VALUE ANALYSIS:

Justifying Decision Support Systems

Peter G. W. Keen

October 1980

CISR No. 64
Sloan WP No. 1160-80

Center for Information Systems Research
Massachusetts Institute of Technology
Alfred P. Sloan School of Management
50 Memorial Drive
Cambridge, Massachusetts, 02139
617 253-1000

VALUE ANALYSIS:

Justifying Decision Support Systems

Peter G. W. Keen

October 1980

CISR No. 64
Sloan WP No. 1160-80

CONTENTS

1. Introduction

Decision Support Systems (DSS) are designed to help improve the effectiveness and productivity of managers and professionals. They are interactive systems frequently used by individuals with little experience with computers and analytic methods. They support, rather than replace, judgement in that they do not automate the decision process nor impose a sequence of analysis on the user. A DSS is in effect a staff assistant to whom the manager delegates activities involving retrieval, computation and reporting. The manager evaluates the results and selects the next step in the process. Table 1 lists typical DSS applications.[1]

Traditional cost-benefit analysis is not well-suited to DSS. The benefits they provide are often qualitative; examples cited by users of DSS include the ability to examine more alternatives, stimulation of new ideas and improved communication of analysis. It is extraordinarily difficult to place a value on these. In addition, most DSS evolve. There is no "final" system; an initial version is built and new facilities added in response to the users' experience and learning. Because of this, the costs of the DSS are not easy to identify.

The decision to build a DSS seems to be based on value, rather than cost. The system represents an investment for future effectiveness. A useful analogue is management education. A company will sponsor a five-day course on strategic planning, organizational development or management control systems on the basis of perceived need or long-term value. There is no attempt to look at payback period or ROI, nor does management expect a direct improvement in earnings per share.

This report examines how DSS are justified and recommends Value Analysis (VA), an overall methodology for planning and evaluating DSS proposals. Section 2 illustrates applications of DSS. Key points are:

1) A reliance on prototypes
2) The absence of cost-benefit analysis
3) The evolutionary nature of DSS development
4) The nature of the perceived benefits

Section 3 relates DSS to other types of innovation. It seems clear that innovation in general is driven by "demand-pull" -response to visible, concrete needs- and not "technology push".

Section 4 briefly examines alternative approaches to evaluation: cost-benefit analysis, scoring techniques and feasibility studies. They all require fairly precise estimates of and trade-offs between and benefit and often do not handle the qualitative issues central to DSS development and innovation in general. The final part of the report defines Value Analysis.

The overall issue the paper addresses is a managerial one:

1) What does one need to know to decide if it is worthwhile building a DSS?
2) How can executives encourage innovation while making sure money is well spent?
3) How can one put some sort of figure on the value of effectiveness, learning or creativity?

It would be foolish to sell a strategic planning course for executives on the basis of cost displacement and ROI. Similarly, any effort to exploit the substantial opportunity DSS provide to help managers do a better job must be couched in terms meaningful to them. This requires a focus on value and a recognition that qualitative benefits are of central relevance. At the same time, systematic assessment is essential. The initial expense of a DSS may be only in the $10,000.00 range but this still represents a significant commitment of funds and scarce programming resources. The methodology proposed here is based on a detailed analysis of the implementation of over twenty DSS. It is consistent with the less formal approaches most managers seem to use in assessing technical innovations. Value analysis involves a two-stage process:

1) Version 0: This is an initial, small-scale system which is complete in itself but may include limited functional capability. The decision to build Version 0 is based on:

   a) An assessment of benefits, not necessarily quantified;

   b) A cost threshold -- is it worth putting at risk this amount of money to get these benefits?

   In general, only a few benefits will be assessed. The cost threshold must be kept low, so that this decision can be viewed as a low-risk research and development venture and not a capital investment.

2) Base System: This is the full system, which will be assessed if the tiral Version 0 has successfully established the value of the proposed concept. The decision to develop it is based on:

a) Cost analysis: what are the costs of building this larger system?

b) Value threshold: what level of benefits is needed to justify the cost? What is the likelihood of this level being attained?

A major practical advantage of this two-stage strategy is that it reduces the risks involved in development. More importantly, it simplifies the trade-off between costs and benefits, without making the analysis simplistic. It is also a more natural approach than traditional cost-benefit analysis; until value is established, *any* cost is disproportionate.

2. Decision Support Systems

The DSS applications shown in Table 1 cover a range of functional areas and types of task. They have many features in common:

1) They are non-routine and involve frequent ad hoc analysis, fast access to data, and generation of non-standard reports
2) They often address "what-if?" questions: for example, "what if the interest rate is X%?" or "what if sales are 10% below the forecast?"
3) They have no obvious correct answers; the manager has to make qualitative trade-offs and take into account situational factors.

The following examples illustrate these points:

1) GADS: in designing school boundaries, parents and school officials worked together to resolve a highly-charged political problem. A proposal might be rejected because it meant closing a particular school, having children cross a busy highway or breaking up neighborhood groups. In a previous effort involving redistricting, only

TABLE 1 Examples of DSS Applications

DSS	APPLICATIONS	BENEFITS
GADS Geodata Analysis Display System	Geographical resource allocation and analysis; applications include sales force territories, police beat redesign, designating school boundaries	Ability to look at more alternatives; improved teamwork; can use the screen to get ideas across; improved confidence in the decision
PMS Portfolio Management System	Portfolio investment management	Better customer relations; ability to convey logic of a decision; value of graphics for identifying problem areas
IRIS Industrial Relations Information System	Ad hoc access to employee data for analysis of productivity and resource allocation	Ad hoc analysis; better use of "neglected and wasted" existing data resource; ability to handle unexpected short-term problems
PROJECTOR	Strategic financial planning	Insight into the dynamics of the business; broader understanding of key variables
IFPS Interactive Financial Planning System	Financial modelling, including mergers and acquisitions, new product analysis, facilities planning & pricing analysis	Better and faster decisions; saving analysts' time; better understanding of business factors; leaveraging managing skills
ISSPA Interactive Support System for Policy Analysts	Policy analysis in state government; simulations, reporting and ad hoc modelling	Ad hoc analysis; broader scope; communication to/with legislators; fast reaction to new situations
BRANDAID	Marketing planning, setting prices and budgets for advertising, sales force, promotion, etc.	Answering "what if?" questions, fine-tuning plans, problem-finding
IMS Interactive Marketing System	Media analysis of large consumer database; plan strategies for advertising	Helps build and explain to clients rationale for media campaigns; ad hoc and easy access to information

one solution had been generated, as opposed to 6 with GADS over a 4-day period. The interactive problem-solving brought out a large number of previously unrecognized constraints such as transportation patterns and walking times and parents' feelings.

2) BRANDAID: a brand manager heard a rumor that his advertising budget would be cut in half. By 5 pm he had a complete analysis of what he felt the effect would be on this year's and next year's sales.

3) IFPS: A model had been built to assess a potential aquisition. A decision was needed by 9 am. The results of the model suggested the acquisition be made. The senior executive involved felt uneasy. Within one hour, the model had been modified and "what if" issues assessed that led to rejection of the proposal.

4) ISSPA and IRIS: data which had always been available but not accessible were used to answer ad hoc, simple questions. Previously no one bothered to ask them.

These characteristics of problems for which DSS are best suited impose design criteria. The system must be:

1) <u>Flexible</u> to handle varied situations

2) Easy to use so it can be meshed into the manager's decision process simply and quickly

3) Responsive: it must not impose a structure on the user and must give speedy service

4) Communicative: The quality of the user-DSS dialogue and of the system outputs are key determinants of effective uses especially in tasks involving communication or negotiation. Managers will use computer systems that mesh with their natural mode of operation. The analogy of the DSS as a staff assistant is a useful one.

Many DSS rely on prototypes. Since the task the system supports is by definition non-routine, it is hard for the user to articulate the criteria for the DSS and for the designer to build functional specifications. An increasingly popular strategy is thus to use a langauge such as APL, an application generator or end-user language. These allow an initial system to be delivered quickly and cheaply. It provides a concrete example that the user can react to and learn from. It can be easily expanded or modified. The initial system, Version 0, clarifies the design criteria and specifications for the full DSS. Examples of this two-phase strategy include:

1) ISSPA: built in APL. Version 0 took 70 hours to build and contained 19 commands. The design process began by sketching out the user-system dialogue. New user commands were added as APL functions. 10 of the 48 commands were requested by users and several of the most complex ones entirely defined by users.

2) AAIMS: an APL-based "personal information system" for analysis of 150,000 time series. The development was not based on a survey of user requirements nor on any formal plan. New routines are tested and "proven" by a small user group.

3) IRIS: a prototype was built in 5 months and evolved over a one year period. An "Executive langauge" interface was defined as the base for the DSS and a philosophy adopted of "build and evaluate as you go".

4) CAUSE: There were 4 evolutionary versions; a phased development was used to build credibility. The number of routines were expanded from 26 to 200.

There have been several detailed studies of the time and cost needed to build a DSS in APL. A usable prototype takes about 3 weeks to deliver. A full system requires another 12-16 weeks.[2]

End-user langauges similarly allow fast development. One such DSS "generator" is Execucom's IFPS (Interactive Financial Planning System), a simple, English-like language for building strategic planning models. The discussion below is based on a survey of 300 IFPS applications in 42 companies.[3] The models included long-range planning, budgeting, project analysis, evalution of mergers and aquisitions.

The average IFPS model took 5 days to build and contained 360 lines (the median was 200). Documented specifications were developed for only 16%. In 66% of the cases, an analyst simply responded to a manager's request and got something up and running quickly. Cost-benefit analysis was done for 13% and only 30% have any objective evidence of "hard" benefits. 74% of the applications replace manual proceedures. (Given that most of the responding companies are in the Fortune 100, this indicates the limited degree to which managers in the planning funcitons make direct use of computers.)

Most DSS are built outside data processing, generally by individuals who are knowledgeable about the application area. Table 2 gives figures on where requests for IFPS applications came from and how they are built.

Table 2 IFPS Development Process

	Data Processing	Staff Analyst	Middle Management	Top Management
Who requested the application	0	4	30	66
Who built it	3	53	22	22
Who uses the terminal	0	70	21	9
Who uses the output	0	6	42	52

The IFPS users were asked to identify the features of the language that contributed most to the success of the DSS. In order of importance, these are:

1) Speed of response
2) Ease of use
3) Package features (curve-fitting, risk analysis, what-if?)
4) Sensitivity analysis
5) Time savings

The evolutionary nature of DSS development follows from the reliance on prototypes and fast development. There is no "final" system. In most instances, the system evolves in response to user learning. A major difficulty in designing DSS is that many of the most effective uses are unanticipated and even unpredictable. Examples are:

1) PMS: the intended use was to facilitate a portfolio-based rather than security-based approach to investment. This did not occur, but the DSS was invaluable in communicating with customers.
2) GPLAN: the DSS forced the users (engineers) to change their roles from analysts to decision makers.
3) PROJECTOR: the intended use was to analyze financial data in order to answer preplanned questions and the actual use was as an educational vehicle to alert managers to new issues.

Table 3 Relative Use of DSS Operators (PMS)

Operator	percentage of use by each manager A	B	C	D	E	F	percentage of use by all users
TABLE	22	22	38	22	76	57	47
SUMMARY	40	10	30	8	0	38	17
SCAN	0	26	5	24	0	0	4
GRAPH	14	4	13	30	5	0	8
DIRECTORY	2	0	0	0	1	4	1
Others	22	38	14	16	18	1	23

Usage is also very personalized, since the managers differ in their modes of analysis and the DSS is under their own control. For example, 6 users of PMS studied over a 6 month period differed strongly in their choice of operators (See Table 3).[4]

The benefits of DSS vary; this is to be exptected given the complex situational nature of the tasks they support and their personalized uses. The following list shows those frequently cited in DSS case studies, together with representative examples.[5] Table 4 summarizes the list.

1) Increase in the number of alternatives examined:

- sensitivity analysis takes 10% of the time needed previously
- 8 detailed solutions generated versus 1 in previous study
- previously took weeks to evaluate a plan; now takes minutes, so much broader analysis
- users could imagine solutions and use DSS to test out hypotheses
- "no one had bothered to try price/profit options before"

2) Better understanding of the business

- president made major changes in company's overall plan, after using DSS to analyze single acquisition proposal

- DSS alerted managers to fact that an apparently successful marketing venture would be in trouble in six month's time
- DSS used to train managers; gives them a clear overall picture
- "now able to see relationships among variables"

3) Fast Response to unexpected situations

- a marketing manager faced with unexpected budget cut used the DSS to show that this would have a severe impact later
- helped develop legal case to remove tariff on petroleum in New England states
- model revised in 20 minutes, adding risk analysis; led to reversal of major decision made 1 hour earlier

4) Ability to carry out ad hoc analysis

- 50% increase in planning group's throughput in 3 years
- the governor's bill was published at noon "and by 5 pm I had it fully costed out"
- "I can now do QAD's: quick-and-dirties"
- system successfully used to challenge legislator's statements within a few hours

5) New insights and learning

- quickened management's awareness of branch bank problems
- gives a much better sense of true costs
- identified underutilized resources already at analyst's disposal
- allows a more elegant breakdown of data into categories heretofore impractical
- stimulated new approaches to evaluating investment proposals

6) Improved communication

- used in "switch presentations" by advertising agencies to reveal shortcomings in customer's present agency
- can explain rationale for decision to investment clients
- improved customer relations
- "analysis was easier to understand and explain. Management had confidence in the results".
- "it makes it a lot easier to sell (customers) on an idea"

7) Control

- permits better tracking of cases
- plans are more consistent and management can spot discrepancies

- can "get a fix on the overall expense picture"
- standardized calculation procedures
- improved frequency and quality of annual account reviews
- better monitoring of trends in airline's fuel consumption

8) Cost savings

- reduced clerical work
- eliminated overtime
- stay of patients shortened
- reduced turnover of underwriters

9) Better decisions

- "he was forced to think about issues he would not have considered otherwise"
- analysis of personnel data allowed management to identify for the first time where productivity gains could be obtained by investing in office automation
- increased depth and sophistication of analysis
- analysts became decision-makers instead of form preparers

10) More effective team work

- allowed parents and school administrators to work together exploring ideas

- reduced conflict: managers could quickly look at proposal without prior argument

11) Time savings

- planning cycle reduced from 6 man-days spread over 20 elapsed days to 1/2 a day spread over 2 days
- "substantial reduction in manhours" for planning studies
- (my) time-effectiveness improved by a factor of 20"

12) Making better use of data resource

- experimental engineers more ready to collect data since they knew it would be entered into a usable system
- "more cost-effective than any other system (we) implemented in capitalizing on the neglected and wasted resource of data"
- allows quick browsing
- "puts tremendous amount of data at manager's disposal in form and combinations never possible at this speed".

Table 4 adds up to a definition of managerial productivity. All the benefits are valuable but few of them quantifiable in ROI or payback terms.

Table 4 - DSS Benefits

	Easy to measure?	benefit can be quantified in a "bottom line" figure?
1. Increase in number of alternatives examined	Y	N
2. Better understanding of the business	N	N
3. Fast response to unexpected situations	Y	N
4. Ability to carry out ad hoc analysis	Y	N
5. New insights and learning	N	N
6. Improved communication	N	N
7. Control	N	N
8. Cost savings	Y	Y
9. Better decisions	N	N
10. More effective teamwork	N	N
11. Time savings	Y	Y
12. Making better use of data resource	Y	N

In few of the DSS case studies is there any evidence of formal cost-benefit analysis. In most instances, the system was built in response to a concern about timeliness or scope of analysis, the need to upgrade management skills, or the potential opportunity a computer data resource or modelling capability provides. Since there is little a priori definition of costs and benefits, there is litttle a posteriori assessment of gains. A number of DSS failed in their aims; but where they are successful, there is rarely any formal analysis of the returns. Many of the benefits are not proven. In managerial tasks there is rarely a clear link between decisions and outcomes, and a DSS can be expected to <u>contribute</u> to better financial performance but not directly cause it. In general, managers describe a successful DSS as "indispensable" without trying to place an economic value on it.

3. The Dynamics of Innovation

DSS are a form of innovation. They represent:

1) A relatively new concept of the role of computers in the decision process;

2) An explicit effort to make computers helpful to managers who on the whole have not found them relevant to their own job, even if they are useful to the organization as a whole;

3) A decentralization of systems development and operation and often a bypassing of the data processing department;

4) The use of computers for "value-added" applications rather than cost displacement.

There is a large literature on the dynamics of technical innovations in organizations.[6] Its conclusions are fairly uniform and heavily backed by empirical data. Table 5 lists some key findings.

Table 5

Dynamics of Innovation	References
Innovations are 1) value driven 2) a response to a perceived problem	Utterback, Rogers & Shoemaker, von Hippel
Early adopters differ from later ones in that they are iconoclastic entrepreneurs willing to accept risk	Haug, Roberts
Informal processes are central to innovation and they require 1) gatekeepers 2) product champions	Allen, Rogers, Chakrabarti
New applications come from the marketplace, not from technologists	Utterback, von Hippel
Cost is a secondary issue in innovation	Haywood
Uncertainty is reduced by "trialability", ease of understanding and clear performance value	Rogers & Shoemaker

Surveys of the use of computer planning models support these conclusions. In nine cases studied[7] the decision to adopt planning models was based on:

1) Comparison with an ongoing system; this involves examining either a manual or partially computerized system and deciding that some change is desirable;
2) Comparison with a related system, such as a successful planning model in another functional area;
3) Initiation of a low-cost project;
4) Comparison with competitors' behavior; this use of a "reference model" reduces the need to estimate the impact of a model not yet constructed on improved decisions and performance.

Even in traditional data processing applications, the emphasis on value rather than cost is common. A survey of all the proposals for new systems accepted for development in a large multinational company found that even though cost-benefit analysis was formally required, it was used infrequently.[8] The two main reasons for implementing systems were:

1) Mandated requirements, such as regulatory reports;

2) Identification or one or two benefits, rarely quantified.

Traditional cost-benefit analysis is obviously effective for many computer-based systems. It seems clear, however, that it is not used in innovation. This may partly be because innovations involve R&D; they cannot be predefined and clear specifications provided. There is some evidence that there is a conflict in organizations between groups concerned with performance and those focused on cost. In several DSS case studies, the initiators of the system stress to their superiors that the project is an investment in R&D, not in a predefined product.

Surveys of product innovations consistently find that they come from customers and users, rather than centralized technical or research staff. Well over three-quarters of new products are initiated by someone with a clear problem looking for a solution.[9] Industrial salesmen play a key role as "gatekeepers" bringing these needs to the attention of technical specialists. Even in the microprocessor industry, the majority of products are stimulated in this way by "demand-pull" not by "technology-push".[10]

Case studies indicate that DSS development reflects the same dynamics of innovation as in other technical fields. Table 6 restates Table 5 in relation to DSS.

Table 6

Dynamics of DSS Innovation

Innovations are value-driven	Main motivation for DSS is "better" planning, timely information, ad hoc capability, etc.
Early adopters differ from late adopters	DSS are often initiated by line managers in their own budgets; once the system is proven other departments may pick it up.
Informal processes are central	DSS development usually involves a small team; key role of intermediaries knowledgeable about the users and the technology for the DSS; data processing rarely involved; frequently DSS are "bootleg" projects.
Cost is a secondary issue	Costs are rarely tracked in detail; DSS budget is often based on staff rather than dollars; little charge out of systems (this may reflect item below).
Uncertainty reduced by trialability, ease of understanding, clear performance value	Use of prototypes; emphasis on ease of use.

4. Methodologies for Evaluating Proposals

There are three basic techniques used to evaluate proposals for computer systems in most organizations:

1) Cost-benefit analysis and related ROI approaches; this views the decision as a capital investment;
2) Scoring evaluation; this views it in terms of weighted scores;
3) Feasibility study; this views it as engineering.

Each of these is well-suited to situations that involve hard costs and benefits and that permit clear performance criteria. They do not seem to be useful -- or at least used -- for evaluating innovations or DSS.

Cost-benefit analysis is highly sensitive to assumptions such as discount rates and residual value. It needs artificial and often arbitrary modifications to handle qualitative factors such as the value of improved communication and improved job satisfaction. Managers seem to be more comfortable thinking in terms of perceived value and then asking if the cost is reasonable. For example, expensive investments on training are made with no effort at quantification. The major benefits of DSS listed in Table 4 are mainly qualitative and uncertain. It is difficult to see how cost-benefit analysis of them can be reliable and convincing in this context.

Scoring methods are a popular technique for evaluating large-scale technical projects, such as the choice of a telecommunications package, especially when there are multiple proposals with varying prices and capabilities. Scoring techniques focus on a list of desired performance characteristics. Weights are assigned to them and each alternative rated. For example:

characteristic	weight	score for alternative	weighted score
response time	.30	15	4.5
ease of use	.20	20	4.0
user manual	.10	17	1.7

Composite scores may be generated in several ways: mean rating, pass-fail, or elimination of any alternative that does not meet a mandatory performance requirement. Cost is considered only after all alternatives are scored. There is no obvious way of deciding if alternative A with a cost of $80,000 and a composite score of 67 is better than B with a cost of $95,000 and a score of 79.

Feasibility studies involve an investment to identify likely costs and benefits. They tend to be expensive and to focus on defining specifications for a complete system. They rarely give much insight into how to build it and assume that the details of the system can be laid out in advance. DSS prototypes are a form of feasibility study in themselves. They are a first cut at a system. Some designers of DSS point out that Version "0" can be literally thrown away. Its major value is to clarify design criteria and establish feasibility, usefulness, and usability. The differences between a prototype and a feasibility study are important:

1) The prototype moves the project forward, in that a basic system is available for use and the logic and structure of the DSS already implemented;
2) The prototype is often cheaper, if the application is suited to APL or an end-user language;
3) The feasibility study is an abstraction and the prototype concrete; since DSS uses are often personalized and unanticipated, direct use of the DSS may be essential to establishing design criteria.

There is no evidence that any of these methods are used in evaluating DSS, except occasionally as a rationale or a ritual. More importantly, almost every survey of the dynamics of innovation indicates that they do not facilitate innovation and often impede it.

5. Value Analysis

The dilemma managers face in assessing DSS proposals is that the issue of qualitative benefits is central but they must find some way of deciding if the cost is justified. What is needed is a systematic methodology that focuses on:

1) Value first, cost second;
2) Simplicity and robustness; decision makers cannot (and should not have to) provide precise estimates of uncertain, qualitative future variables;
3) Reducing uncertainty and risk;
4) Innovation, rather than routinization.

The methodology recommended here addresses all these issues. It relies on prototyping which:

1) Factors risk, by reducing the initial investment, delay between approval of the project and delivery of a tangible product;
2) Separates cost and benefit, by keeping the initial investment within a relatively small, predictable range.

If an innovation involves a large investment, the risk is obviously high. Since estimates of costs and benefits are at best approximate, the decision maker has no way of making a sensible judgement. Risk is factored by reducing scope. An initial system is built at a cost below the capital investment level; the project is then an R&D effort. It can be written off if it fails. By using the DSS one identifies benefits and establishes value. The designer is also likely to learn something new about how to design the full system. The prototype accomplishes the same things as a feasibility study, but goes further in that a real system is built.

The benefits of a DSS are the incentive for going ahead. The complex calculations of cost-benefit analysis are replaced in value analysis by simple questions that most managers naturally ask and handle with ease:

1) What exactly will I get from the system?
   - -- It solves a business problem;
   - -- It can help improve planning, communication and control;
   - -- It saves time.
2) If the prototype costs $X, do I feel that the cost is acceptable?

Obviously the manager can try out several alternatives; "If the prototype only accomplishes two of my three operational objectives, at a lower cost of $Y, would I prefer that?" The key point is that value and cost are kept separate and not equated. This is sensible only if the cost is kept fairly low. From case studies of DSS, it appears that the cost must be below $20,000 in most organizations for value analysis to be applicable.

This first stage of value analysis is similar to the way in which effective decisions to adopt innovations are made. It corresponds to most managers' implicit strategy. The second stage is a recommendation; there is no evidence in the literature that it is widely used, but it seems a robust and simple extension of Version "0". Once the nature and value of the concept has been established the next step it to build the full DSS. The assessment of cost and value needs now to be reversed:

1) How much will the full system cost?
2) What threshold of values must be obtained to justify the cost? What is the likelihood they will occur?

If the expected values exceed the threshold, no further quantification is required. If they do not, then obviously there must either be a scaling down of the system and a reduction in cost or a more detailed exploration of benefits.

Value analysis follows a general principle of effective decision making -- simplify the problem to make it manageable. A general weakness of the cost-benefit approach is that it requires knowledge, accuracy and confidence about issues which for innovations are unknown, ill-defined and uncertain. It therefore is more feasible to

1) Establish value first, then test if the expected cost is acceptable.

2) For the full system, establish cost first then test if the expected benefits are acceptable.

Instead of comparing benefits against cost, value analysis involves merely identifying relevant benefits and testing them against what is in effect a market price: "Would I be willing to pay $X to get this capability?" It is obviously essential that the benefits be accurately identified and made operational. "Better planning" is not operational. The key question is how would one know that better planning has occurred? The prototype is in effect an experiment in identifying and assessing it.

FIGURE 1 VALUE ANALYSIS

STAGE I

ESTABLISH VALUE:

Define operational list of benefits:
e.g., solves urgent business problem
provides a flexible tool for recurrent analysis
makes planning data quickly accessible
saves time in recurrent ad hoc reporting

↘

DETERMINE COST THRESHOLD:

Define maximum one would be ready to pay to gain the benefits
Determine if a prototype can be built that delivers the necessary capabilities

↙

BUILD VERSION 0:

Define an architecture that permits the full system to be evolved from the initial version 0
Define routines for prototype

↘

STAGE II

ASSESS PROTOTYPE:

Review benefits; revise and extend list
Review desired and obtainable capabilities
Define functional capabilities of full system

↙

ESTABLISH COST OF VERSION 1:

How much will the full system cost?

↘

DETERMINE BENEFIT THRESHOLD:

What level of benefits must be obtained to justify the investment in the full system?
What is the likelihood these can be obtained?

↓

BUILD VERSION 0

↓

EVOLVE VERSION N

Review usage, evaluate new capabilities desired or obtainable
Establish cost
Determine benefit threshold

Figure 1 illustrates the logic and sequence of value analysis. The specific details of the method are less important than the overall assumptions, which have important implications for anyone trying to justify a DSS whether as a designer or user. Marketing a DSS requires building a convincing case. Figure 1 can be restated in these terms:

1) Establish value; the selling point for a DSS is the specific benefits it provides for busy managers in complex jobs.
2) Establish cost threshold; "trialability" is possible only if the DSS is relatively cheap and installed quickly. If it costs, say, $200,000, it is a capital investment, and must be evaluated as such. This removes the project from the realm of R&D and benefits as the focus of attention to ROI and tangible costs and inhibits innovation.
3) Build Version "0"; from a marketing perspective this is equivalent to "strike while the iron is hot." Doing so is possible only with tools that allow speedy development, modification and extension.
4) Assess the prototype; for the marketer this means working closely with the user and providing responsive service.

Two analogies for DSS have been mentioned in this report: The staff assistant and management education. The strategy used to justify DSS depends upon the extent to which one views such systems as service innovations and investments in future effectiveness as opposed to products, routinization and investment in cost displacement and efficiency. The evidence seems clear -- DSS are a potentially important innovation. Value is the issue, and any exploitation of the DSS approach rests on a systematic strategy for identifying benefits, however qualitative, and encouraging R&D and experimentation.

REFERENCES

1. Detailed descriptions of each DSS shown in Table 1 can be found in the relative reference:

   GADS: Keen & Scott Morton, Carlson & Sutton
   PMS: Keen & Scott Morton, Andreoli & Steadman
   IRIS: Berder & Edelman in Carlson (ed.)
   PROJECTOR: Keen & Scott Morton, Meador & Ness
   IFPS: Wagner
   ISSPA: Keen & Gambino
   BRANDAID: Keen & Scott Morton, Little
   IMS: Alter

   Other DSS referred to in this paper are:

   AAIMS: Kless in Carlson (ed.), Alter
   CAUSE: Alter
   GPLAN: Haseman in Carlson (ed.)

2. See Grajew & Tolovi for a substantiation of these figures. They built a number of DSS in a manufacturing firm to test the "evolutive approach" to development.

3. IFPS is a proprietary product of Execucom, Inc., Austin, Texas. The survey of IFPS users is described in Wagner.

4. See Andreoli & Steadman for a detailed analysis of PMS usage.

5. This list (pp. 12-16) is taken verbatim from Keen "Decision Support Systems and Managerial Productivity Analysis".

6. See bibliography to the National Science Foundation working draft "Innovation Processes and their Management: A Conceptual, Empirical, and Policy Review of Innovation Process Research" by L.G. Tornatzky, et al., October 19, 1979.

7. See Blanning, "How Managers Decide to Use Planning Models", Long Range Planning, Volume 13, April 1980.

8. See Ginzberg.

9. See Utterback.

10. See von Hippel.

BIBLIOGRAPHY

Alter, Steven, Decision Support Systems: Current Practice and Continuing Challenges, Addision-Wesley, 1980.

Andreoli, P. and J. Steadman, "Management Decision Support Systems: Impact on the Decision Process", Master's thesis, MIT, 1975.

Blanning, R., "How Managers Decide to Use Planning Models", Long Range Planning, Volume 13, April 1980.

Carlson, Eric D., ed., "Proceedings of a Conference on Decision Support Systems", Data Base, Volume 8, Number 3, Winter 1977.

Chakrabarti, A.K., "Role of Champion in Product Innovation", California Management Review, Volume 17, pp. 58-62, 1974.

Grajew, Jakow, and Jose Tolovi, Jr., "Conception et Mise en Oeuvre des Systems Interactifs d'aide a la Decision: l'approche Evolutive," Doctorial dissertation, Universite des Sciences Sociales de Grenoble, Institut d'Administration des Entreprises, 1978.

Ginzberg, M.J., "A Process Approach to Management Science Implementation", PhD dissertation, MIT, 1975.

Keen, Peter G.W., "Decision Support Systems and Managerial Productivity Analysis", paper presented at the American Productivity Council Conference on Productivity Research, April 1980.

__________ and Thomas Gambino, "The Mythical Man-Month Revisited: Building A Decision Support System in APL", paper presented at the APL Users Meeting, Toronto, September 1980.

__________ and Michael S. Scott Morton, Decision Support Systems: An Organizational Prespective, Addison-Wesley, 1978.

Rogers, E., "Re-Invention During the Innovation Process", in Radnor, Feller & Rogers, eds., "The Diffusion of Innovations: An Assessment", Final Report, Northwestern University, 1978, Grant Number PRA-7680388, National Science Foundation.

__________ and F.F. Shoemaker, Communication of Innovations: A Cross-Cultural Approach, Free Press, New York, 1971.

Utterback, J.M., "Innovation in Industry and the Diffusion of Technology", Science, Volume 183, February 1974.

von Hippel, E., "The Dominant Role of Users in the Scientific Instrument Innovation Process", Research Policy, July 1976.

Wagner, G.R., "Realizing DSS Benefits with the IFPS Planning Language", paper presented at the Hawaii International Conference on System Sciences, January 1980.

Date Due

DEC [illegible]	NOV 13 1998	
FEB 20 1985	AUG 31 2001	
APR 19 1985		
JUN 5 1985		
[illegible] 5 1985		
MAR 25 '87		
APR 08 '88		
FEB 24 [illegible]		
DEC 18 1989		
MAY 13 1997		

Lib-26-67