Conceptual Perspectives on Key Factors in DSS Development: A Systems Approach

Author(s): Taesik Ahn and Gary Grudnitski

Source: *Journal of Management Information Systems*, Summer, 1985, Vol. 2, No. 1 (Summer, 1985), pp. 18-32

Published by: Taylor & Francis, Ltd.

Stable URL: https://www.jstor.org/stable/40397815

Conceptual Perspectives on Key Factors in DSS Development: A Systems Approach

TAESIK AHN and GARY GRUDNITSKI

TAESIK AHN is currently a doctoral student in the Department of Accounting at the University of Texas at Austin. He received his B.B.A. degree from Seoul National University, Korea. His research interests include accounting applications of decision support systems, systems development and implementation methodologies, and the evaluation of information systems and their development process.

GARY GRUDNITSKI is Associate Professor of Accounting at the University of Texas at Austin. He holds a B.Comm. and an MBA degree from the University of Saskatchewan, Canada, and a PH.D. from the University of Massachusetts, Amherst. His publications appear in several journals, including the *Journal of Systems Management*, *The Accounting Review*, the *Journal of Accountancy*, and the *Journal of Management Information Systems*. He is co-author of the text *Information Systems: Theory and Practice*. Currently he is Chairman of the Management Advisory Services/Information Systems section of the American Accounting Association.

ABSTRACT: It is a mistake to categorize decision support systems (DSS) as just another recent variant of traditional computer-based information systems. While traditional computer-based information systems are oriented toward performing structured processing tasks and producing regular reports, DSS have the purpose of supporting nonroutine activities and evolving in response to user learning. Thus a different subset of factors should be considered when developing a decision support system. This paper proposes and defines a conceptual model of DSS development whose origins are based on an open systems approach and whose components capture technical, behavioral, and organizational aspects of the development process.

KEY WORDS AND PHRASES: Decision support system, systems development, system approach, user-system interface, learning-oriented systems, organization support, redistribution of power, feedback.

We gratefully acknowledge the helpful comments from two anonymous reviewers.

Introduction

TRADITIONAL COMPUTER-BASED INFORMATION SYSTEMS (CBIS), which are intended primarily to perform basic transaction processing and information reporting functions [30, 28] are comprised of structured and predefined tasks. Because of the degree of foreknowledge about these tasks, the required characteristics of CBIS can be laid out in advance of their development. Hence the challenge to the developer becomes one of selecting the design that performs the given tasks as efficiently as possible in relation to some predefined performance criteria.

Decision support systems (DSS) go beyond the usual functions of transaction processing and production of periodic reports. Their role is to support nonroutine tasks [28]; they are, by their nature, evolutionary. Moreover, because DSS are intended to provide flexible, ad hoc decision support under user/manager control, they often rely on prototypes and evolve in response to user learning. These differences between CBIS and DSS are compelling reasons for considering the development process of DSS from a unique perspective.

The fundamental emphasis in the 1970s was on the technical aspects of systems development [11]. When a decision support system was considered, however, researchers stressed the need for a development approach that reflected a set of descriptors such as heuristic, adaptive, iterative, and evolutionary. While such descriptors can help us gain an understanding of the nature of DSS, they are inadequate development guideposts because they are ex post (after the development process) descriptors about the overall nature of DSS development process. The intent of this paper, then, is to present a model that identifies a number of key components to evaluate and monitor throughout the process of developing DSS. These key components make up important guidelines as system developers, and ultimate users work through the process of developing a decision support system.

Closed versus Open System Approach to DSS Development

GINZBERG [11] argues that a central problem in CBIS development is the failure to attend to nontechnical aspects of the project. Many system implementation failures are caused by emphasizing technical aspects of the system to the exclusion of behavioral and organizational impacts within an organization.

Ackoff [2] defines closed and open systems as follows:

> A closed system is one that has no environment. An open system is one that does. Thus a closed system is one which is conceptualized so that it has no interaction with any element not contained in it.

Previous development approaches, which emphasize only the technical aspects of the system, reflect a closed system approach to the development process because they do not take into account various interactions with the other parts of an organization. To improve the effectiveness of the development process of DSS,

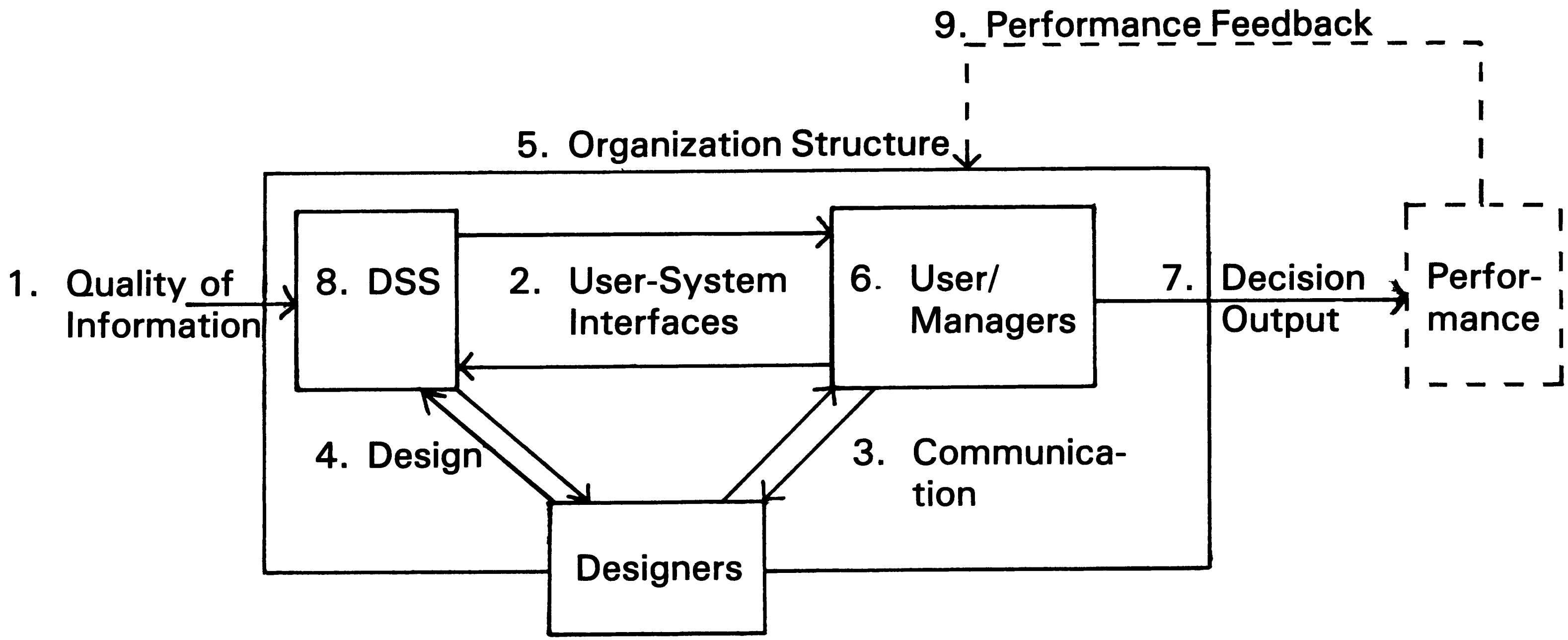

Figure 1. Development framework from a systems viewpoint

we should acknowledge their open system origins and pay explicit attention to monitoring and evaluating all possible external interactions.

A Model of a System's Components

FIGURE 1 represents a model of the components that interact with, influence, and are influenced by a decision support system within an organization. The components identified in this model take into account not only technical but also behavioral and organizational aspects of the development process.

Quality of Information within the System

One of the first steps in designing a computer-based information system is to determine the information requirements of its users. On one hand, Ackoff [1] tells us that "efficient" CBIS often produce irrelevant decision support information. On the other hand, research by Davis and Olson [7] and Langefors [18] indicates that the ability of CBIS designers to specify information perceived by managers as useful and relevant for decision-making is instrumental to the success of the system. As Ackoff [1] points out, simply "asking" managers what they need may not work because managers may not know their own information needs. No ready-made way exists, however, to specify managers' information needs for ill-structured problems. To be able to discriminate information that is useful and relevant from information that is irrelevant and useless within an organizational context, Kennedy and Mahapatra [16] suggest a method that ranks the importance (relevance) of information elements. Their method incorporates the relative rankings of importance of the information elements, the decision influenced by that information element, the importance of a particular department in an organization, and the importance of a particular decision in a department. The importance (rank) of the information element in the total system can be estimated as,

$$Ui = \sum_{k=1}^{r} \sum_{j=1}^{n} (\hat{\alpha}_{ijk} \cdot (\hat{\beta\sigma})_{jk} \cdot \gamma_k)$$

where,
σ_{jk} = the frequency of decision j in department k,
α_{ijk} = the importance (rank) of factor i in decision j in department k,
β_{jk} = the importance (rank) of decision j in department k,
γ_k = the importance (rank) of department k in the system.

More specifically, beginning with a list of decisions for which a manager is responsible, the information evaluation procedure of Kennedy and Mahapatra asks a manager to rate each of the decision types as to its importance to the overall success of his or her job and its contribution to the organization. The ratings are then scaled by the relative frequencies of occurrences of the different decision types and normalized so that they sum to unity. To evaluate information elements, a manager is asked to rate the importance of the contribution of each information element to the decision under review. The ratings are then normalized so that the rankings of all information elements for any given decision also sum to unity.

The above procedures allow us to arrive at the importance index (μ_i) for all information kept in the CBIS. Using this index, the quality (relevance) of an information element stored in a decision support system can be determined before the quality of the decision output is evaluated. The index also allows us to monitor changes in the usefulness or relevance of an information element for strategic or managerial decision tasks as the organization attempts to adapt to a perpetually dynamic external environment.

User-System Interface

We know that humans have limited information processing capability. They have limited channel capacity [20] and limited short-term memory [22]. We also know that processing heuristics, such as representation, availability, and anchoring, can lead to a systematic bias in decision-making [27]. In traditional CBIS interactions between users/managers and computers are weak and limited. DSS are aimed, however, at unstructured tasks that have a high degree of uncertainty. The greater the uncertainty of the task, the greater the amount of information that has to be processed during the decision-making process [10]. With computer support, the amount of information that must be stored and processed by user/managers can be reduced. By explicitly recognizing the importance of user-system collaboration [25], the effectiveness of decision-making can be improved, and the computer interfaces and task allocations in a dynamic problem-solving situation can be evaluated.

Dewaele [8] suggests we think about and take into consideration the fit between DSS and decision styles of users/managers (i.e., heuristic versus analytic). If a decision support system can be designed to match a user's preferred style more closely, then we might predict more frequent and satisfactory use of this system. While the above assertion should not lead us to the conclusion that it is practical to design DSS to fit the cognitive style of a specific user (i.e., individualized DSS), DSS, however, must stress the attribute of flexibility so that they can support multiple styles of decision-makers solving several different types of tasks. Moreover, DSS should be process independent so that they are able to support a variety of decision-making processes. A decision support system designed to complement the decision-making of a user has the capability to overcome threshold resistance to its use.

By incorporating the findings of previous studies about decision-makers and decision-making processes, Carlson [6] specifies the following design characteristics of the interfaces:

1. DSS should provide familiar data representations (e.g., tabular or graphic) to help decision-makers conceptualize the decision-making process.

2. DSS should support all of the "Intelligence, Design, and Choice" activities [26] while making a decision.

3. DSS should provide a variety of memory aids to assist decision-makers in carrying out the decision-making process.

4. DSS should help decision-makers work in their own idiosyncratic ways. Idiosyncratic ways can be translated into the attribute of design flexibility, a point of elaboration in the above paragraph.

5. DSS should provide aids to help decision-makers exercise direct, personal control over the system.

In summary, the concept of user-system interaction is pivotal in the design of DSS. A human must be the controlling element of the user-system interface. DSS should support the decision-making of users/managers rather than replace it.

Interactions between Users/Managers and Designers

The very existence of a working computer-based information system results in the realization of the benefits associated with it. For the benefits to be realized by a decision support system, the expertise of both users/managers and designers must be commonly focused (to achieve the objective of improved decision-making). User involvement or design participation has been suggested as a way to achieve this purpose. It is a mistake, however, always to assume a spirit of cooperation exists between users/managers and designers, even if user involvement and participation is mandated by top management. Research about the social dynamics of user participation or involvement [24] suggests that user participation alone does little to resolve the conflicts that occur in the system development process. Participation is only effective if it allows users to exert influence toward both conflict generation and its constructive resolution.

Bardach [5] characterizes the process of implementing a system as a "game." In this game, an asymmetry of information exists between users/managers and designers. In other words, managers have better information than designers about the nature of the decision-making processes that DSS are supposed to suport. Designers, however, have better knowledge than user/managers about computer-related matters. Also, no formal hierarchy exists between the two groups when outside consultants are hired to design the system. Even though designers are willing to cooperate with users/managers, users/managers might refuse to cooperate because of their resistance to innovation. Users/managers might blame designers for poor performance even when DSS are technically well designed. Conversely, designers might blame users/managers for poor performance even when users/managers cooperate fully with designers. What should be recognized is that if one group's goal is in conflict with the other group's goal, an appropriate incentive scheme must be devised to induce cooperation or behavior directed toward a common goal. In this incentive scheme the payoff of the designers and of the users/managers should be a function of the level of cooperation, assuming the two groups have a previously agreed on measure of cooperation. Throughout the development process the coordination mechanism of the payoff scheme should attempt to implement the previously agreed on terms of cooperation and intervene if any corrective action is required. Without an appropriate coordinating mechanism as a control device, it is difficult to achieve the overall goal of the development project.

The above method of inducing cooperation between users/managers and designers can be supplemented by setting up a coordinating function and having each group report to it. Top management is a good candidate to perform this function because it has the authority to direct and control the activities of each group.

Design Activity

Design activity can be considered as a delegation of authority or a commitment of resources from top management. Whenever resources are delegated, agency problems occur. In a traditional agency model two parties (principal and agent) are involved. Principals (in this case, top management) hire agents (designers) to perform certain tasks. The model assumes that the agents who are hired are work-averse and act in their own interest.

A common way of formulating the utility function of the agents is as follows:

$$U(X, a) = X - a$$

where:

U = utility of the designer,
X = payoff for the design effort, and
a = level of work (design effort).

If a fixed payoff is specified in the hiring contract, then $U(X,a)$ becomes a decreasing function of the effort level because X is independent of the effort level. To maximize their utility, designers will reduce their effort level.

The above situation can be avoided by making the designers' payoff an increasing function of their effort level above some fixed minimum fee. The problem associated with this payoff function, however, is that principals (top management) cannot directly observe the level of effort of the designers. In other words, perfect information about the level of effort of the designers is not available to top management. The problem can be mitigated by obtaining at the negotiation stage at least agreement about the measure of effort and thereafter periodically evaluating the measure to avoid this kind of shirking behavior.

Organization Support

The existence of an organization structure can act as a constraint to the development and implementation of DSS. Conversely, because information systems redistribute data and are sometimes intended to break up data (information) monopolies, they can be explicitly used to modify organization structures. Thus the interactions of

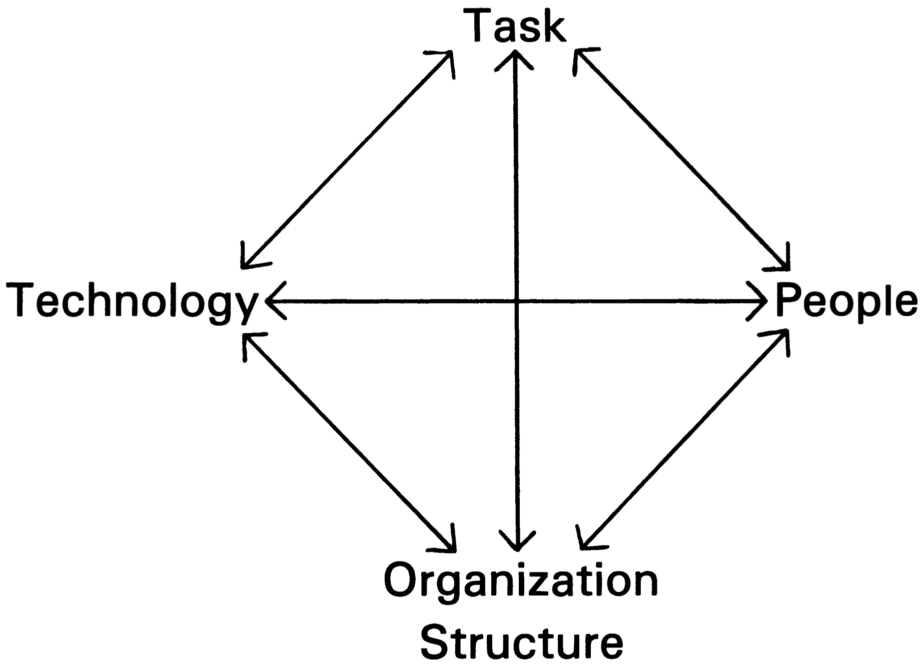

Figure 2. The Leavitt's "Diamond": components of an organization

existing organization structures and information systems should be taken into account when developing and implementing a DSS.

One way of characterizing an organization is in the form of a diamond [19]. In the diamond illustrated in Figure 2, "Technology," "People," and "Organization Structure," are interrelated facets of the organization, mutually adjusting to each other. When technology changes (e.g., a decision support system is implemented) the other facets of the organization often adjust to dampen the impact of the innovation. How much they adjust will depend on the "fit" of the change to the other facets.

Ein-Dor and Segev [9] categorize organizational context variables affecting the success and failure of systems development into uncontrollable, partially controllable, and fully controllable variables. Uncontrollable variables are organization size, organization structure, timeframe of change, and extraorganizational situation. They propose that the likelihood of success in developing CBIS is high when the organization size is large, the organizational timeframe is long, the organization is centralized, and external resources (e.g., software and hardware technology, personnel) are available immediately.

Systems formalization, level of quantification and availability of decision relevant data, and the psychological climate are partially controllable variables for successful development of a system. Rank and location of the executive responsible for information systems are suggested as fully controllable variables. Ein-Dor and Segev propose that the responsible executive be of high rank and not be in any specific functional area.

Uncontrollable variables, by definition, cannot be manipulated during the development process. Therefore, before the development process begins, adequate measurement and evaluation of the uncontrollable organizational context variables should be undertaken to determine the feasibility of a successful implementation of the decision support system. During the development process, however, the manager who is responsible for the development project needs to evaluate and monitor both partially and fully controllable variables so that the organizational environment can be made as favorable as possible to the development of a decision support system.

Relationships among Users/Managers

Organizations are viewed as groups of individuals who often have conflicting priorities, objectives, and values [4]. Information is recognized as a resource that symbolizes status, enhances authority, and (re)shapes relationships [29].

DSS cause a greater redistribution of information than CBIS. This redistribution of information and a corresponding redistribution of power can occur horizontally (between departments) and vertically (within a department). For example, even if the development of a decision support system increases the organization's total profit "pie," departments within the organization may resist (i.e., horizontal conflict) because the decision support system will cause them to lose control over the data they used to monopolize. Moreover, development of a decision support system may cause certain performance data to be integrated into a database [13]. Subordinates may be reluctant to participate (i.e., vertical conflict) in development efforts of the decision support system because they fear their superiors will better be able to evaluate their performance and, perhaps, eventually expect them to achieve higher performance goals.

The above examples give good reason for designers to take into account the redistribution of power that may accompany development of DSS. Designers need to ask the following questions:

- Who owns the data or information?
- Who will share it?
- What will be the perceived impact of the redistribution on managers' influence, authority, and communication?

Implementing a decision support system may include redesigning parts of the organization, disrupting the patterns of communication, and relocating authority [13]. These political influences may cause users/managers to engage in counterproductive implementation activities during the development process of a decision support system. Designers should take these political and psychological factors into account and devise appropriate development strategies and tactics.

Decision Output

The actual results of decisions are good surrogate measures for evaluating the effectiveness of a decision support system development project [15]. Outcomes such as an increased cash inflow or a higher inventory turnover ratio with the same degree of customer service are good examples of improved decision-making. It is unrealistic, however, to infer that DSS will exhibit the same consistency of cause-and-effect relationships we expect from traditional CBIS. This is so because decision output of DSS is partly determined by an environmental-organizational interaction, which cannot be set to a *ceteris paribus* condition. Moreover, learning is the central aspect of DSS. As experience is gained in using a new DSS, users/managers learn how to exploit the systems better. Thus consideration as to where users/managers are on the learning curve should be factored into evaluation of decision output.

The shape of the learning curve of the decision support system is usually unknown in advance and is presumed to be different for each type of system. For example, in one instance we may find a system very easy to use but not particularly powerful. Users can understand how to use the system readily, and hence the level of their decision output will increase quickly. Learning may reach a plateau on the curve rapidly. The learning curve 1 of Figure 3 is an approximate shape for this type of system. In a contrasting instance we may find a system is difficult to learn but very powerful. Learning here occurs very slowly, but the level of maximum output level can be high. The learning curve for this type of system may take the form of curve 2 in Figure 3. Therefore we should not be misled by the decision output level measured only at one specific moment. The decision output level must be evaluated periodically, and the sign and magnitude of the change in decision output should be emphasized in addition to the absolute level of decision output. In Figure 3, for example, D1 is higher than D2 at time t-1. But D1 is at the maximum output level of curve 1, while D2 is located on the increasing slope of curve 2.

Evaluation of the DSS

For traditional CBIS, whose details can be well defined before implementation, ex ante evaluation becomes a matter of choosing the proposal that accomplishes a design in the most efficient way. Traditional CBIS also usually have an exact sign-off date, which makes ex post evaluation possible.

As was stated previously, the nature of DSS development is iterative, adaptive, and evolutionary. The details of most DSS cannot be specified in advance, and a point at which the systems are "final" does not exist. Neither ex ante nor ex post evaluation is appropriate for DSS. Instead evaluation should be embedded in the implementation process so that it is an ongoing activity throughout the implementation process [12].

DSS are learning-oriented systems. The hardware and software of DSS should be integrated so as to allow users/managers to employ models and data easily. DSS architecture facilitating use and flexibility [17] and user-oriented service measures should be emphasized along with technical and organizational components in grading the technical side of DSS. Cost-benefit analysis has been the usual yardstick applied for evaluating traditional CBIS. The nature of the perceived benefits of DSS, such as improved communication and better understanding of the business, is difficult to quantify. In addition, because of the evolutionary nature of DSS development,

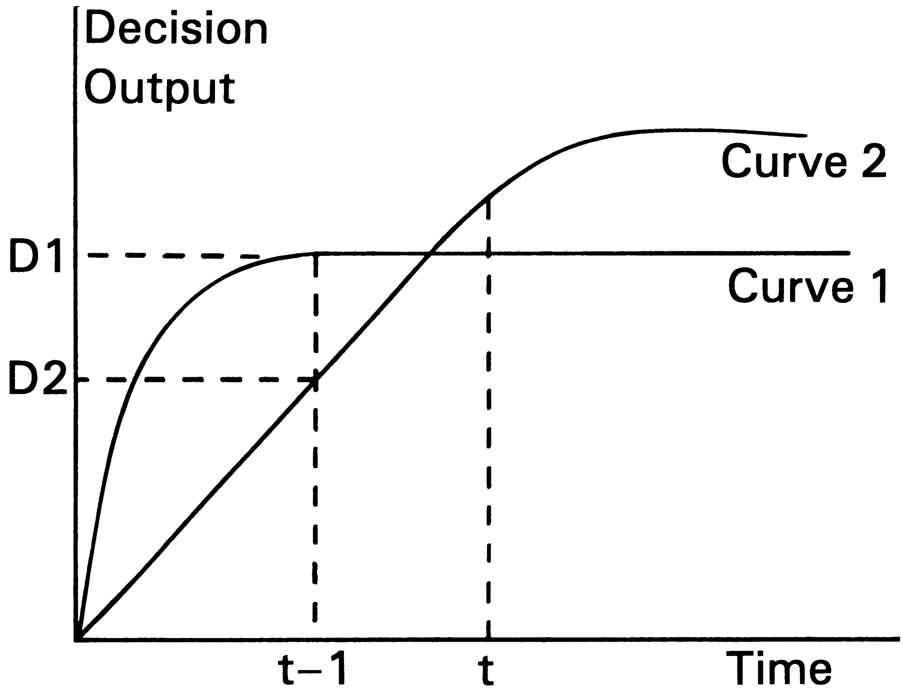

Figure 3. Learning curves for DSS

identifying all of the costs of a decision support system is not easy. Recognizing the problems associated with cost-benefit analysis, Keen [14] suggests using the technique of "value analysis" for economic evaluation of DSS.

Value analysis is comprised of the following sequence of steps. Initially, an operational list of benefits of a system is defined. A cost threshold is then determined based on what would be paid for the benefits obtained. If the expected cost of the system is acceptable, a prototype (first stage) of the decision support system is built. After the prototype is built, the benefits of the prototype are reviewed and revised, and an extended list of benefits is defined. Based on this review of the prototype, the capabilities of a full system are defined, costs are established, and acceptability of the expected benefits is tested. If acceptable, a full system (second stage) is built.

In summary, value analysis is based more on consideration of "value added" than cost displaced. By separating the evaluation into the two stages described above, the risks involved in development can be reduced, and the requirement for a fairly precise estimate of costs and benefits is simplified. The technique of value analysis fits the nature of the evolutionary development of DSS and focuses on the primary purpose of DSS, which is cut from the fabric of value added, not cost saved.

Feedback

For an organization to survive, it must receive constant feedback from its environment. More specifically, feedback about decision-making performance is perceived by users/managers and transmitted to designers. This feedback enables designers to change the decision support system, thereby reducing the system's deviation from its predefined objectives.

Appropriate feedback mechanisms, such as timely performance reporting, are a necessary condition for the implementation of any type of computer-based information system to have a reasonable chance of success. For implementation of DSS to be successful, more extensive feedback mechanisms are necessary. More specifically, it has been suggested [9] that organizations having high-level steering committees tend to be more successful in implementing CBIS than organizations having low-level or no steering committees. Moreover, those steering committees exhibiting top management support can be used as a means of positive feedback to encourage favorable users'/managers' attitudes toward DSS. The efficiency and effectiveness of these mechanisms for transmitting relevant feedback should be explicitly considered during the development process of the decision support system.

Summary

DEVELOPMENT OF DECISION SUPPORT SYSTEMS (DSS) is a complicated, dynamic process. DSS development should be recognized as meaning more than just the replacement of clerical operations by machine. DSS aim at unstructured tasks at managerial and strategic levels in organizations. Their development involves not only technical but also behavioral and organizational problem solving.

We began this paper by defining a model of the key conceptual components to be evaluated and monitored throughout the development of DSS, irrespective of the nature or sequence of implementation steps taken. For each component in the model we argued that the development of DSS should be thought of in a different way than the development of traditional computer-based information systems (CBIS). In summary form our rationale leads us to the following conclusions about the key factors in DSS development:

1. Because the settings for DSS are less structured than those of CBIS, potential users often find it difficult to assess the relevance of information elements that might enter into their decisions. Therefore a special effort must be made to value the importance of information elements that form the input bases for DSS. We describe a procedure that produces an importance index for all information maintained by CBIS.

2. DSS demand greater user-system collaboration than CBIS. DSS store and process more information and are expected to fit the decision styles of users/managers better than CBIS. The design characteristics of DSS user-system interface should include familiar data representations (e.g., tabular, graphic); support as many parts of the decision-making activities as possible; provide a variety of memory aids to assist users; be flexible enough to allow decision-makers to work in their idiosyncratic ways; and be under direct, personal control of the user.

3. In contrast to the benefits that accompany the existence of a computer-based information system, the benefits of a decision support system are only realized if the expertise of users/managers and designers has a common focus. We also know that

because it is unwise to assume a high level of cooperation among users/managers and designers, explicit action must be taken to induce cooperation. Given the validity of the characterization of the implementation process of a decision support system suggested in the paper (i.e., a game), this action might include making the payoff to users/managers and designers a function of the level of cooperation and supplementing this payoff by establishing a top management coordinating function and having each group report to it.

4. The design of a decision support system can be portrayed as a special case of a general process wherein authority is delegated or a fixed amount of resources is committed by top management. Whenever a fixed amount of resources is committed to a project, designers can behave as agents and attempt to maximize their utility by reducing their effort level. This shirking behavior can be avoided by making designers' payoff an increasing function of their effort level above some minimum resource commitment.

5. The importance of the interaction of existing organization structures with the development and implementation of a decision support system should not be underestimated. At the outset, organizational variables that are not fully controllable and that could affect the success of the decision support system's development efforts should be identified, made known to top management, and monitored throughout the implementation process.

6. In comparison to CBIS, DSS normally cause a greater redistribution of information and a correspondingly larger distribution of power both among and within departments. Designers of DSS should be mindful of the possible counterproductive activities of users/managers in response to this redistribution and be prepared at the outset to respond by devising appropriate implementation strategies and tactics.

7. Unlike CBIS, at any point in time the quality of output of a decision support system may be an incomplete measure of its effectiveness. The missing attribute that should be factored into the effectiveness measure is where users of the system are on the learning curve. Especially for DSS that are difficult for users to learn, the level of decision output should be evaluated periodically, and the sign and magnitude of the change in decision output, rather than the absolute level of the decision output, should be emphasized.

8. Because the nature of DSS development is different than CBIS development, in that DSS development is more iterative, adaptive, and evolutionary, choosing a proposal that accomplishes the design of a decision support system in the most efficient way is likely to be both inappropriate and unfeasible. To evaluate the economics of a decision suport system, a value analysis is advocated in place of the traditional cost-benefit analysis. Value analysis not only reduces the risks involved in development of the decision support system but also simplifies the requirement for a reasonably precise estimate of the costs and benefits associated with the development project.

9. To be of value to an organization, DSS must be more responsive than CBIS to their environment. For a decision support system to be responsive to its environment, its adaptive ability will be directly related to the quality and quantity of feedback its designers receive about its decision-making performance. While feed-

back mechanisms normally associated with CBIS, such as timely performance reporting, should be in place during the implementation process, research indicates that a systematic relationship exists between the level of the steering committee and the success of the implementation process. In other words, all things being equal, the probability of successful implementation of the decision support system increases if members of its steering committee are drawn from higher levels of management.

In conclusion, the above factors are not meant to be a detailed, step-by-step recipe to be followed in developing a decision support system. Instead, our intent is to present a perspective of behavioral and organizational considerations which, to the inexperienced decision support designer, can be easily overshadowed by technical concerns and restrictions inherent in traditional computer-based information systems development approaches.

REFERENCES

1. Ackoff, R. L. Management misinformation systems. *Management Science*, 1967, 14, 4, 147–56.

2. Ackoff, R. L. Towards a system of systems concepts. *Management Science*, 1971, 17, 11, 661–71.

3. Akoka, J. A framework for decision support systems evaluation. *Information and Management*, 1981, 4, 3, 1–23.

4. Allison, G. T. *Essence of Decision*. Boston, Mass.: Little Brown, 1971.

5. Bardach, E. *The Implementation Game: What Happens After a Bill Becomes a Law*. Cambridge, Mass.: MIT Press, 1977.

6. Carlson, E. D. An approach for designing decision support systems. *Proceedings, 11th Hawaii International Conference on Systems Sciences*, 1978, 76–96.

7. Davis, G., and Olson, M. H. *Management Information Systems: Conceptual Foundations, Structure, and Development* (2nd Ed.). New York: McGraw-Hill, 1984.

8. Dewaele, M. Managerial style and the design of decision aids. *Omega*, 1978, 6, 5–13.

9. Ein-Dor, P., and Segev, E. Organizational context and the success of management information systems. *Management Science*, 1978, 24, 10, 1064–77.

10. Galbraith, J. Organization design: an information processing view. *Interface*, 1974, 4, 3, 28–39.

11. Ginzberg, M. J. A prescriptive model for system implementation. *Systems, Objectives, Solutions*, 1981, 1, 33–46.

12. Keen, P. G. W. Computer based decision aids: the evaluation problem. *Sloan Management Review*, 1975, 16, 3, 17–29.

13. Keen, P. G. W. Information system and organizational change. *Communications of the ACM*, 1981, 24, 1, 24–33.

14. Keen, P. G. W. Value analysis: justifying decision support system. *Proceedings, Fourth International Conference on Systems Science*, 1981, 1, 556–79.

15. Keen, P. G. W., and Morton, M. S. *Decision Support Systems: An Organizational Perspective*. Boston, Mass.: Addison Wesley, 1978.

16. Kennedy, M. H., and Mahapatra, S. Information analysis for effective planning and control. *Sloan Management Review*, 1975, 16, 2, 71–83.

17. Kosaka, T., and Hirouchi, T. An effective architecture for decision support systems. *Information and Management*, 1982, 5, 1, 7–17.

18. Langefors, B. Discussion of "Determining management information needs: a comparison of methods." *MIS Quarterly*, 1977, 1, 4, 53–56.

19. Leavitt, H. J. Applying organizational change in industry: a structural, technological, and humanistic approach. In J. G. March (ed.). *Handbook of Organizations*. Chicago: Rand McNally, 1965.

20. Miller, G. The magical number seven, plus or minus two: some limits on our capacity for processing information. *The Psychological Review*, 1965, 63, 2, 52–60.

21. Neuman, S., and Hadas, M. DSS and strategic decision. *California Management Review*, 1982, 22, 1, 77–84.

22. Newell, A., and Simon, H. S. *Human Problem Solving*. Englewood Cliffs, N.J.: Prentice Hall, 1972.

23. Robey, D. Cognitive style and DSS decision: a comment on Huber's paper. *Management Science*, 1983, 29, 5, 580–82.

24. Robey, D., and Farrow, D. User involvement in information system development: a conflict model and empirical test. *Management Science*, 1982, 28, 1, 73–85.

25. Rouse, W. Human computer interaction in the control of dynamic systems. *Computing Surveys*, 1981, 13, 1, 77–99.

26. Simon, H. A. *The New Science of Management Decision*. New York: Harper and Row, 1960.

27. Tversky, A., and Kahneman, D. Judgement under uncertainty: heuristic and bias. *Science*, 1974, 185, 1124–31.

28. Wetherbe, J. C. *Systems Analysis and Design: Traditional, Structured, and Advanced Concepts and Techniques* (2nd Ed.). St. Paul: West Publishing, 1984.

29. Wildavsky, A. *The Politics of the Budgetary Process* (2nd Ed.). Boston: Little Brown, 1974.

30. Zmud, R. W. *Information Systems in Organizations*. Glenview, Ill.: Scott Foresman, 1983.