An Assessment of Systems Development Methodologies

T. HILLMAN WILLIS
LOUISIANA TECH UNIVERSITY

AND

DEBBIE B. TESCH
NORTHEAST LOUISIANA UNIVERSITY

ABSTRACT

This paper investigates the data processing manager's perception of systems development techniques. Of particular interest is how MIS/DP managers perceive user involvement in the Systems Development Life Cycle (SDLC) and the employment of systems development tools. A survey was focused on the following systems development considerations: perceptions toward user involvement in systems development, perceptions toward systems development technologies, the use of systems development tools. The authors contend that early user involvement leads to improved systems quality and user acceptance of the developed system.

INTRODUCTION

Since the mid-1970s, systems development personnel have been following various Systems Development Life Cycle (SDLC) methodologies [2]. As a disciplined approach to creating information systems, the SDLC (while not guaranteeing success) can improve the chances of developing a successful system. Most DP personnel would agree that a phased approach to systems development is appropriate even though within various companies and organizations, and among textbook authors, particular phases of the SDLC may differ. These differences are a result of the tendency to overlap phases and/or to receive varying degrees of emphasis, but do not necessarily reflect differences in the systems development philosophy [11].

Regardless of the particular SDLC phases employed, successful systems development is affected by the degree of user involvement and the development tools employed. Ives and Olson [4, p. 586] contend that "existing research is poorly grounded in theory and methodologically flawed." This leaves practitioners with intuition, experience, and unsubstantiated claims asa their best guide to systems development. The MIS literature indicates that user involvement is a necessary condition for successful systems development. Indeed, research shows that user involvement leads to improved systems quality and user acceptance of developed systems. For examples, see Ives and Olson [4].

Increased user acceptance through user involvement may lead to several improvements: (1) users' more realistic assumptions regarding systems' capabilities, (2) user ownership of the system, (3) user commitment to the system, (4) a decrease in user resistance to change, and (5) a vehicle for conflict resolution.

Previous measures of user involvement in systems development have been focused on either general involvement in Computer-Based Information System (CBIS) development or user involvement in the design of a specific system. Typically, these measures are single or multiple-item Likert scale tested — based on self-reports of the users own involvement. Some studies had users and information systems managers rate user involvement [4]. Measures of outcome variables have included system quality, system usage, information satisfaction, and user/behavior perceptions.

In defense of systems development personnel, it is difficult to visualize the best design for a system in advance. Design errors may result from unexpected changes in the user environment during development, or from simple misunderstandings due to poor communication between developer and user. Communication problems, a major source of system failure, can result in ambiguous definitions in the requirements and analysis phases [9].

Case studies of systems development by several researchers [7] have shown exponential increases in the cost of correcting specifications during the progression of a project's life cycle. Project managers need to provide quality systems and not just meet budget and time constraints. Precise systems definition requirements decrease the need for systems maintenance while increasing the likelihood for delivering an acceptable system [3]. Therefore, a strong case can be made for the considerable savings potential in achieving an acceptable systems definition as early as possible.

Table 1
MIS/DP Perceptions of User Involvement and SDLC Procedures

	Questions (n = 71)	MIS/DP Perceptions Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
1	User involvement in systems design is a hindrance.	73% (52)	27% (19)	0% (0)	0% (0)	0% (0)
2	User lack of familiarity with systems development technology makes it difficult to participate in systems development.	17% (12)	44% (31)	10% (7)	24% (17)	4% (3)
3	Users are unable to effectively communicate their needs.	17% (12)	52% (37)	3% (2)	25% (18)	3% (2)
4	User participation in systems development results in less revision after implementation.	4% (3)	7% (5)	3% (2)	44% (31)	42% (30)
5	User stated requirements and delivered systems often differ.	1% (1)	20% (14)	7% (5)	59% (42)	13% (9)
6	Successful systems followed a formal approach to systems development.	1% (1)	10% (7)	14% (10)	61% (43)	14% (10)
7	Management is receptive to the idea of using available applications prototyping tools.	1% (1)	17% (12)	31% (22)	48% (34)	3% (2)
8	People involved in developing systems are unfamiliar with traditional system development life cycle techniques.	3% (2)	34% (24)	24% (17)	38% (27)	1% (1)
9	There is not enough time to follow prescribed procedures for developing systems.	8% (6)	38% (27)	8% (6)	35% (25)	10% (7)

Traditional systems development tools — observation, interviews, questionnaires, flowcharts, decision tables — are conducive to user involvement in systems definition requirements, and in facilitating systems design (although flowcharts seem to generate an "early freeze in the design"). To compensate for this tendency, new structured analysis tools — data flow diagrams, HIPO charts, pseudocodes, etc. — were proposed [11].

Available software technology for applications prototyping (screen generating software, fourth-generation languages, and program code generators, for example) also provide viable enhancements to SDLC methodologies. As Janson and Smith [5] point out, applications prototyping encompass techniques which can improve the ability to meet users' needs while reducing a need for systems revision. Prototyping is a four-phased procedure emphasizing identification of essential systems requirements [1]. The essential advantage of prototyping lies in increased user involvement.

The purpose of this paper is to investigate the data processing manager's perception of user involvement in the Systems Development Life Cycle (SDLC) and the use of available systems development tools. Data processing practitioners were surveyed by a questionnaire submitted to members of two chapters of the Data Processing Management Association (DPMA), Region 3. One hundred, thirty-three questionnaires were mailed. Questionnaires were restricted to business members associated with systems development practices. Seventy-one usable questionnaires were returned, representing a 53 percent response rate. The study results include (1) MIS/DP personnel perceptions, (2) systems development tool utilization, (3) user involvement, and (4) implications for systems development success.

MIS/DP PERSONNEL PERCEPTIONS

Respondents were asked to indicate the extent of agreement/disagreement with items requiring a judgement of perceptions. Interestingly, two major perceptions were of particular interest — user involvement in systems development and perceptions toward the use of systems development techniques.

Perceptions of User Involvement

Questions one through five in Table 1 show responses to the issue of the respondents' perception toward user involvement in systems development. It is important here to keep in mind that responses pertain to MIS/DP personnels' perceptions of user involvement and do not reflect self-reports of users. These perceptions reflect impressions of general involvement in CBIS development and are not system specific.

The responses indicate that user involvement in systems design would not be a hindrance. A majority (61 percent) felt that users could participate in systems development without being familiar with systems development techniques — this opposed to 28 percent feeling that it was difficult for users to participate in systems development without being familiar with systems development techniques. This suggests that there is some concern over a possible lack of user knowledge.

If users do not understand the technical terminology, could communication during the requirements phase be hindered? This issue can be discussed in light of the answers to questions three and five. Sixty-nine percent of the respondents to question three perceived the users as able to effectively communicate their needs; seventy-two percent of the respondents to question five felt that user-stated requirements often differed from delivered systems. Therefore, disappointment with the delivered system is not a result of poor communication, but, rather, a multitude of other possible factors — at least in the sample.

Practitioners strongly agreed (86 percent in question four) that user participation in systems development would result in more acceptable systems. It is apparent from these results that practitioners believe a successful system is highly dependent on improving the user input process.

Perceptions of System Development Techniques

Items six through nine (Table 1) queried the respondents' perception toward the use of systems development techniques. While there was a consensus (75 percent in question six) that a formal approach to systems development led to successful systems, respondents were inconclusive (37 percent disagreed, 39 percent agreed) in judging whether systems development personnel were familiar with SDLC techniques (item eight). Perhaps this can be explained by the possibility that practitioners' definitions of the SDLC differ.

It is interesting to observe that management is inclined to support the use of prototyping tools in the systems development process (only 18 percent disagreed with this position in question seven). Responses to question nine were mixed as to whether or not there was adequate time in systems analysis and design projects to follow prescribed procedures for developing information systems. However, this is probably more of an indication of how each DP organization actually functions rather than being an evaluation of the technique when properly applied.

SYSTEM DEVELOPMENT TOOL UTILIZATION

Practitioners were questioned about the extent to which they utilized various development tools within their organization. Table 2 shows the survey results for 12 development tools. An examination of Table 2 indicates that the development tools most often utilized included interviews (99 percent), observation (83 percent), data flow diagrams (74 percent) and flowcharts (71 percent). These results are in line with previous studies [6,10]. HIPO charts and pseudocodes were the tools with the least amount of usage. HIPO charts may be considered more of a documentation tool than a strategy for development. It seems that the proposition of replacing flowchart methods with pseudocodes is not happening, perhaps because pseudocoding reflects a need for certain programmng logic skills [11].

Forty-three percent of those surveyed indicated that Structured Analysis and Design Technique (SADT) was not used at all, 41 percent indicated that it was seldom used, and 20 practitioners did not respond to the question. These data suggest that a number of respondents may not have been familiar with SADT which was developed by a private firm and consists of a graphic language for model-building, a model development methodology, and management procedures for model development control [8].

Design tools (including screen generating software, fourth-generation languages, and program code generators) are specifically required to utilize application prototyping techniques. These three tools were described as being used frequently by 43 percent, 43 percent, and 31 percent of the respondents, respectively. Whether practitioners have the skills needed to implement prototyping application on a wide scale is an important consideration requiring further examination.

An investigation of the relationship between traditional development tools, structured tools, application prototyping tools, Structured Analysis and Design Technique (SADT), and the degree of usage of these tools was conducted. From Table 2, the frequencies for items 1-5 were grouped as traditional tools; items 6-8 were grouped as structured; item 9 represented SADT; and items 10-12 were grouped as application prototyping techniques.

Table 2
Perceived Utilization of Development Tools

	Development tool	Perceived Use: Extensive	Frequent	Seldom	Not at all	Other
1	Observation	27% (19)	56% (40)	13% (9)	3% (2)	1% (1)
2	Interviews	59% (42)	39% (28)	0% (0)	0% (0)	1% (1)
3	Questionnaires	4% (3)	14% (10)	52% (37)	27% (19)	3% (2)
4	Flowcharts	27% (19)	44% (31)	24% (17)	4% (3)	1% (1)
5	Decision Tables	11% (8)	25% (18)	38% (27)	23% (16)	3% (2)
6	HIPO Charts	3% (2)	9% (6)	46% (31)	40% (27)	1% (1)
7	Pseudocodes	9% (6)	10% (7)	41% (28)	38% (26)	2% (2)
8	Data Flow Diagrams	20% (14)	54% (38)	18% (13)	7% (5)	1% (1)
9	SADT	2% (6)	6% (3)	41% (21)	43% (22)	8% (4)
10	Screen Generating Software	15% (11)	28% (20)	27% (19)	25% (18)	4% (3)
11	Fourth-Generation Languages	13% (9)	30% (21)	20% (14)	35% (25)	3% (2)
12	Program Code Generators	8% (6)	23% (16)	24% (17)	42% (30)	3% (2)

Results shown in Table 3 indicate that the various types of systems development tools are not used in the same proportion (X^2 = 106.06959, significant at .000001). Moreover, the positive association of types of tools with lower use of the tools (φ^2 = .35835, C = .33734, τ_b = .26798, all significant at .000001) seems to indicate that nontraditional systems development tools are less used than traditional tools. Perhaps what happens is that the more recent tools (structured and prototyping) are used in addition to the traditional ones when the practitioners feel that they could complement or increase the success of the systems design effort.

PERCEIVED USER INVOLVEMENT IN SDLC PHASES

Table 4 indicates how users are involved in various phases of the systems development process. As expected, users are most often involved in the problem definition and post-implementation phases. Eighty-nine percent of users were involved from 60 percent to 100 percent of the time during problem definition — the point at which users should reveal problems and begin to specify needs.

Remember that more than one-third of those surveyed felt that people involved in developing systems were not familiar with SDLC techniques. If data processing personnel are unfamiliar with the traditional requirements for these phases it could distort responses to questions of user involvement in these phases.

Users did not appear to be heavily involved in the analysis and design phases. The traditional development approach leaves the main responsibility for analyzing requirements and designing the appropriate systems to the systems development staff. Frequently, the design is completed with minimal user feedback. The survey results support this possibil-

Table 3
Contingency Table for the Perceived Use of Development Tools

TOOLS	Count / Row Pct	USE extensive 1	frequent 2	seldom 3	not at all 4	other 5	Row Total
Traditional	1	91	127	90	40	7	355
		25.6	35.8	25.4	11.3	2.0	43.0
Structured	2	22	51	72	58	4	207
		10.6	24.6	34.8	28.0	1.9	25.1
Prototyping	4	26	57	50	73	7	213
		12.2	26.8	23.5	34.3	3.3	25.8
SADT	3	1	3	21	22	4	51
		2.0	5.9	41.2	43.1	7.8	6.2
	Column Total	140	238	233	193	22	826
		16.9	28.8	28.2	23.4	2.7	100.0

Chi-Square (X^2)	Value	DF	Significance
Pearson	106.06959	12	.00000
Likelihood Ratio	113.31109	12	.00000

Statistic	Value	T-value	Approximate Significance
Phi (φ^2)	.35835		.00000 *1
Contingency Coefficient (C)	.33734		.00000 *1
Kendall's Tau-b (τ_b)	.26798	9.67867	.00000

*1 Pearson X^2 probability

Table 4
Perceived Degree of User Involvement in the SDLC Phases

	SDLC Phases	Perceived degree of user involvement 80–100%	60–79%	40–59%	20–39%	19–0%
1	Problem definition	66%	23%	3%	6%	2%
		(47)	(16)	(2)	(4)	(2)
2	Systems analysis	11%	27%	35%	17%	10%
		(8)	(19)	(25)	(12)	(7)
3	Systems design	4%	24%	28%	21%	23%
		(3)	(17)	(20)	(15)	(16)
4	Implementation	31%	20%	27%	14%	8%
		(22)	(14)	(19)	(10)	(6)
5	Post-implementation	51%	23%	14%	6%	6%
		(36)	(16)	(10)	(4)	(5)

ity. On the other hand, the degree of user involvement in systems implementation was much higher. It is at this point that users can begin to determine whether or not the system meets their needs.

There was even greater user involvement in the post-implementation phase. This phase includes the maintenance and enhancements necessary to provide users with a workable system that satisfies the requirements. Greater user participation in the analysis and design stage could significantly shorten the time spent with post-implementation maintenance.

IMPLICATIONS FOR SYSTEMS DEVELOPMENT SUCCESS

The survey results provide important implications concerning managers' perceptions of user involvement in the Systems Development Life Cycle process and their utilization of available software development tools. While more rigorous treatment of this subject is necessary (i.e., inclusion of users of target systems as respondents; treatment of a wider geographical nature; and breakdown of employment capacities of MIS/DP personnel), observations here my be applied to any organization's system development practice.

The survey results indicate that MIS/DP personnel believe in user involvement. They agree that user involvement in systems development is not a hindrance and that users do not necessarily need to be familiar with systems development techniques to make a contribution. Indeed, extensive user involvement can be of para,mount importance. When users are involved in system development, it is logical that there will be less need for revisions.

The most interesting observation is that users, in general, are able to communicate their needs, yet final systems frequently do not satisfy their expectations. If users are able to effectively communicate their needs, there must be other explanations for the difference between user-stated requirements and delivered systems. It may be that during the time it takes to specify the requirements of the system, a user's environment and needs may have changed. Application prototyping offers a timesavings that could lessen chances of obsolescence; screens and programs can be built faster than the time it takes to write specifications. Although the tried and true traditional tools of design (observation, interview, flowcharting) still prevail in practice, the availability of prototyping tools (screen generators, fourth-generation languages, program code generators) is significant.

A second, and more common, explanation of user dissatisfaction stems from the difficulty users have in accurately specifying their problems and requirements. Not that users lack the necessary communicative skills, it is simply an uncertain, if not impossible, task. Users cannot be expected to foresee all the design options that may exist or the effect that their design features might have on other system components. A strategy to get the user more involved in the earliest stages of the requirements and design phase would be beneficial.

In conclusion, while the literature indicates that user involvement is a necessary condition for successful systems development, successful integration of any technology into an organization can be achieved only with management's support. This support can be solicited only if management believes in the methodology. New systems development technologies will become more widely recognized and accepted as part of the systems development methodology when management becomes aware of the benefits demonstrated through successful application.

REFERENCES

[1] Alavi, M., "An Assessment of the Prototyping Approach to Information Systems Development," *Communications of the ACM,* Volume 27, Number 6, June 1984, pp. 556-563.

[2] Appleton, D.S., "Data-Driven Prototyping," *Datamation,* Volume 29, Part 3, November 1983, pp. 259-268.

[3] Boar, B.H., *Application Prototyping,* John Wiley and Sons, New York, New York, 1984.

[4] Ives, B. and Olson, M.H., "User Involvement and MIS Success: A Review of Research," *Management Science,* Volume 30, Number 1, May 1984, pp. 586-601.

[5] Janson, M.A. and Smith, L.D., "Prototyping for Systems Development: A Critical Appraisal," *MIS Quarterly,* Volume 9, Number 4, December 1985, pp. 305-316.

[6] Kievit, K. and Martin, M. "Systems Analysis Tools — Who's Using Them?," *Journal of Systems Management,* July 1989, pp. 26-30.

[7] Kull, D. "Designs on Development," *Computer Decisions,* Volume 17, April 9, 1985, pp. 86-105.

[8] Lucas, H.C., Jr., *The Analysis, Design, and Implementation of Information Systems,* McGraw-Hill, New York, New York, 1985.

[9] Sroka, J.M., and Rader, M.H. "Prototyping Increases Chance of Systems Acceptance," *Data Management,* Volume 24, Number 3, March 1986, pp. 12-19.

[10] Sumner, M. and Sitek, J., "Are Structured Methods for Systems Analysis and Design Being Used?," *Journal of Systems Management,* June 1986, pp. 18-27.

[11] Whitten, J.L., Bentley, L.D., and Ho, T.I.M., *Systems Analysis and Design Methods,* Times Mirror/Mosby College Publishing, St. Louis, Missouri, 1986.

ABOUT THE AUTHORS

T. Hillman Willis, Ph.D., C.P.I.M., C.D.P., is associate professor of management and marketing in the College of Administration and Business at Louisiana Tech Univer-

sity. He received his Ph.D. in quantitative methods from Louisiana State University, where he earlier had earned a bachelor's degree. His MBA was completed at Memphis State University. Dr. Willis is a retired Captain in the U.S. Coast Guard Reserve. He is a member of the Decisions Sciences Institute, American Production and Inventory Control Society, and the Data Processing Management Association. He is certified in data processing and in production and inventory management. His articles have appeared in several journals.

Debbie B. Tesch, *M.B.A., C.D.P., is a graduate of the University of North Carolina at Greensboro with a bachelor's degree in education and a master's de gree in business administration. She was employed for five years as a programmer and programmer/analyst. She received her certificate in data processing (CDP) in 1986. Currently, she is an instructor in computer information systems at Northeast Louisiana University, and is a graduate student in the DBA program at Louisiana Tech University.*