diagram	— Pseudocode
— Decision table	— Program flow diagram
— Functional description	
— Structured English	— Decision table

Importance of attributes
Based on requirements for a given environment, organization, and the systems project, attributes will assume different levels of importance. Thus any weights applied to the attributes are not universal, and must be decided by the systems/ project manager in the proper context. In essence, the context-specific relative importance of the attributes leads to the contingency analysis. Different IS requirements for different situations have been discussed in [12,15], and similar contingency approaches have been advocated in the literature [e.g., 9,18].

Comparison of problem structure and system type
As a by-product of our analysis, we found that there is a correspondence between well structured problems and OIS/MIS systems, and not-well structured problems and DSS/EIS systems. Figures 2 and 3 do show the high correspondence for methodology use for well structured problems and OIS/MIS system. Although there is not as close a correspondence for not-well structured problems and DSS/EIS, the trends are in the right direction. What this means is that while IS designers may not consciously consider the structure of the problem domain, they do understand that DSS/EIS systems are of the type that requires a more iterative methodology (a requirement for not-well structured problem domains).

System dichotomy
While we considered four types of systems, it seems that the practitioners are more comfortable with a dichotomy. In terms of the use of methodologies and techniques, the development characteristics of OIS and MIS are similar, while those of DSS and EIS are also similar. Given this practitioner experience, it may seem appropriate to accept the dichotomy, classified as: management information systems (which also includes OIS; in reality, the common usage of the term MIS includes OIS anyway), and management sup-port systems (which is made up of both DSS and EIS) [22].

Conclusions
This article has identified the many system development methods that are in existence, and those in most use. For those in most common use, a comprehensive evaluation was made based on field experiences of IS professionals. The most apparent utility of the results is in the selection of methods for IS development: our results provide one set of important parameters.

We also emphasize the need for contingency analysis on part of the IS organization for the purpose of selecting methods. Depending on the relative importance of attributes in the organization, environment, and for the system project, different methods will be suitable. We have provided base line data for such a contingency analysis.

Notes:
1 Partial results of this article have been presented at the 1990 Hawaii International Conference on System Sciences and 1989 Decision Sciences Institute annual conference
2 Some methods have now been renamed. For example, SADT is now IDEF, SREM is RDD, and ISDOS has moved out of the University of Michigan.

References
[1] Colter, M.A. A Comparative Examination of Systems Analysis Techniques. MIS Quarterly. vol 8, no. 1, March 1984.

[2] Connor, D. Information System Specification and Design Road Map. Prentice-Hall, 1985.
[3] Couger, D., Colter, M. and Knapp, R. Advanced System Development/Feasibility Techniques. John Wiley and Sons. 1982.
[4] Davis, G.B. and Olson, M.H. Management Information Systems: Conceptual Foundations, Structure, and Development. New York: McGraw-Hill, 1985.
[5] DB Directions 1985. Kahn, B.K. and March, S.T. IRM and the System Life Cycle. Section on Methodologies, Tools and Techniques by Cline, J.E., Ketabchi, M.A., Link, J., Palvia, P. and 01le, T.W. Database Engineering, June 1986. pages 43-48.
[6] Doke, E.R. and Myers, L.A. Jr. A Matrix Approach to the Analysis of Productivity Aids in the System Development Life Cycle. Proc. 1986 Nat'l DSI conf. Hawaii.
[7] Gingberg, M.J. and Ariav, G. Methodologies for DSS Analysis and Design: A Contingency Approach to Their Application. ICIS 1986 conference, San Diego.
[8] Gore, M. and Stubbe, J. Elements of Systems Analysis. WCB. 3rd. Ed., 1983.
[9] Gremillion, L.L. and Pyburn, P. Breaking the Systems Development Bottleneck. Harvard Business Review. March—April, 1983.
[101 Guimares, T. A Study of Application Program Development Techniques. Communications of the ACM, vol. 28, no. 5.
[11] Juhn, S.H. and Naumann, J.D. The Effectiveness of Data Representation Characteristics on User Validation. 1985 ICIS Conference, Indianapolis.
[12] Ives, B., Hamilton, S. and Davis, G. A Framework for Research in Computer-based Management Information Systems, Management Science, September, 1980.
[13] Kendall, K.E. and Yoo, S. Software Design with the PB (Pseudocode-Box) Diagram: Performance Evaluation Using a Quasi-Experimental Method. Proc. 1986 Nat'l DSI conf. Hawaii.
[14] Martin, J. Fourth-Generation Languages, Vol. I, Prentice-Hall, 1985.
[15] Mason, R.O. and Mitroff, I.J. A Program for Research on Management Information Systems. Management Science, Vol. 19, No. 1, 1970.
[16] McDonald, C.W., Riddle, W. and Youngblut, C. Stars Methodology Area Summary. Vol. II: Preliminary Views on the Software Life Cycle and Methodology Selection. ACM Software Engg. Notes. Vol II, No. 2, April 1986.
[17] Naumann, J.D. and Jenkins, A.M. Prototyping: The New Paradigm for Systems Development. MIS Quarterly, Vol. 6, No. 3, September 1982.
[18] Naumann, J.D., Davis, G.B. and Mckeen, J.G. Determining Information Requirements: A Contingency Method for Selection of a Requirements Assurance Strategy. The Journal of Systems and Software, vol 1, 1980.
[19] Necco, C.R., Gordon, C.L., Tsai, N.W., Systems Analysis and Design: Current Practices. MIS Quarterly, vol 11, no 4, December 1987.
[20] Nosek, J.T. Contingency Approaches to IS Development Methodology Selection: Current Practices and Future Directions. Proceedings, National DSI, 1988, Las Vegas.
[21] Sumner, M. and Sitek, J. Are Structured Methods for Systems and Analysis and Design Being Used? J. of Syst. Mgmt. June, 1936.
[22] Turban, E. Decision Support and Expert Systems. 2nd Ed., Macmillan, 1990.
[23] Vessey, I. and Weber, R. Structured Tools and Conditional Logic: An Empirical Investigation. CACM. Jan. 1986. Vol 29, 1.
[24] Wells, C.E. and Naumann, J.D. An Empirical Comparison of User Understandability: System Flow Charts and Data Flow Diagrams. Proc. 1985 Nat'l DSI conf. Las Vegas.