WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

The Impact of Knowledge and Technology Complexity on Decision Making Software Develpment

Marc H. Meyer[1]
Kathy F. Curley[2]

May, 1993 WP # 89-93

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

The International Center for Research on the Management of Technology

The Impact of Knowledge and Technology Complexity on Decision Making Software Develpment

Marc H. Meyer[1]
Kathy F. Curley[2]

May, 1993 WP # 89-93

Sloan WP# 3583-93

1 Northeastern University and MIT
2 Northeastern University

Sloan School of Management
Massachusetts Institute of Technology
38 Memorial Drive, E56-390
Cambridge, MA 02139-4307

Abstract

This paper reports the findings of a field study on managing the development of software applications used for decision making. The study is based upon a sample of 108 systems that have been operational for at least one year. Collectively, these systems represent a broad spectrum of complexity with respect to decision making and computer technology. At one extreme are stand-alone systems with simple decision making logic. At the other extreme are systems with logic for highly complex decision domains. Some systems are widely distributed throughout firms and linked to suppliers, distributors, or customers.

The study gathers data regarding the origins of systems ideas, development costs, project durations, management controls, and the composition of the software development teams. Its develops measures to assess and categorize systems in terms of two dimensions of complexity: that of decision making or the knowledge embodied in a system, and that of the computer technology used to build and deploy a system. Successful approaches to systems development are found to be contingent on these two dimensions of complexity.

(MANAGEMENT OF TECHNOLOGY, SOFTWARE , KNOWLEDGE-BASED SYSTEMS)

Introduction

What is the effect of decision making or "knowledge" complexity on managing software development? To what extent does this type of complexity determine how managers approach the development of a new information system? What are the implications for management when high levels of knowledge complexity are combined with high levels of complexity in the computer technology of a new system? Can these two forms of complexity be measured systematically across different systems?

Consider the experience of a large insurance company that began building an expert systems for underwriting life insurance applications in 1987.[1] This was a firm in which the centralized data processing department had traditionally designed, staffed, and managed the development of all large-scale "strategic" software applications. Development of the underwriting expert system, however, prompted an important departure from this conventional practice. In this case, business executives actively participated in setting the focus and planning the higher level design of the system.

The leadership of this firm established a new organizational unit to house the development effort. This new department, which later was incorporated as a subsidiary, acted as a melting pot for different types of domain expertise and computer expertise. Medical researchers and senior insurance underwriters were formally assigned to the project, as were programmers recruited from throughout the firm. The "systems analysts" were individuals with medical degrees who had become proficient in knowledge engineering. Programmers recruited from across the firm worked closely with them to build systems. As the the opportunity to market the new system to the rest of the insurance industry was recognized, the new unit developed selling expertise. Medical researchers and senior underwriters were formally assigned to the project. These individuals, representing many discrete areas of insurance and operations expertise, built applications "platforms" consisting of core decision making code, databases, and administration procedures tailored to

---

[1] Lincoln National Life Company, Fort Wayne, Indiana, described in Meyer, DeTore, Siegel, and Curley (1992).

produce specific systems for different market niches. Indeed, this firm's experiences and approaches challenged much of our own thinking on how best to organize and manage the development of decision making systems. The "knowledge complexity" of the application had certainly affected organization, staffing, development time, and cost of the project in ways that were both similar and different to what one might consider a technically complex application.

We began this research to determine the role of decision making complexity in the management of software development. Just as managers have come to associate the technology complexity of a new system with the time, budget, and staffing needed to build it, we hypothesized that the embodied knowledge complexity of an application would have important implications for managing its development. The following pages report a framework and research findings that provide an empirical foundation for a contingent approach to applications development.[2]

Assessing Knowledge and Technology Complexity

We content that management of software development must be approached with an understanding of the embodied complexity of the target application. Different management approaches work for different types of systems. Understanding the type of system is imperative for formulating the appropriate development strategy. We believe that this understanding can be facilitated by a system of classification that considers a development project along two dimensions of complexity: 1) the domain-specific knowledge and decision making complexity supported by an application, and 2) the

---

[2] Prior studies on managing software developing includes Brooks (1975), who consider the problems in team communications, productivity, and effectiveness that were incurred as team size increased for the IBM 360 systems software development effort. Cusumano (1989,1991) reported the effectiveness of process and technical standardization in the "software factory" approach employed by many large Japanese companies. Benbasat and Vessey (1980) and Zmud (1980) examined the the difficulties in estimating the length of development time and cost of large scale software efforts. Ruth (1988) found that the expected development costs of knowledge-based systems could be used to revise or anticipate the size and composition of the development teams.

complexity of the underlying computer technology used to develop, integrate, and diffuse an application throughout an organization.

Figure 1 shows these two dimensions of complexity and our own generic labels for the types of systems they give rise to. Four actual cases from our research sample illustrate each quadrant in that Figure:

- Personal Productivity Systems: a stand-alone PC-based application made by a large airline company requires only a handful of "rules" to effectively assess the tax implications of employee travel from foreign countries back "home" to England.

- Knowledge Intensive Systems: a stand-alone PC-based application developed by a large engineering services firm to provide decision assistance for scheduling the operations of a large chemical plant. This system has logic that incorporates supplier management and logistics, manufacturing cost and quality, and production scheduling.

- Technology Intensive Systems: a sales force information system made by a computer manufacturer automatically downloads new product and component information to a global salesforce. The scope and frequency of distributing these data makes the technical aspect of this application highly complex.

- Strategic Systems: the insurer described above combines both knowledge and technology intensity into a system that provides both automatic underwriting for "simple" cases and decision assistance to underwriters for complex ones. It incorporates rich logic from underwriting, medical, and financial disciplines, and is delivered in a distributed systems platform that accesses different types of information needed for decision making from multiple external sources.

Insert FIGURE 1 about here

Our initial effort to develop measures for classifying a particular system along these two dimensions of complexity was made through a pilot study that involved intensive discussion with software managers of six large firms. This initial study focused on knowledge-based systems in which expert system shells and knowledge engineering were key aspects of development. We attempted to isolate "what is important" in each complexity dimension by having the software managers assign weights of importance to their own lists of important activities, and then comparing weights between these lists to work towards a final list in an iterative process. This took approximately six months. The study was subsequently expanded to approximately thirty other firms with operational expert system applications.

While the study of expert systems was the foundation of our research, we believed that the utility of understand knowledge and technology complexity extended far beyond applications with formal "knowledge bases" developed with expert system shells or languages. Prompted by this belief, we broadened the context and scope of the method so that it might assess knowledge and technology complexity for any information system deemed by the company to encompass significant decision making logic. In the cumulative sample presented here, there are systems that employ shells and special languages, but there are also sophisticated applications that are useful in decision making capability developed in more general purpose environments such as C, C++, or simpler spreadsheet style tools.[3]

Assessing Decision Making Complexity

Decision theorists (Shannon 1948) differentiate between the decision processes, or logic, and the information, or data, used in that decision making. There are three basic components at the core of decision making: the decision maker's knowledge of the paradigms and heuristics of a given field(s); the "raw" information used by the decision-maker; and the interpretation and

---

[3] This framework is adapted from the work of Meyer and Curley (1991). Support in the literature for focusing on these two types of complexity for understanding the differences in information systems may be found in Newell and Simon (1972), Davis (1984), and Gory and Scott-Morton (1971). Clancy (1985) also used "heuristic" or decision-making style classification methods on a sample of early expert systems.

synthesis of that information by applying domain-specific logic to resolve uncertainty and come to partial or complete decisions. We amplified these three basic into a set a variables to assess decision making complexity for a given application.

The Breadth of Decision making Domain(s). The breadth of the embodied decision making areas in a system is a measure of the number of specific distinct fields of expertise employed by the decision makers whose reasoning is incorporated into an expert system. By examining the domain content for a specific decision making process with knowledge engineers and experts, we identified the number of specific domains or distinct fields of expertise modeled into the system. [4] The tax system described above has only one domain, tax law. In contrast, the insurance underwriting system incorporates medical science, actuarial science, and financial analysis. We scored systems as follows:

Description	Score
A single domain	1
Two distinct domains	2
More than two distinct domains	3

The Depth of Decision making Domain(s). We assessed the depth of embodied domains by combining the level of advanced educational training with the duration of applied work experience required of the key decision makers.[5] When teams of experts were activity involved on a project, we used the highest level of education and work experience

[4] Davis (1984) argued that "the knowledge of how to use the rules must be separated from the rules themselves, dividing the expert system into an inference engine and a knowledge base" .

[5] Much has been written about the characteristics of experts, their methods of acquiring skill, and their application of that skill to problem solving (Newell and Simon 1986, Schank and Childers 1984). Dreyfus and Dreyfus (1986) describe five stages of skill acquisition from novice to expert based on field experiments with a variety of professionals including, doctors, nurses, and chess masters. Their research indicates that expertise requires both mastery of a body of context-free rules and theories as well as the know-how and intuitive ability gained by experience.

respectively. Similarly, if more than one domain was identified, then domain depth was assessed for each of the domains, and the final score was derived by taking the highest of the individual domain depth scores. We combined these data to produce a score as follows:

Depth	Description	Score
Common	To be an expert, the individual(s) did not need to have the knowledge gained through an advanced degree in the given domain(s) (i.e. Ph.D., MD.), nor substantial applied work experience (10 years or more in the specific domain(s)).	1
Deep	To be an expert, the individual(s) required either A) an advanced degree and less than 10 years work experience in the specific domain(s), or B) more than 10 years work experience but no advanced degree.	2
Deepest	To be an expert, the individual(s) required both an advanced degree and more than 10 years work experience.	3

The Rate of Change of Decision making Domain(s). This measures the extent to which decision makers become obsolete and ineffective in their problem-solving ability without ongoing domain-based education and/or research. Some domains are relatively fixed; others change only occasionally; yet others continuously advance. High rates of domain change increase knowledge complexity for a given expert system (Partridge, 1987). In a decision making process where more than one domain was identified, we used the highest score for the collective set of domains.

Rate	Description	Score
Low	Someone who was an expert 5 years ago who added no new knowledge could still be an expert today.	1
Moderate	An expert of 2 years ago, and who added no new knowledge, could still be an expert today.	2
High	Remaining an expert requires updating of knowledge on a continuous, i.e. yearly basis.	3

Decision making Domain Penetration. The degree of penetration by a system into areas of decision making is a measure of the completeness of computerization for each of the specific domains embodied in a system. A particular decision maker may be a world authority in his or her field, but the system in which he or she participated may only capture a small portion of that expertise. Decision makers involved in the sample systems were asked to assess the degree of domain penetration for each of the embodied domains on the scale below. If multiple domains were involved, each domain was evaluated for this measure, and then the highest score was employed subsequent calculations.

Degree of Penetration	Score
Little	1
Partial	2
Moderate	3
Substantial	4
Complete	5

Comprehensiveness of Decision Outputs : By cataloging decision making outputs, we sought to assess the extent to which a system solves its targeted problems or decision areas. Reflecting prior work in the field of decision theory (Shannon 1948, Newell and Simon 1972), we categorized specific outputs (i.e. problem resolution information displayed, printed, or in some other way communicated as in an electronic message) as being either:

- Problem Diagnosis
- Recommended Actions
- Actual Solutions
- Hypothesis Testing

For example, a number of systems in our sample only determined the absence of important information for decision making. Others systems would take steps to automatically gather that data. Others would make decisions, such as determining the credit worthiness of a loan applicant or finding the reason why a machine broke down. Other systems, such as the plant scheduling system, would allow users to do "what if" testing.

A system was evaluated for comprehensiveness depending on the presence of these generic categories of outputs in the visual and/or printed information outputs of the system, and scored as follows:

Solution Outputs	Score
Just One Type	1
Any Two Types	2
Any Three Types	3
All Four Types	4

Breadth of Information Inputs: We examined the "raw" information inputs for a given expert system. For example, a system developed by an engineering services company to advise utilities on coal purchasing decisions uses data regarding coal characteristics, plant configuration, operating costs, and power generation experience to produce a purchase recommendation. The scoring of breadth of information inputs was:

Breadth of Information Inputs	Score
One or two inputs	1
Three of four inputs	2
Five or more inputs	3

Interpretation Required of Information Inputs: In many instances (such as the utility company coal buying system), the "raw" information inputs referred to above is certain. In other instances, however, it is ambiguous and requires that the problem-solver interpret and in other ways provide additional meaning to these data. Medical examinations, for example, must be interpreted by medical diagnosis assistance systems, or by underwriting systems in life and health insurers.

Level of Ambiguity	Description	Score
Low	Little or no additional interpretation is required.	1
Moderate	Some interpretation of the inputs is required.	2
High	A high degree of additional interpretation is required for the information inputs to be useful in decision making.	3

To produce overall assessments of knowledge complexity, the data for each of these measures for a given system were tabulated using an additive model whereby all variables were summed with equal weights. Since these measures had different scales, we created the equal weights by multiplying to create a common denominator and then normalizing on a scale to a maximum of 100 in the following manner:

Knowledge Complexity = ((Breadth of Decision making Domains* 20) +
(Depth of Decision making Domains* 20) +
(Rate of Change of Decision making Domains* 20) +
(Decision Making Domain Penetration * 12) +
(Comprehensiveness of Decision Outputs * 15) +
(Breadth of Information Inputs * 20) +
(Information Inputs Interpretation * 20)) / 4.2

The technological complexity of a system was also defined by a set of measures. No attempt was made to ascertain whether a company's choice of technology tools was optimal; our only objective was to document and classify the technologies actually used by a given system.

Diversity of Platforms. The diversity of platforms is a measure of the degree to which a system has been ported across different computer architectures and operating systems. Creating a system for multiple platforms, or for a very special purpose computer, increases the difficulty of the development effort. For measurement purposes, we differentiated between development and delivery platforms, and gathered information only for the latter--i.e. the types of hardware on which the operational or production system was made to function for users. Additionally, we distinguished between computer architecture (i.e. Intel versus VAX versus Sparc versus IBM 4331, for example), and operating system type (DOS versus Unix versus VMS versus CICS, (Deitel 1984)). These data were scored as follows:

Platform Range	Score
A single hardware and operating system environment	1
Two hardware and operating system environments	2
Three of more hardware and operating system environments	3

The Diversity of Technologies. The presence of multiple development technologies and the integration required between them can substantially increase technical complexity and require additional specialization in the programming team. For example, Bobrow et. al., 1986, Kerschbert 1987, and Pedersen 1988 have noted the past difficulty of integrating expert systems languages and shells into distributed, data intensive computing environments.

In this measure, we sought to evaluate the diversity of software tools used in developing a system. We factored out the presence of networking technology, which was set aside to be addressed separately. The following choices were used to catalogue technologies:

- Knowledge based development languages and shells
- Neural Networks
- Graphics (as in graphics programming libraries)
- Database management systems
- Spreadsheets
- Imaging
- Voice

For the scoring of this variable, the number of technologies embodied in the delivered systems was counted and incorporated into the technology complexity calculation. One basic technology tool was scored as 1. An additional point was assigned for each additional technology. The scores for this variable ranged from 1 to 7.

Database Intensity. The majority of the systems we have encountered used database management systems tools. Larger databases add to the complexity of the development effort, requiring more extensive logical design, database implementation, and in some cases transactions processing. We measured the cumulative size of the underlying reference databases accessed by the system in the course of processing. An initial dividing point of one megabyte of data for size seemed reasonable, since beyond this point developers must begin to worry about optimizing access methods, transaction logging, access synchronization, and rigorous backup and recovery mechanisms. We scored the data as follows:

Database Intensity	Score
less than 1 megabyte	1
1 to 10 megabyte	2
more than 10 megabyte	3

Networking Intensity. As with database management, distributed computing within an application adds to technological complexity. This measure examines the use of computer networking for accessing other applications or databases by a given system. We assessed networking intensity with the following scale:

Network Intensity	Score
Stand-alone Systems	1
Infrequent Networking	2
Regular Networking	3

A system operating on a single computer and not employing any type of computer network, either to receive data or send back results to another system, would be classified as "stand-alone". The use of computer networks for the purposes of interacting with other databases or applications on only a periodic basis would be classified as "infrequent." Many decision making systems receive information at the beginning of user sessions and then send results back to other systems at the end of sessions. Lastly, some advanced decision systems continuously exchange information with a variety of systems in the course of processing. These would receive the highest rating in this measure.

<u>The Scope of the Decision/Knowledge Base Programming Effort:</u>
Harmon, Maus, and Morrissey (1988) considered the degree of difficulty in encoding decision makers' expertise as a major aspect of systems development complexity. Our first thought was to measure person-days spent specifically in knowledge base (or its equivalent) development might provide insight into this difficulty. We soon found in the pilot study group that variations in programmer productivity between projects made person-days unreliable as a comparative measure.

We decided to gather data regarding: a) the number of "rules" (since many of the systems in the sample employed some form of rule specification as a basic element of logic), and b) the total size of the "knowledge base" (which in more complex systems will include "object" specifications as well as "rules," or in others, binary program modules containing simple "if-then" expressions in a language such as C). We differentiated between the knowledge base itself and associated databases used by or within the expert system (the database portion is considered in a prior variable). Our method for deriving an assessment was:

The Scope of the Decision/Knowledge base Programming Effort:	Score
Less than 500 Rules or less than 500k	1
Between 500 and 1500 rules or between 500k and 1.5 megabytes	2
More than 1500 rules or greater than 1.5 megabyte	3

<u>Diversity of Information Sources:</u> While examining the "raw" information inputs for a system, we also recorded their sources or

providers. Increasing the number of information sources involving computer access increases development complexity. For example, one information provider (internal or external to the organization) may provide a series necessary data through a single interface module. In other instances, a system may acquire many different types of data from many different sources, each needing its own access module. We score the variable as follows:

Diversity of Information Sources	Score
1 to 3 Computerized Data Sources	1
3 or 4 Computerized Data Sources	2
5 or more Computerized Data Sources	3

Diffusion of the System. The level of diffusion of an information system can increase the technical complexity of the development effort through distributed data sharing and program processing requirements. Knowledge and database updating for a system diffused throughout many departments of a company or across different sites can become the subject of major technical activity. The example of the widely distributed sales support system in the computer manufacturer mentioned earlier illustrates this.

Following Roger's (1962) diffusion module, a system may proceed through approximate levels of usage and types of users that range from limited to widespread. We assessed diffusion according to the following scale:

Level of Diffusion	Score
A single user	1
Two or three users in a single department	2
Three or more users in a single department	3
Many users in several departments	4
Company-wide usage	5

Systems Integration Effort This variable is a measure of the degree of programming and other technical effort required for integrating the expert system with existing information systems. For example, the XCON product configuration system developed by Digital Equipment Corporation is reported to have numerous linkages to systems in sales, product engineering, manufacturing, and field installation departments (Bachant and McDermott, 1984). Developing these linkages involves substantial effort and resources, sometimes beyond those expended for knowledge base development and programming. We asked participants to describe the level of systems integration required for their systems and scored their responses from 1 to 5 respectively.

Level of Systems Integration	Score
Not a factor	1
Some work	2
Moderate work	3
A major effort	4
The largest, most difficult job	5

We produced the overall assessment of technological complexity for a given system (with a maximum score of 100) as follows:

$$\text{Technology Complexity} = ((\text{Diversity of Platforms} * 10) + (\text{Technology Diversity} * 5) + (\text{Database Intensity} * 10) + (\text{Network Intensity} * 10) + (\text{Decision/Knowledge base Programming Effort} * 10) + (\text{Diversity of Information Sources} * 10) + (\text{Diffusion} * 6) + (\text{Systems Integration} * 6)) / 2.4$$

Methods for Applying the Classification Framework

This classification method was applied to a sample of 108 "successful" information systems. [6] The operational definition of a successfully developed system is one that had passed the design, prototyping, and testing stages, and had been fielded as a working application and used within an organization (and possibly by its suppliers, distributors, or customers) for the system's intended purpose.

Gleaning insights from failures can be just as useful as the study of successes. While we were able to get firms to identify some failures, in no case were the data for them sufficient for our research purposes. This is why we constrained the sample to installed, working successes.

The sample was gathered by convenience. We first sampled firms in the United States and England. Our structured questionnaire was translated into German by the Fraunhaufer Institute, and administered to firms in Europe. The research instrument was translated into Japanese by the Nikkei Intelligent Systems Journal and administered to more than 50 Japanese firms. We translated these surveys into English with the help of graduate students fluent in the language.

---

[6] The test study group of a half dozen firms was not included in the 108-member sample.

A broad range of industries is represented: financial services, engineering, transportation, sales, energy, health services, construction, and government. Brief descriptions are provided in Appendix 1. For example, underwriting expert systems for life insurance, property and casualty, and health underwriting were typical of insurance company systems. In banking, systems assisting in loan preparation and authorization were included. A variety of computer component assembly, machine tool configuration, and product testing systems were developed in the manufacturing arena. A number of systems were developed to assist in the selling of complex products. In fact, the sheer diversity of knowledge-oriented systems that we encountered reaffirmed the need to create a generic classification method as a means of finding common learnings across diverse experiences.

A structured questionnaire was developed to gather data for specific factors within the dimensions of knowledge and technological complexity.[7] The data gathering process involved the completion of interview forms by a number of key individuals associated with the respective systems: project managers, domain experts, knowledge engineers, and key computer programmers. The depth of the interviewing helped to insure the reliability of the information collected. The majority of the data gathered were concerned with objective and verifiable information. In the knowledge complexity dimension, we used the education level and years of work experience of domain experts who worked on a system as the facts behind our measurement. Similarly, the specific number of information inputs and solution outputs could be enumerated by project engineers. In the technology dimension, the number of hardware platforms, the size of databases, and the variety of different technologies used, the scope of the knowledge base programming effort, and the levels of systems integration and diffusion of the system proved reasonable to gather.

Insert Figure 2 about here

---

[7] The survey is computerized and the classification automated in an English language version for personal computers using Claris' FileMaker Pro.

Appendix 2 contains the knowledge and technology complexity scores for these systems. Figure 2 provides a visual mapping of the system on our framework based on these scores. The range for knowledge complexity in the sample was 33.10 to 100, with a median value of 57. The range for technology complexity was 27.92 to 80, with a median of 49. We categorized the sample into the four generic groups of systems shown earlier in Figure 1 using median scores for each dimension of complexity. The distribution of systems in each quadrant was as follows:[8]

Distribution of the Sample by Quadrant

Quadrant	Cases	% Of Sample
Low Knowledge/Low Technology	35	32
High Knowledge/Low Technology	19	18
Low Knowledge/High Technology	21	19
High Knowledge/High Technology	33	31

The classification of these systems was then used as the foundation examining differences in R&D management approaches for different levels of knowledge and technology complexity.

The Origin of Ideas

What did the research suggest about the sources of ideas, i.e., the origins of the software applications in the sample?

von Hippel (1986) has shown that "lead users" can be excellent sources of new product ideas for manufacturing equipment producers. In the case of the information systems, these "users" are the management and staff associated with the various line units within an organization. Similarly, the "manufacturer" or "producer" is represented by data processing personnel in

[8] Sensitivity testing was also performed by increasing the weight of each of the respective variables for each dimension of complexity by 25%, keeping the others variables constant, to discern differences in basic classification along the lines of Figure 2. No appreciable change in the number of cases within each of the four quadrants occurred.

a firm. "Suppliers" are those hardware, software, and computer services firms which provide the company with computers, software tools, and programming assistance. Our hypothesis was that as the knowledge complexity of systems increased, we would find line affiliated expert decision makers playing the role of von Hippel's "lead users".

We asked respondents to identify the primary origin of the system. The categories and observed frequencies are shown below:

Origin Category	Cases	% Of Sample
An "expert" in the decision making area inside your company working with an existing business department.	60	57%
A "programmer" or programming manager inside your company's general Data Processing department or one of its subsidiary groups.	35	33%
An outside "expert", from a customer, a competitor, or a consultant.	11	10%

We created a contingency table based on the four quadrants emerging from the classification method and the two internal sources of ideas (in the first two categories of the table above). The resulting chi-square of 7.75 was significant. ($p = .05$, $df = 3$)

	A decision making expert inside a business department	A computer professional from Corporate DP	
Personal Productivity Systems	17	13	30 31.58%
Knowledge Intensive Systems	13	3	16 16.84%
Technology Intensive Systems	8	11	19 20%
Strategic Systems	22	8	30 31.58%
	60 63.13%	35 36.84%	95

For systems initiatives involving complex, specific areas of decision making, experts in the line units were excellent sources for important application ideas. In terms of the framework, this applies to the Knowledge Intensive and Strategic Systems quadrants. Many of these systems started as prototypes created by "power users" on PC's with easy to use shells and database management systems, and these were used to a limited extent as Personal Productivity systems. Examples from the sample included a manufacturing quality control application for the manufacture of computer disk drives, an "intelligent" market research database application, and an insurance underwriting system.

Conversely, when knowledge complexity is low, and technology complexity is high, systems were more likely to be conceived by computer professionals (the Technology Intensive quadrant of the framework). Clever data processing staff initiated important applications within the confines of their own domain or business decision making understanding. A system developed for the Red Cross to schedule the recertification of the cardio-pulminary resusitation (CPR) program is an example of this.

When both types of complexity are low, there are few education or experiential impediments to either decision makers or programmers in conceiving new uses of computers. This is the realm of end-user computing, which is continuously empowering individuals with respect to the application of computers to their work.

Project Duration and Cost

What did the research suggest about the impact of knowledge and technology complexity on the duration and cost of systems development?

One would expect that the time and cost required to develop a system should typically increase with the embodied knowledge and technological complexity of the system. We also wanted to find which dimension of complexity might have a greater impact on duration and cost respectively. We expected that knowledge complexity would exhibit a strong association with longer project durations than would technical complexity in the creation of a first working version of a system. The reasons for this expectation is the time it takes a team to correctly understand and model multiple domains of expertise, to penetrate areas of expertise deeply, and to formulate strategies for resolving uncertainty in the information entered into the system.

The effect of higher levels of technological complexity on duration and cost, on the other hand, should be softened by the ever-improving productivity of software tools. The nature of the systems in our sample tended to utilize tool foundations that consisted of knowledge-based programming shells or environments, database management systems, and networking and communications tools. While these tools are most productively used in the hands of trained, experience computer professionals, they are clearly becoming easier and more productive in the hands of less specialized personnel doing applications development. In contrast to applications development, the experiences of the managers in our sample showed that applications integration, which means linking one of these knowledge-based systems to older administrative and/or decision making systems in the company, remains difficult, time consuming, and costly.

We recorded the development history timelines of each system, noting the formal project start date (defined as when the effort was formally budgeted by the organization), the completion of the working prototype of the system, the completion of the first release of the system (where the application was placed into "production mode", i.e. users employing the system on a regular basis), and the completion dates of subsequent fielded versions of the system. These data were then coded into the following six duration categories and scored on an ascending scale from 1 to 6, with 1 representing the shortest time and 6 representing the longest. We also computed the mean values of the interval categories to be used as class value marks in subsequent statistical analyses. The distribution of responses was:

Cumulative Project Time Categories	Category Mean	Time to develop First Fielded System: Frequency of Responses	Duration of Latest Fielded Version: Frequency of Responses
within 6 months	3 months	51	11
within 1 year	9 months	44	33
within 2 years	18 months	25	26
within 3 years	30 months	6	30
within 5 years	48 months	1	23
more than 5 years	69 months	0	4

Respondents also indicated cumulative project expenditures[9] (using the scale shown in the table below) and the cumulative elapsed time required to achieve a *fully integrated systems implementation* (using the time scale shown above). Managers felt more comfortable identifying a particular category interval than identifying discrete dollar figures. Additionally, interval categories facilitated the conversion of foreign currencies to dollar amounts.

Cumulative "To Date" Project Budget Categories	Category Means	First Fielded System Budget: Frequencies of Responses	Fielded System Total Budget: Frequencies of Responses
less than $100,000	$50,000	70	40
>= $100,000 and < $250,000	$175,000	26	34
>= $250,000 and < $500,000	$375,000	14	26
>= $500,000 and < $1 m	$750,000	6	11
>= $1m and < $2m	$1,500,000	5	5
>= $2m and < $5m	$3,500,000	2	5
> 5 m and < $10m	$7,500,000	0	1

9 Data from the European and Japanese systems were converted from their local currencies into appropriate dollar categories.

Multiple regressions were performed between project duration and cost respectively (using the interval category means) and the classification scores for knowledge and technological complexity. The results clearly show that both knowledge and technology complexity, as measured and determined above, are strong predictors of the time and cost of fielding an initial working version of a knowledge-based system.

The regression of project duration to knowledge and technology complexity was significant at $p = .0001$ ($n=101$, $df = 2$, $F = 10.71$). Increasing the level of knowledge complexity was also shown to have strong association with project duration. (Knowledge complexity: $\beta = .21$, t-value = 2.94, $p = .0041$; technology complexity: $\beta = .13$, t-value = 1.95, $p = .0546$). Modeling complex decision making simply cannot be rushed.

For the budget required to field a first working version of a system as a function of knowledge and technology complexity, the result was significant at $p = .0001$ ($n = 97$, $df = 2$, $F = 13.6$). In this case, however, technology complexity had a stronger association with project cost than knowledge complexity (knowledge complexity: $\beta = .9.31$, t-value = 2.02, $p = .0461$; technology complexity: $\beta = 15.59$, t-value = 3.45, $p = .0008$). Looking at the systems in our sample, it became evident that intensive systems integration and distributed database access were the most frequent causes of high technological complexity. Achieving these two goals required large data processing staffs working for many months to integrate the system with other administration type systems in the company. This created the large budgets.

Figure 3 shows three means for each of the four quadrants: the elapsed time required to complete the first working version of the system, the budget for doing this, and the total cumulative cost for the development and maintenance to date. Kruskal-Wallis tests performed on these data confirmed that the differences in these for project duration and cost between the four quadrants of our classification scheme were highly significant.[10]

---

[10] For project duration in developing the first working version, H corrected for ties was 12.8, $df = 3$, $p = .005$. For budget in developing the first working system, H was 22.18, $df = 3$, $p = .0001$. For

Insert Figure 3 about here

The data for budgets and time to fully integrated deployments of the applications in the sample showed little variation for first release data in the Quadrant I and Quadrant II (technologically uncomplex) applications. However, for Quadrant III applications, achieving full integration cost on average about $500,000 and 25.5 months. For Quadrant IV applications, full integration averaged more than $1 million and 30 months.

Organizational Locus of Project Control

What did the research show in terms of managerial control over the development systems? When should line units control these projects? When should data processing departments control development? Were hybrid forms of control present in the sample?

The locus of management control over technical projects for both product and process innovations has been framed in several basic ways in prior research. Ginzberg (1980) considered the organizational contingencies of information systems development in the accounting and financial disciplines. Roberts and Berry (1985) posited different organizational alternatives for new technical initiatives as a function of the technological and market applications newness. In each of these works, corporate R&D is contrasted with operating or line units in terms of the advantages and disadvantages of their respective control in certain situations.

In the case of knowledge-based systems, the "operating division" is the business department that will use the system. "Corporate R&D" is the firm's central data processing department. We began with two hypotheses about the locus of management control. The first hypothesis was that for successful systems development efforts, one would find that greater levels of knowledge

---

total cumulative budget, H was 34.02, df = 3, p = .0001. For total time required to achieve a fully integrated systems implementation, H was 14.02, df = 3, p = .003.

complexity would require business unit control. This would insure continuous and substantive access to decision making experts) and follows the general trend of decentralizing applications development to user departments.

Our second hypothesis was that greater levels of technology complexity would require management control by corporate DP. Such control makes it more likely that the project gets the programming and systems integration resources it needs . "Experts" in the decision making area participate on a part-time basis, suited to the more limited amounts of expertise required in this category of systems.

If both knowledge and technology complexity are low, then we believed that it would not matter who controlled the project so long as the individual user/developer was granted sufficient time to develop the system and used software tools appropriate to his or her own skill level.

Lastly, we believed that systems containing high levels of both knowledge and technological complexity would usually be controlled jointly by operating business units and corporate DP. In fact, we suspected that many firms would create new organizational units to build and maintain strategic systems. With separate budgets and their own reporting structures, these new units would make it easier for decision making experts, analysts, and programmers to work together in a manner and level of intensity sufficient to build the system. In some cases, these new units might become profit centers if the systems developed were marketable by the firm.

To test these ideas, we gathered data regarding the types of organizational entities that controlled the development of the system in the sample. [11] We defined "control" as the deciding control over objectives, budgets, and project performance. These data were categorized into three categories: existing business departments, existing corporate DP departments, and new "hybrid" departments formed to have both decision making experts and computer

[11] Three of the 108 systems in the sample did not provide this information.

professionals. In some cases, these new departments were specialized "Artificial Intelligence" applications groups, which, while ultimately reporting to the vice president of information systems, enjoyed substantial autonomy and were highly intrapraneurial. When development was actually managed by an outside vendor contracted by corporate DP (which occurred in four instances in the sample) or a business unit, the case was deemed to be controlled by whomever did the contracting. A contingency table was then generated based on the four groups from the classification method and these three organizational types. The resulting X^2 (15.21, df = 6, p = .02) shows a significant relationship between embodied complexity and organizational locus of control over development.

	Existing Business Department	Corporate DP	A New Department	
Personal Productivity Systems	13	8	12	33 31.43%
Knowledge Intensive Systems	12	4	3	19 18.1%
Technology Intensive Systems	4	10	6	20 19.05%
Strategic Systems	6	13	14	33 31.43%
	35 33.33%	35 33.33%	35 33.33%	105

For the Personal Productivity systems, business unit control is a reasonable finding because such systems tend to be PC-based, stand-alone applications that focus on single fields of expertise with only moderate levels of what we call "domain penetration". For a number of applications in this quadrant, it is also interesting to note that their respective companies had formed "AI" departments to explore new applications. The new departments often made a series of more simple applications before striking on a "hit" project with high levels of knowledge and technology complexity, and large potential impact on the firm.

As one crosses over into Knowledge Intensive systems, access to decision making experts becomes an essential ingredient for success. The strong shift in our data towards business unit control confirms this. Technology Intensive systems show a similarly strong shift to corporate DP control when the mastery of complex software tools and systems integration assumes primary importance.

Lastly, our data show that a company might consider forming new departments for the development of Strategic Systems, systems where decision making experts and programming experts must work in an intensive and sustained manner. In our sample, these projects tended to be "big ticket," large scale efforts. These new units, with their own space, their own structure of authority, and performance reviews normally provide far greater teamwork than do so-called "co-operative" projects characterized by periodic interchanges by persons from different physical and organizational locations. (Allen 1977)

Some Strategic Systems had unexpected results. For example, in the insurance company mentioned earlier, a new department was formed to develop a risk management system for the company's own insurance business. That system has since been sold to over thirty other insurance companies. In another case, a new department was formed in an engineering services firm to build a family of highly specialized expert systems for the manufacturing sector. A spin-off firm from an Asian investment firm has developed a highly successful personal portfolio and investment management system. These are cases of clear intrapraneurship.

Respondents also articulated the problems they encountered in development. When DP departments controlled the development of Knowledge Intensive applications, the most frequently cited problem was gaining the necessary "access to the experts." Lacking such access, the systems suffered from incorrect approaches to problem solving, inappropriate user interfaces, or failed to provide sufficient depth in handling the "hard decision making cases." Similarly, when business units controlled systems that were technically complex, they sometimes encountered "technical show-stoppers" that would take significant time and effort to circumvent. Frequently, this meant failing to share information or otherwise connect to other systems within or external to the company. The lack of technical capability within business units limited the effectiveness of the application.

Team Composition

How large were the development teams and what types of individuals worked on the systems in our sample? Were team size and composition associated with different levels of knowledge and technology complexity?

As one might expect, the relationship between the team size needed to produce the first working versions of these applications and knowledge and technological complexity mirrored that of project cost and complexity. The range of team size in the sample was from 1 to 25 full-time person equivalents. The multiple regression between team size and the two types of complexity produced a result significant at $p = .0001$. ($n = 108$, $df = 2$, $F= 17.67$, $p= .0001$). Team size increased with embodied complexity. A Kruskal Wallis test on the difference in team size between the four quadrants of our framework was also significant at $p = .0001$ ($n = 108$, groups = 4, $df = 3$, H corrected for ties = 22.58, $p= .0001$).

With respect to the types of individuals assigned to development teams, the literature has frequently cited particular "roles" in development (Harmon, Maus, and Morrissey, 1988, Feigenbaum, McCordick and Nii 1988, Hart 1986, Roberts and Fusfeld 1982). These include decision making experts (often called domain experts), knowledge engineers (systems analysts who model the logic and structure information flows in the application), and programmers. We hypothesized that greater levels of specialization in staffing roles would be required to build the most complex types of systems, i.e.. the Strategic Systems having both high levels of knowledge and technological complexity.

We observed four forms of team composition with respect to the roles mentioned above. First, on some projects, there were distinctly separate domain experts, knowledge engineers or systems analysts, and programmers.

Second, in a limited number of projects, all three roles were performed by the same exceptional, hybrid individual(s). Third, in some cases decision making experts were sufficiently trained in computer systems design to do their own knowledge engineering, and worked with programmers to implement their designs. Lastly, in the majority of cases, programmers had taken additional training in knowledge-based systems development and were able to perform both the knowledge-engineering and programming roles with the domain experts. The frequencies for the sample are shown in the following table.

Team Roles	Observed Frequency
The roles of domain experts, knowledge engineers, and programmers were all performed by different individuals	25.93%
The domain experts did their own knowledge engineering, but the programming was performed by different individuals	11.11%
The knowledge engineering and programming were performed by the same individual(s), but domain experts were different individuals	53.7%
All three roles were performed by the same individual(s)	9.26%

While the fourth alternative (all roles performed by the same individual) was reserved exclusively for Personal Productivity Systems (low knowledge, low technology), there was no significant distribution of the other forms of team composition among the other three quadrants in our framework. The personnel that management had available for project assignment (i.e. the mix of skills in individuals) had more bearing on these roles than on embodied complexity.

We believed that two things would happen as systems exhibit higher levels of knowledge complexity:

1. the participation of decision making experts in development would itself increase, both in terms of the number of experts and the formality of their assignment to the project; and

2. the level of domain familiarity on the part of systems analysts would also increase.

The latter would be necessary for systems analysts to understand what the experts were talking about, and to correctly and fully model the decision making process.

Respondents provided information regarding the number of decision making experts involved in systems development. The largest number of experts assigned to work on any system in the sample was 12. The regression between the number of decision making experts and the knowledge complexity of systems as assessed with our classification method was significant at $p = .02$ ($n = 107$, $df = 1$, $F = 5.13$, $p = .02$).

Respondents also indicated the type of the participation of the "lead" expert(s), be it informal and irregular participation (22%), part-time but regular participation (65%), or full-time and formal assignment for the duration of the project (13%). Full-time participation of decision making experts was more likely to be present for the more knowledge intensive applications. The fact that more than three quarters of the sample required the regular or full-time participation of decision making experts to successfully develop knowledge-based systems is itself an important finding for structuring such projects. An organization's best decision making experts are typically very busy performing their regular jobs. The cost of taking these experts "off-line" for significant periods of time to work on software applications is a cost that management must consider before embarking on decision-intensive systems initiatives.

For those systems in which the knowledge engineering role was performed by individual(s) other than the decision making experts, a clear relationship between higher levels of knowledge complexity and domain familiarity on the part of the systems analysts emerged. For those systems in the first and third categories in the table above, we asked respondents to identify the level of domain familiarity on the part of knowledge engineers at the outset of the project according to a five point Likert scale (ranging from no familiarity to very high levels of familiarity). The resulting regression between knowledge complexity and domain familiarity on the part of the knowledge engineers was significant at $p = .004$. ($n = 86$, $df = 1$, $F = 8.75$, $p = .004$). Several examples clarify this point. In a sophisticated manufacturing equipment problem diagnosis system, the knowledge engineer had a masters degree in mechanical engineering and substantial work experience; he needed this background to understand the models and processes of the two PhD. experts assigned to the project. Similarly, in the insurance underwriting example, the two knowledge engineers both had medical degrees.

What were the differences among the programming staffs in the sample? Except for the simplest of the Personal Productivity Systems (low knowledge complexity and low technology complexity), the applications in our sample required programming by professional programmers. We asked respondents to categorize the full-time equivalent assignments of programming personnel for the development of the first working versions of their systems according to three types of individuals: applications programmers (using expert systems shells, C++, LISP, or some other language), database management systems specialists, and systems integration specialists (communications and applications integration with other systems).

Programming staffs for systems with lower levels of technological complexity (being Personal Productivity Systems and Knowledge Intensive Systems) tended to be dominated by "applications programmers." The technologically complex systems (Technology Intensive Systems and Strategic Systems) tended to have fairly equal representation of applications programmers, database specialists, and systems integrators in their respective teams. Staffing differences in the four quadrants of the classification framework for these three areas of specialization were found to be significant. The Kruskal Wallis results (corrected for ties) are summarized in the table below (Group = 4, df=3).

Full time equivalents	H corrected for ties	p
Applications programmers	34.22	.0001
Database specialists	16.20	.001
Systems integration specialists	9.34	.024

Summary of Results and Thoughts for Management

Figure 4 summarizes the findings of the research. Knowledge and technology complexity, when measured with our method, were associated with significantly different management approaches for development, ranging from who might best control projects to the time, money, and staffing needed to create working systems. We learned that effective management of applications development is contingent upon embodied complexity.

Insert Figure 4 about here

These findings may have predictive value for managers considering new systems initiatives. Readers may estimate the complexity of a new initiative using our measures and position it within the four quadrant framework. They might then consider the experience of firms in this study with whom they share similar levels of embodied knowledge and technology complexity.

Many managers are thinking about how to identify and leverage the core competencies and capabilities of their firms to improve existing areas of business and to create new ones. (Prahalad and Hamel 1990, Quinn, Doorley and Paquette 1990, Meyer and Utterback 1993). Decision making skills and processes are certainly among the core capabilities of the firm, and managers increasingly value the information and knowledge in their organization as a basis for creating new or better products and to provide better service. We believe that the combination of these two thought architectures, competencies-based management and value-added information, serve as a rich foundation for initiating the development of ever more powerful decision making applications and make the present research highly relevant to such endeavors.

References

Allen T., **Managing the Flow of Technology**, MIT Press, Cambridge, MA, 1977.

Bachant, H. and McDermott, J. , "R1 Revisited: Four Years in the Trenches," **AI Magazine** (5:3), Fall 1984, 21-32.

Benbasat, I. , and Vessey, I., "Programming and Analyst Time/Cost Estimation", **MIS Quarterly**, June 1980, 31-43.

Bobrow, D.G., Mittal, S., and Stefik, M.J., "Expert Systems: Perils and Promise", **Communications of the ACM**, September 1988, 880-894.

Brooks, F.P., **The Mythical Man Month**, Addison-Wesley, Reading, MA, 1975.

Clancy W.J., "Heuristic Classification", **Artificial Intelligence**, (27), 1985, 289-350.

Cusumano, M., "The Software Factory: A Historical Interpretation", **IEEE Software**, March 1989, 23-30.

Cusumano, M., and Kemerer, C., " A quantitative analysis of US and Japanese software-engineering practice and performance", **Management Science**, 1991.

Davis, R., "Amplifying Expertise with Expert Systems", **The AI Business**, Winston, P. and Prendergast, K. , (Eds.), MIT Press, Cambridge, MA, 1984, 17-40.

Deitel, H.M., **An Introduction to Operating Systems**, Addison-Wesley Publishing Co., Reading, MA, 1984.

Dreyfus, H. and Dreyfus, S., **Mind Over Machine**, The Free Press, New York, NY, 1986.

Feigenbaum, E., "Knowledge engineering: The Applied Side of Artificial Intelligence", **Annals of the New York Academy of Sciences** (426), 1984, 91-107.

Feigenbaum, E., McCordick, P., and Nii, H., **The Rise of the Expert Company**, Times Books, New York, NY, 1988.

Ginzberg, M.H., "An Organizational Contingencies View of Accounting and Information Systems Implementation", **Accounting, Organizations, and Society**, (5) 1980,369-382.

Gory, G.A. and Scott Morton, M., "A Framework for Management Information Systems", **Sloan Management Review**,(13:1), 1971, 55-70.

Harmon, P., Maus, R., and Morrissey, W., **Expert System Tools and Applications**, Wiley and Sons, New York, NY, 1988.

Hart, A., **Knowledge Acquisition for Expert Systems**, McGraw-Hill, New York, NY, 1986.

Hayes-Roth, F., Waterman, D., and Lnat, D., (Eds.), **Building Expert Systems**, Addison-Wesley Publishing Co., Reading MA, 1983.

Kerschbert, L., (ed) **Proceedings from the First International Conference on Expert Database Systems**, Benjamin Cummings, Menlo Park, CA, 1987.

Leonard-Barton, D. and Sviokla, J., "Putting Expert Systems to Work", **Harvard Business Review** (66:2), March-April 1988, 91-98.

Liebowitz, J., (Ed.) **Expert Systems for Business & Management**, Yourdon Press, Englewood Cliffs, N.J.,1992.

McGraw, K. and Harbison-Briggs, L., **Knowledge Acquisition: Principles and Guidelines**, Prentice Hall, Englewood Cliffs, NJ., 1989.

Meyer, M.H. and Curley, K. F., "An Applied Framework for Classifying the Complexity of Knowledge-Based Systems", **MIS Quarterly**, (15:4), December 1991, 455-475.

Meyer, M.H., DeTore, A., Siegel, A., and Curley, K., "The Strategic Use of Expert Systems for Risk Management in the Insurance Industry", **Expert Systems With Applications**, (5), Summer 1992, 15-24.

Meyer, M.H. and Utterback, J.M., "The Product Family and the Dynamics of Core Capability," **Sloan Management Review**, (34:3), Spring 1993, 29-47.

Moore, G.C., and Benbasat, "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation", **Information Systems Research** (2:3) 1991, 192-222.

Newell, A. and Simon, H., **Human Problem Solving**, Prentice-Hall, Englewood Cliffs, NJ, 1972.

Norris, K., "The Accuracy of Project Cost and Duration Estimates in Industrial R&D", **R&D Management**, (2:1), 1971, 25-36.

Olson, M.H., "New Information Technology and Organizational Culture", **MIS Quarterly**, (6:1) 1982, 71-92.

Partridge, D., "The Scope and Limitations of First Generation Expert Systems", **Future Generation Computer Systems** , (3:1), February 1987, 1-10.

Pedersen, K., " Connecting Expert Systems and Conventional Environments", **AI Expert** (3:5) May 1988, 26-35.

Pelz, D.C. and Andrews, F.M., **Scientists in Organizations**, Wiley and Sons, New York, NY, 1966.

Perrow, C.A., " Framework for the comparative analysis of organizations", **American Sociological Review**, (32), 1967, 194-208.

Prahalad, C. K. and G. Hamel, "The Core Competence of the Corporation," **Harvard Business Review**, Vol. 68, No. 3, May-June, 1990, 79-91.

Prerau, D., "Selection of an Appropriate Domain for an Expert System", **AI Magazine**, (6:2) Summer 1985, 26-30.

Quinn, J.B., T. L. Doorley and P. C. Paquette, "Technology in Services: Rethinking Strategic Focus," **Sloan Management Review**, Vol. 32, No. 2, Winter 1990, 79-87.

Roberts, E. and Fusfeld, A., "Critical Functions: Needed Roles in the Innovation Process", in Katz, R. (ed), **Career Issues in Human Resource Management**, Prentice-Hall, Englewood Cliffs, NJ, 1982, 182-207.

Roberts, E. and Berry, C., "Entering New Businesses: Selecting Strategies for Success", **Sloan Management Review**, Spring 1985, 9-16.

Rogers, E.M. and Shoemaker, F., **Diffusion of Innovations**, The Free Press, New York, NY, 1962.

Ruth, S. and Carlson, C.. "Shell-Based Expert Systems in Business: A Return on Investment Perspective", **Proceedings of the Ninth International Conference on Information Systems**, November 30, 1988, Minneapolis, Minnesota.

Shannon, C., "A Mathematical Theory of Communication", **Bell Systems Technical Journal** (27:3) July 1948, 379-423.

Sviokla, J., "An Examination of the Impact of Expert Systems on the Firm: The Case of XCON", **MIS Quarterly**, (14:2), June 1990, 127-140.

Tanenbaum, A., **Computer Networks**, Prentice-Hall, Englewood Cliffs N.J., 1981

Vicinanza, S.S., Mukhopadhyay, T.,Prietula, M.J., "Software-Effort Estimation: An Exploratory Study of Expert Performance", **Information Systems Research**, (2:4) 1991, 243-262.

Whitey, M.; Daft, R.L., and Cooper W.H., "Measures of Perrow's Work Unit Technology: An Empirical Assessment and a New Scale", **Academy of Management Journal** (26) 1983, 45-63.

Zmud, R.W., "Management of Large Software Development Efforts", **MIS Quarterly**, June 1980, 45-55.

Figure 1
The Classification Framework
Four Generic Types of Software Applications

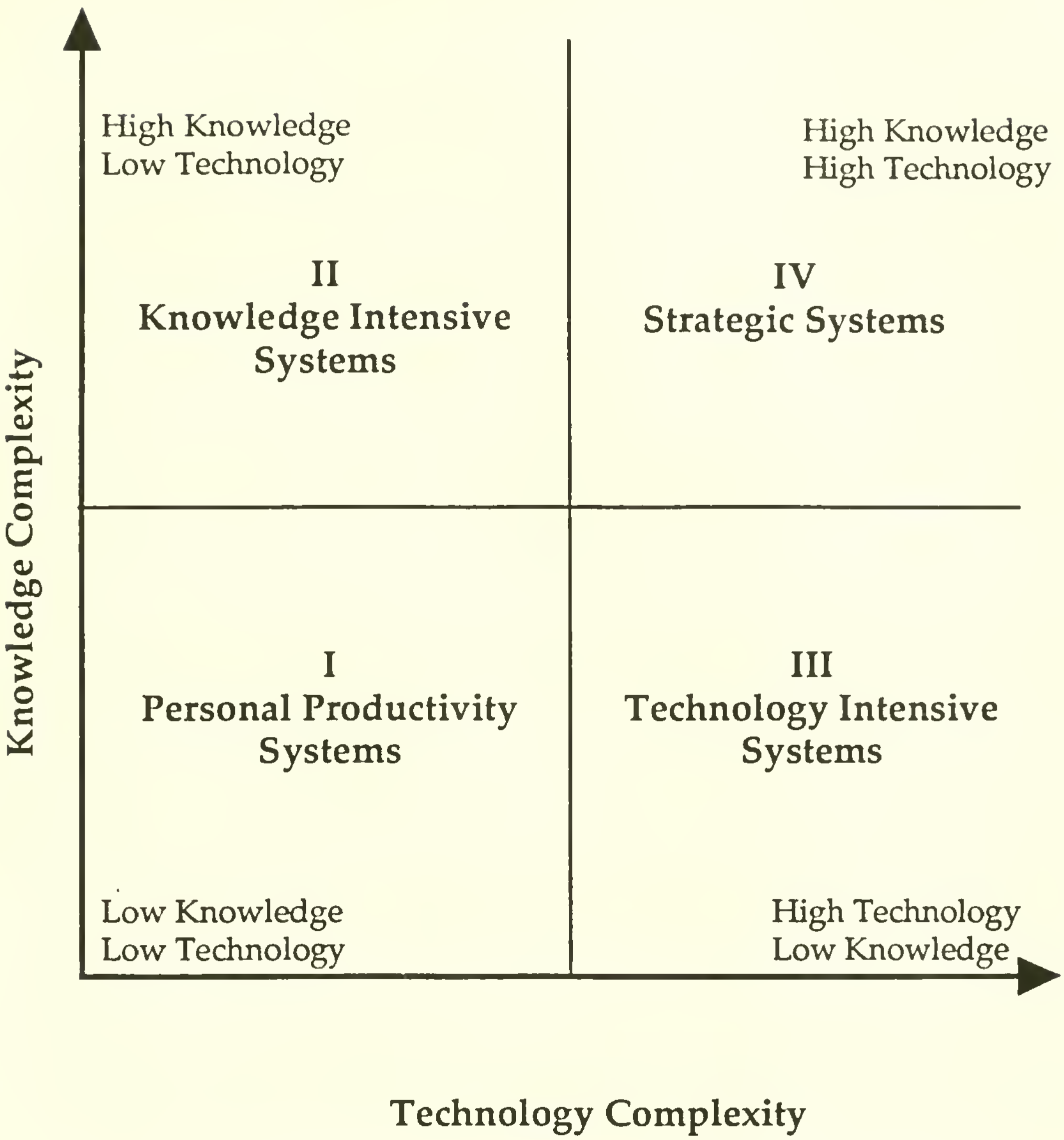
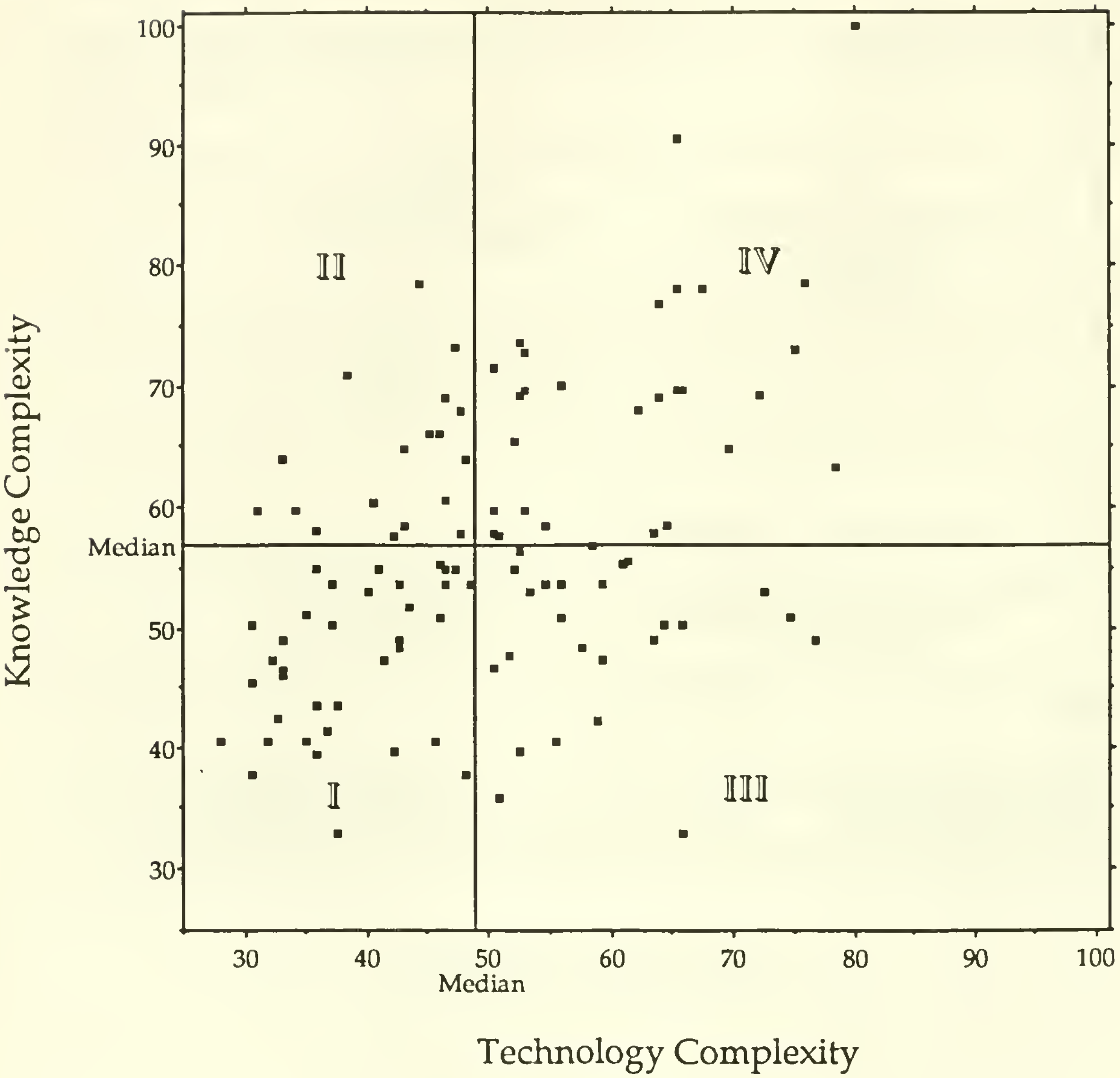


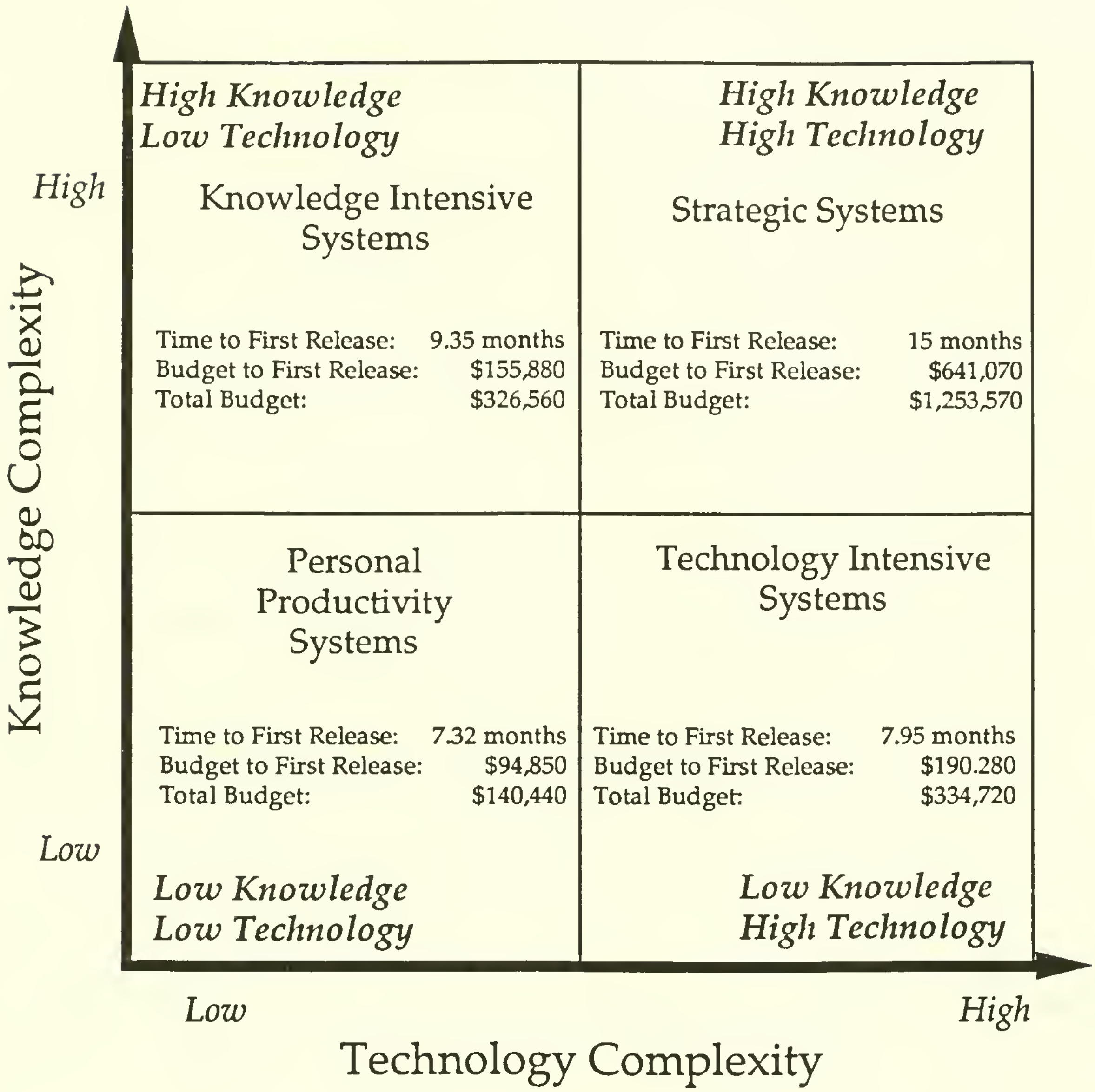
Figure 2

108 Software Applications
Classified by Knowledge and Technology Complexity



Quadrant 1	N = 35
Quadrant 2	N = 19
Quadrant 3	N = 21
Quadrant 4	N = 33

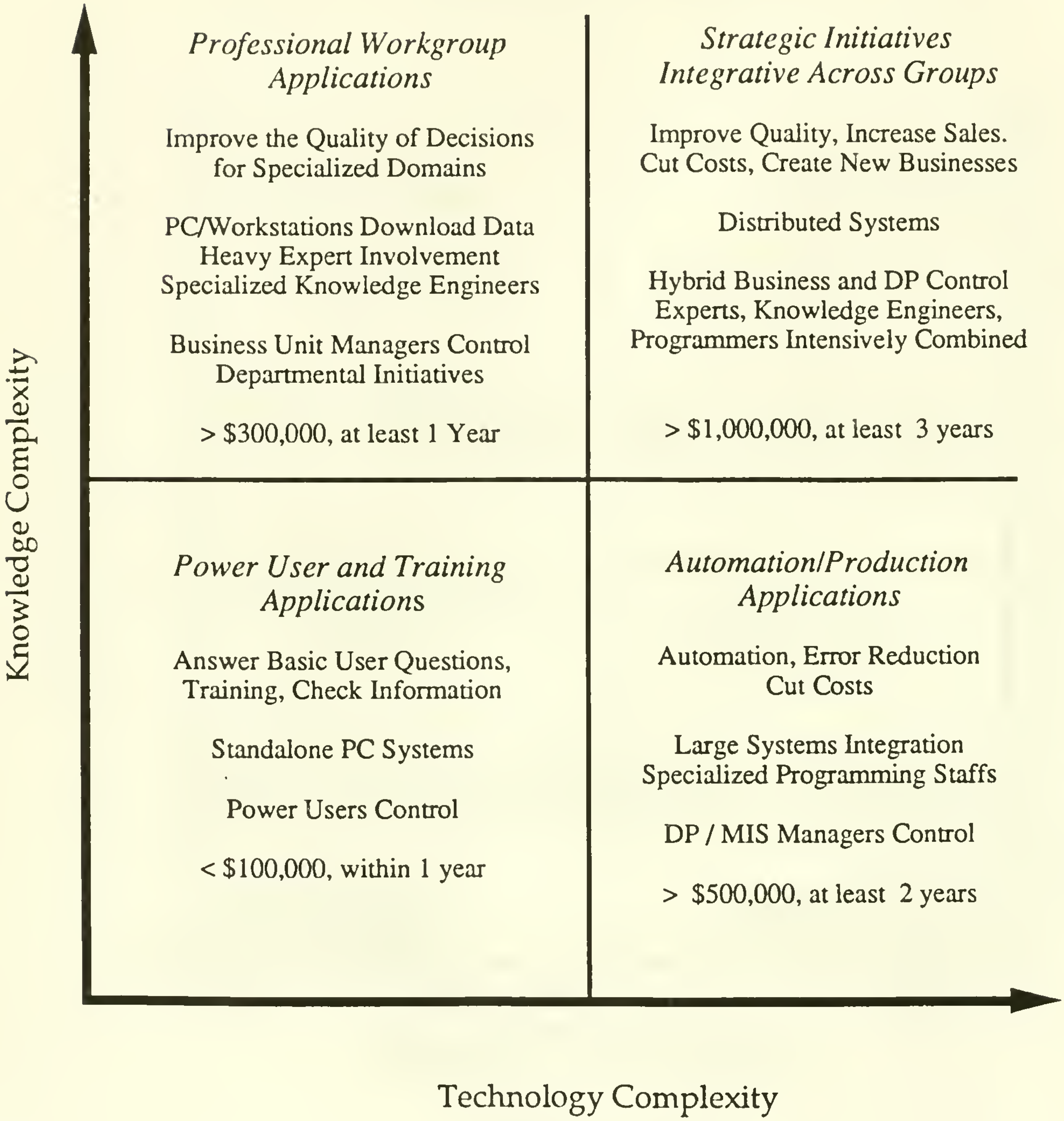
Figure 3
Project Duration and Budget Means by
Classification Quadrant



Sample Means

Time to First Release:	10.07 months
Budget to First Release:	$280,930
Total Budget:	$532,500

Figure 4
Contingent Approaches for Planning and Implementation



Appendix 1
Systems Descriptions

System Name	Area	Industry	Brief Description
ADVISE	Japan & Asia	Manufacturing	Sets appropriate and high-level cutting conditions that limit variations among production technicians in the cutting machining field.
AME5000	US & Canada	Manufacturing	This system aids operators in increasing mean time to failure (MTF) and decreasing mean to repair by transferring engineering and maintenance knowledge for disk drive production.
ANALOG	Japan & Asia	Engineering	Circuit Design. The specifications of an analog amp (DC amp) are inputs, and circuitry is designed with circuit diagrams and expected characteristics as outputs.
ANALYTIC	US & Canada	Manufacturing	Scans database of manufacturing test results to detect adverse trends early, with reduced effort by engineers or analysis. It is used during Manufacturing to monitor quality of large disk drives.
ARCHES	US & Canada	Health Services	Helps the Red Cross in updating "instructorships" and to offer courses.
ASTRA	Europe	Government	Used by Government Magistrate to determine whether technical installations are subject to Government Authorization for pollution and emissions control.
AUTOQ	Europe	Transportation	A batch system that reads a succession of flight inventories and for certain simple cases, makes recommendations for amending the inventory control parameters which are automatically incorporated into the main database. It helps keep track of when planes are required to be inspected.
BATCH	US & Canada	Manufacturing	Advises chemical plant operators on additive quantities for non-reversible processes for plastics/polymers production.
BIGBOX	Japan & Asia	Financial Services	Scheduling and problem fixing for mainframe computers
BUILDSAFE	Japan & Asia	Financial Services	Built by a P&C insurer for disaster prevention when structures are newly built. This system uses data such as construction conditions, structure details, and purpose of use in order to propose the most appropriate equipment.
CADDIS	Europe	Transportation	Failure diagnostics in avionics black boxes, used in automated testing. The system provides intelligent support to the avionics test technician when carrying diagnostic checks by helping with problem analysis and understanding.
CADPOS	Japan & Asia	Energy	Operator support through on-line real-time performance of failure analysis for the instrumentation of powered devices, and diagnosis of abnormalities in operation, with display of suspected causes and suggested responses.
CATALOG	US & Canada	Engineering	Allow users to browse and select PC computing equipment hardware items to form a PC configuration based on different company division rules, regulations, and standards. The system has a user base of 5500 users spread throughout the company and located in various sites across the US.
CEMENT	Japan & Asia	Construction	Cement kiln operator support system in normal conditions.
CHECKER	Europe	Transportation	Tracks an inventory database of seats and flights and helps make changes to basic seating parameters.
CHEMTRAIN	Europe	Engineering	An advisory system for a chemical production installation. Training tool for new employees and maintenance support.

COAL	US & Canada	Engineering	Assists coal buyers in making coal purchasing decisions for their power plants. Assesses impact of 100 characteristics of coal, such as burning, hard to pulverize, environmental impact, emissions etc., costs, temperatures operations, handling of the coal.
COALMIX	Japan & Asia	Manufacturing	Expert system for planning raw coal mixing. The aim of this system is to propose plans for blending many types of raw coals of varying degrees of quality in order to make coke in a coke furnace at lower cost and while maintaining uniform quality.
COMBUST	Japan & Asia	Engineering	Intelligent combustion control system for trash incinerators that diagnoses sensor errors and performs combustion control operation based on past operating history and overall combustion regime decisions.
CURTAINS	Europe	Transportation	To save time in a busy department by "expertly" dealing with the more mundane parts of yield management on airplane flights. The day before departure of a flight-it performs automatically some of the functions previously known as the "final day check".
DEATH	US & Canada	Financial Services	Claims notification processing system for life insurance. s When the company is notified of a death this system obtains quote values and performs an analysis of what forms or information will be required to complete the claim.
DETERGENT	US & Canada	Manufacturing	Color advisor for linear detergent alkylide to provide improved product quality in the linear detergent alkylide operation.
DISABILITY	Japan & Asia	Financial Services	Simulates calculation of financial loss to persons suffering after-effects of physical impairment due to traffic accidents.
DUMP	Europe	Engineering	This system works through the core dumps of IBM Mainframes (system crashes) and detects what went wrong.
EINSTEIN	Japan & Asia	Financial Services	Supports leasing decision-making for Asian firms moving in the US, analyzing the client's position and suggesting optimum financing.
ELECTRON	Japan & Asia	Manufacturing	A qualitative observation analysis tool to complement electron probe micro-analysis in testing materials.
ELUS	US & Canada	Financial Services	Evaluates life insurance applications entered through a PC based front end entry system. The system will approve a percentage of the applications and refer the balance to an underwriter with expert system guidance provided.
EMIT	US & Canada	Manufacturing	Provide assistance instrument operators in chemical packing operations.
ENDOSER	US & Canada	Financial Services	Automatically processes endorsements (changes in P&C insurance policies) as without human intervention, and refers the remainder to underwriters with a message explaining the reason for the referral.
EQUIPMAIN	Japan & Asia	Manufacturing	Identifies the causes in the event of trouble and provide guidance on response methods for a manufacturing plant.
ERRORS	Japan & Asia	Manufacturing	A programming tool to diagnose the location of errors and relationships between them in code.
EXKREA	Europe	Financial Services	Supports the acquisition of credit information by loan officers to make sure they have the necessary information and that is "good" information upon which to make decisions and present credit evaluations.
EXMONT	Europe	Manufacturing	Helps to fix the painting machines in a car plant.
EXPERFORM	Japan & Asia	Manufacturing	Uses system operation measurement data to analyze problems in performance on mainframe computers.
EXTEL	Europe	Sales	Used to help customers in choosing the best phone system for their needs. It takes the requests of customers and configures the correct phone system and calculates monthly fees.
FARM	Japan & Asia	Manufacturing	Breakdown diagnosis for agricultural machinery.

FAST	US & Canada	Sales	Assists sales account teams to determine the computer configuration required to meet the customer's computer performance requirements
FERRITE	Japan & Asia	Manufacturing	Ferrite material design system.
FIELD AUDIT	Japan & Asia	Financial Services	Assists inspection tasks in a securities company's branch offices. Creates all the necessary task lists by account.
FITNESS	Japan & Asia	Health Services	This is a system for prescribing exercise regimens (mainly walking) for office workers.
FORKLIFT	Japan & Asia	Manufacturing	Electric forklift breakdown diagnosis system. Signal voltages from the various areas of an electric forklift are read in via a special piece of equipment, and the system uses these voltage levels to determine the status level of the vehicle.
FTS	Europe	Engineering	A diagnostic system for the failures in a driverless transportation system.
FXTRADER	US & Canada	Financial Services	Audits foreign exchange trades by sifting through completed trades in a batch mode, and then presenting a graphical user interface for hypothesis testing and investigation.
GASINOIL	Japan & Asia	Energy	Gas-in-oil analysis diagnosis expert system used by a utilities company for equipment repair. When an abnormality occurs within a piece of equipment, peripheral insulating oil and insulation material breaks down from heat due to factors such as electrical discharge and overheating. Gases are produced which are characteristic of the nature of the abnormality and type of insulation, and these gases are dissolved in the insulating oil.
HOMEHEAL	US & Canada	Financial Services	Evaluates home health care claims for the government without using a registered nurse to do the evaluation of the claim.
INSTALL	Europe	Engineering	Aids in the management and monitoring of a technically sophisticated plant installation.
INSULATOR	Japan & Asia	Manufacturing	A production control system focusing on load packing and process scheduling.
INTERBANK	Japan & Asia	Financial Services	Interpretation of telexes giving instructions on inter-bank yen payments (automatic slip preparation).
INVENTORY	US & Canada	Manufacturing	Tracks shipments between locations in dozens and dollars for product shipments in a large consumer products/clothes company.
JIT	Japan & Asia	Manufacturing	Reduces stock and improves productivity by systematizing just-in-time production by giving instructions for an optimized line setup sequence, taking into consideration ease of retooling, delivery date, and task leveling.
KOMEX	Europe	Engineering	Diagnosis and repair of copy machines
KONGAC	Europe	Manufacturing	A machine configuration system for turning machines: flywheels, turntables, rotors.
LDEX	Europe	Manufacturing	Diagnosis system for process and machine problems in the assembly and manufacture of hard disks.
LEONARDO	Europe	Sales	Configuration and production of bids involving computer equipment in a large services company.
LTD-UA	US & Canada	Financial Services	Examining renewals of long term disability insurance for groups of 200-1000 persons.
LUS	US & Canada	Financial Services	Underwrites life insurance applications. This is a comprehensive system, going from the front-end batch screening to the back-end decision-support.
MACHCAD	Japan & Asia	Manufacturing	Machine part drawing data created by CAD system is used as input to automatically determine the sequence and tools used to machine the parts, machining conditions, and toll movement paths. It outputs a program for controlling the equipment used for machining.
MAINSCHED	Japan & Asia	Financial Services	Prepares mainframe schedules.
MEDAFFAIR	Japan & Asia	Financial Services	Takes the place of a physician for some aspects of medical risk selection for life insurance.
MEDBENEFIT	Japan & Asia	Financial Services	Determines benefit payments in medical claims within a large heath insurer.

MEDCLAIMS	Japan & Asia	Financial Services	Another benefit payments system for medical claims in a large heath insurer.
MEGAFILEX	Europe	Manufacturing	Diagnosis system in the quality assurance area related to disk manufacture and repair
MOLDCOST	US & Canada	Engineering	Estimates the cost of purchasing a mold for plastic injection molded parts. Allows the users to estimate a cost in 5 minutes, usually within 85-90% of actual cost, with little moldmaking knowledge required.
MOMOTARO	Japan & Asia	Engineering	High pressure gas governmental regulation support system.
NATURAL	Japan & Asia	Manufacturing	Extracts key works from Japanese language texts for subsequent natural language processing.
OHCS	Japan & Asia	Manufacturing	A chemical plant production management system.
OPTICAL	Europe	Manufacturing	Used to plan the manufacture of optical parts, including laminates and coatings..
PAMPA	Europe	Transportation	Assists in planning regular aircraft maintenance.
PENSION	Japan & Asia	Financial Services	Supports pension consultations, which have been increasing in complexity in recent years. Data such as work history, number of years on the job, and amount of pay is input for various pension proposals.
PERSONPLAN	Japan & Asia	Financial Services	General personal financial planning system.
PILETYPE	Japan & Asia	Construction	When designing a building, the appropriate foundation pile type is selected according to such conditions as the site, ground, and construction. Outline designs and rough cost estimates are made.
PIPING	Japan & Asia	Energy	Assists in designing vibration-resistant piping in nuclear power plants.
PREDICT	Europe	Engineering	Helps engineers select, use, and combine different calculation methods for interpreting experiments on translucent materials.
PROCESS	Japan & Asia	Energy	Process monitoring diagnosis system.
PRODPLAN	Japan & Asia	Manufacturing	Assembled product production planning.
PRODSCHED	US & Canada	Manufacturing	Defines production schedules for a multiple product producing chemical plant.
QGEN	Europe	Engineering	Formulates ordinary queries into SQL format for use with DB2 database. Used in the Marketing department to help with queries.
RAMSES	Europe	Financial Services	Used in the securities and bonds department for customer oriented consultation.
REALTIME	Japan & Asia	Engineering	General purpose front-end tool for knowledge base development in an engineering company.
RESIN	Japan & Asia	Engineering	Creates production plans for synthetic resin manufacturing.
REVA	US & Canada	Engineering	Detects potential problems and helps solve current problems for various types of rotating drum plant equipment by using complex vibration theory.
RUBBER	Japan & Asia	Manufacturing	Production planning in a large rubber manufacturer.
SALESPLAN	Japan & Asia	Manufacturing	Sales planning system that works with market forecasts to integrate them into production planning.
SEES	Japan & Asia	Manufacturing	Adhesive composition design support system.
SELLPROD	Japan & Asia	Financial Services	Supports selling of derivative financial products in a large Asian bank.
SERVCREW	Japan & Asia	Transportation	Allocate the service crew on a monthly basis. support system.
SHADE	US & Canada	Engineering	Helps human debuggers find logic problems in software more quickly and efficiently.
SHIELD	Japan & Asia	Construction	Supports the formulation of execution plans for shield construction in building roads.
SOLAR	Europe	Transportation	Assists aircraft controllers to reallocate aircraft to flights when flights are disrupted due to weather, technical faults, or air traffic control.
SOLEIL	Europe	Manufacturing	Optimizes the alloying process in the production of solar cells.
SOS	Japan & Asia	Engineering	Standardize new employee HRM processes.

SPA	Japan & Asia	Manufacturing	Supports analysis and improvement of software programs by applying language construction and bug analysis expertise.
STEELWELD	Japan & Asia	Manufacturing	Steel frame welding advisory system in a large steel manufacturer.
STICKY	Japan & Asia	Manufacturing	Supports improving efficiency of assembly line in a large plant.
SWEET	Japan & Asia	Sales	International telephone service assistance support system. Response to queries regarding usage of international communications (international direct dialing, international telephone fax).
SYCTEX	Japan & Asia	Construction	Produces work order lists from designers drawings in building construction.
TAX	Europe	Transportation	Monitors the movement of residents stationed abroad for more than a year to and from the home country, and identifies the periods for which no national income tax is incurred.
TES	US & Canada	Financial Services	Explains the taxation of individual life insurance to the customers of a large insurance company.
TOS	US & Canada	Manufacturing	Gather orders for teleprocessing services and distributes them to service providers.
TRANSFORM	Japan & Asia	Energy	Transformer and pumping equipment operation support system. This is an on-line system providing task instructions for power outages and internal breakdowns for transformer equipment.
TRASH	Japan & Asia	Manufacturing	Trash incineration plant operation support system. The goals were to simplify and stabilize operation of a trash incinerator plant, which previously had relied on the experience and intuition of veteran operators.
TVADVERT	Japan & Asia	Sales	Helps create proposals for television commercials. Decides which TV. stations to use and the budget for each station. .
UCONF	Europe	Engineering	Configures an industrial machine product line which is made up of various standardized manufactured elements.
VITAMINS	Japan & Asia	Manufacturing	Bulk vitamin production scheduling system.
VM	US & Canada	Manufacturing	Generates manufacturing build information from customer orders for a large computer manufacturer.
WAREHOUSE	Japan & Asia	Manufacturing	Simple automatic warehouse system reference support system, incorporating stackers, cranes, racks, and loading stations.
WORKERS	US & Canada	Financial Services	A worker compensation insurance risk management system used by underwriters on all prospective accounts up to $25,000 annual premium which have been submitted by independent insurance agents.
XCONF	US & Canada	Manufacturing	Configuration system for manufacturing customized mid-sized computer systems.

Appendix II
Complexity Classifications

System Name	Knowledge Complexity	Technology Complexity	Framework Quadrant
ADVISE	45.48	30.42	I
AME5000	71.67	50.42	IV
ANALOG	57.62	50.83	IV
ANALYTIC	40.71	55.42	III
ARCHES	43.57	37.50	I
ASTRA	53.81	48.33	I
AUTOQ	33.10	65.83	III
BATCH	47.38	41.25	I
BIGBOX	55.48	45.83	II
BUILDSAFE	51.19	35.00	I
CADDIS	64.05	47.92	II
CADPOS	39.52	35.83	I
CATALOG	50.24	65.83	III
CEMENT	49.05	63.33	III
CHECKER	47.38	59.17	III
CHEMTRAIN	60.48	40.42	II
COAL	70.95	38.33	II
COALMIX	53.81	55.83	III
COMBUST	46.67	50.42	III
CURTAINS	33.10	37.50	I
DEATH	50.24	64.17	III
DETERGENT	50.24	30.42	I
DISABILITY	48.33	57.50	III
DUMP	60.71	46.25	II
EINSTEIN	37.86	30.42	I
ELECTRON	37.86	47.92	I
ELUS	73.10	75.00	IV
EMIT	55.00	40.83	I
ENDOSER	53.10	53.33	III
EQUIPMAIN	58.57	64.58	IV
ERRORS	40.71	27.92	I
EXKREA	42.62	32.50	I
EXMONT	41.43	36.67	I
EXPERFORM	58.57	42.92	II
EXTEL	50.95	74.58	III
FARM	50.95	45.83	I
FAST	58.10	35.83	II
FERRITE	57.86	47.50	II
FIELD AUDIT	76.90	63.75	IV
FITNESS	69.05	46.25	II
FORKLIFT	35.95	50.83	III
FTS	47.86	51.67	III

FXTRADER	51.90	43.33	I
GASINOIL	46.43	32.92	I
HOMEHEAL	90.71	65.42	IV
INSTALL	68.10	62.08	IV
INSULATOR	57.86	63.33	IV
INTERBANK	39.76	42.08	I
INVENTORY	39.76	52.50	III
JIT	55.00	46.25	I
KOMEX	46.19	32.92	I
KONGAC	53.81	46.25	I
LDEX	40.71	45.42	I
LEONARDO	53.81	42.50	I
LTD-UA	69.29	72.08	IV
LUS	100.00	80.00	IV
MACHCAD	53.10	40.00	I
MAINSCHED	50.24	37.08	I
MEDAFFAIR	56.90	58.33	IV
MEDBENEFIT	57.86	50.42	IV
MEDCLAIMS	47.38	32.08	I
MEGAFILEX	68.10	47.50	II
MOLDCOST	57.62	42.08	II
MOMOTARO	70.24	55.83	IV
NATURAL	69.29	52.50	IV
OHCS	55.00	52.08	III
OPTICAL	72.86	52.92	IV
PAMPA	55.48	60.83	IV
PENSION	73.33	47.08	II
PERSONPLAN	56.43	52.50	IV
PILETYPE	69.76	65.83	IV
PIPING	59.76	34.17	II
PREDICT	66.19	45.00	II
PROCESSCHECK	53.81	54.58	III
PRODPLAN	59.76	30.83	II
PRODSCHED	55.00	47.08	I
QGEN	59.76	50.42	IV
RAMSES	55.71	61.25	IV
REALTIME	69.76	65.42	IV
RESIN	53.81	37.08	I
REVA	66.19	45.83	II
RUBBER	69.76	52.92	IV
SALESPLAN	42.38	58.75	III
SEES	59.76	52.92	IV
SELLPROD	69.05	63.75	IV
SERVCREW	53.10	72.50	III
SHADE	57.86	63.33	IV
SHIELD	55.00	35.83	I
SOLAR	65.00	69.58	IV
SOLEIL	73.81	52.50	IV
SOS	49.05	42.50	I

SPA	58.57	54.58	IV
STEELWELD	78.57	44.17	II
STICKY	40.71	35.00	I
SWEET	49.05	32.92	I
SYCTEX	65.48	52.08	IV
TAX	43.57	35.83	I
TES	53.81	59.17	III
TOS	49.05	76.67	III
TRANSFORM	78.10	67.50	IV
TRASH	78.10	65.42	IV
TVADVERT	50.95	55.83	III
UCONF	64.05	32.92	II
VITAMINS	40.71	31.67	I
VM	65.00	42.92	II
WAREHOUSE	48.33	42.50	I
WORKERS	63.33	78.33	IV
XCONF	78.57	75.83	IV