Communications of the Association for Information Systems

Volume 1, Article 17
June 1999

DEVELOPING AND TEACHING IS'97.2: PERSONAL PRODUCTIVITY WITH INFORMATION TECHNOLOGY

Gordon B. Davis
J. David Naumann
Gove Allen
Carlson School of Management
University of Minnesota
gdavis@csom.umn.edu

TUTORIAL

DEVELOPING AND TEACHING IS'97.2: PERSONAL PRODUCTIVITY WITH INFORMATION TECHNOLOGY

Gordon B. Davis
J. David Naumann
Gove Allen
Carlson School of Management
University of Minnesota
gdavis@csom.umn.edu

ABSTRACT

It is common for university students either to have introductory skills in the basic desktop software packages or to obtain these skills by self-study modules or short courses. What students lack is an understanding of how to use desktop packages effectively to improve their productivity. This is the missing link in the curriculum. This tutorial describes one solution: the IS'97.2 course on Personal Productivity with Information Technology.

An appendix presents a sample lesson used with this course.

Keywords: IS curriculum, productivity, desktop computing, knowledge work.

I. INTRODUCTION

The desktop personal computer is fundamental to knowledge work. Over the years, the software packages have grown in size and scope. For example:

- Microsoft Word version 2 had 311 commands; Word 97 has 1,033
- WordPerfect version 4.2 had 1 megabyte of programs and supporting materials. Corel WordPerfect 8 includes 100 million

bytes

- PowerPoint 97 has seven pages of user options
- Excel 97 includes 23 different customizable toolbars

When faced with large numbers of features and options in software packages, users tend to learn a minimum set of commands and never explore further. Again and again, we find users with significant experience using a package have not explored options that would be useful to them. Users report to us that they run only one program at a time. They are confused by hierarchical file organizations. Students enter college more proficient than ever before. They can surf the web and use e-mail. Most are comfortable enough with word processing software to generate a professional-looking resume. By the time they are admitted to the business school (or not long after), students can solve simple accounting problems using spreadsheet software and can prepare colorful slide presentations. These skills are only the foundation of the set that is required to achieve the potential productivity improvements that information technology offers knowledge workers. If a student who has been introduced to basic features and functions wants instruction to help improve productivity in use of desktop technology, schools usually do not provide a course. Fortunately, such a course exists. It is defined in the IS curriculum models, and we have had three years experience in teaching it.

The IS'97 model undergraduate curriculum developed cooperatively by

- ACM (Association for Computing Machinery),
- AIS (Association for Information Systems), and
- AITP (Association for Information Technology Professionals)

contains a new course not previously taught, IS'97.2*, Personal Productivity with Information Technology*. The course, defined briefly in the model curriculum, is described in Appendix A. This tutorial describes the reasoning behind the course and how it is taught at the University of Minnesota. It includes a suggested syllabus, course topics, exercises, projects, and teaching methods. The discussion centers on pitfalls and problems the authors experienced in designing and delivering of the course, including the source and use of teaching materials.

II. OBJECTIVES

The objective of IS’97.2 is to improve an undergraduate student’s knowledge work productivity through the effective use of information technology. Understanding how to apply information technology to achieve improved productivity in knowledge work is an important element in the general computer knowledge and skills of all students. It is essential for information system specialists.

Prerequisite to the IS’97 productivity course is elementary skill in using a practical “toolkit” of software packages vital to knowledge work. Caution: do not use the productivity course as a remedial course for the basic toolkit technology; to do so dilutes its purpose. The toolkit includes software for word processing, spreadsheets, presentation graphics, database management, Internet access, and e-mail. The toolkit may be learned in different ways:

- through software skill development course modules,
- in connection with a course making use of the software, or
- by individual self-directed study.

The scope of this instruction generally is limited to basic features and functions.

The IS’97 model curriculum specifies a survey course in concepts, terminology, and practice for information technology and information systems in organizations. This survey course is separate from and different in scope and purpose than the productivity course. It may be taken before or after IS’97.2.

Although these two courses (or course modules) provide basic knowledge and skill, they are insufficient academic preparation for productive use of information technology. The IS’97.2 Personal Productivity with Information

Technology course addresses issues and methods not found in either the prerequisite toolkit or the information systems survey course. It focuses on how to be more productive in using information resources in individual and collaborative knowledge work. Knowledge work is defined as:

> "*human mental work performed to generate useful information. In doing it, knowledge workers access data, use knowledge, employ mental models, and apply significant concentration and attention."* [Davis and Naumann, 1997, p. 6]

When knowledge workers perform individual or collaborative work, their productivity depends on what they know, their ability to obtain information from a variety of sources, and their ability to design and carry out analytical and information processing activities. Information technology improves productivity in knowledge work.

Organizations are increasingly aware of the importance of knowledge work productivity. Drucker was prophetic in his 1969 statement:

> "*To make knowledge work more productive will be the great management task of this century, just as to make manual work productive was the great management task of the last century.*" [Drucker, 1969, p. 290]

COURSE OBJECTIVE

The over-all objective for the course is to help students apply information technology to improve productivity. This goal is achieved by three broad areas of instruction:

1. Concepts and principles for productivity in knowledge work. Emphasis is on relating concepts and principles to information technology features and functions.
2. Analyzing requirements for information technology in knowledge work and evaluating the productivity effects of alternatives.
3. Developing skill in software functions and features and development alternatives that provide knowledge work productivity improvement.

These three areas may be modified to suit the knowledge work focus of students, especially in the problems and exercises employed, and functions and features studied. For example, accounting students might take more time with error-control while marketing students might learn more about analytical functions.

COURSE OUTCOMES

A student who completes the course successfully will have learned to apply:

- new ways to think about individual and group productivity in knowledge work,
- new approaches to understanding the value of information technology in knowledge work,
- new ways to manage the time investment to find and learn new features in software packages and to achieve payback from the investment, and
- selected high productivity payoff software package functions and features.

COURSE ORIENTATION

The course is oriented to packaged software. It assumes that individuals should always look first for existing solutions such as packages or specialized applications. If packages or existing applications do not fit, then tailoring or customizing is recommended. When existing software cannot be tailored to fit a problem, it may often be extended or replaced with a programmed solution. The course introduces methods that range from developing simple enhancements of existing packaged software to complete end user applications. End user application development is discussed but treated as a less desirable alternative. This content emphasizes the difference in scope and quality assurance for end user systems developed for a single user or a small homogeneous group and high quality, robust applications that can be deployed to multiple unknown users. The course includes elementary interface development and procedural programming in a high-level object-based language.

The course provides the future specialist with insight into individual systems and how they can be built and supported. Also, the course provides a simplified introduction to principles and procedures for analysis and design. A "system development in the small" approach is used. This subset of knowledge and skill supports and enhances learning in subsequent "standard" application development and programming courses.

III. ASSUMPTIONS ABOUT STUDENT COMPUTER AND SOFTWARE KNOWLEDGE

We assume that students already know how to use the basic functions and features of software packages for word processing, spreadsheets, presentation graphics, e-mail, and the Internet. Some students can also use the basic capabilities of a database package. The course builds upon this elementary knowledge and experience. Of course, students will have a wide range of software package knowledge and skill. Even students with significant experience do not know the concepts and principles of productivity to guide them in selecting and using productivity enhancing functions and features. However, students with prior training and experience with programming, systems development, and the software package toolkit do not fit the structure and pace of IS’97.2 and should be provided an alternative path.

IV. APPROACHES TO IMPROVED PRODUCTIVITY

Information technology can be applied to three approaches to improved knowledge work productivity:

1. Increasing knowledge work efficiency. Many software functions and features reduce the time and/or energy (attention and concentration) needed.
2. Increasing effectiveness. Software functions and features support knowledge workers in extending the scope, depth, and quality of the work they perform. Many activities are possible with information technology that are not possible without it, (e.g., communication and coordination software, search tools, analytical functions, error detection/correction functions, and presentation features)
3. Expansion and/or conservation of the limited time and attention that knowledge workers can devote to activities. Communications, coordination, scheduling of work, design of work activities, and managing motivational factors can conserve or expand the time and attention that can be applied.

All three approaches can employ economic analysis, although such analysis is most applicable to efficiency effects. The cost of productivity improvements is learning time plus lower productivity during the first few uses (when it may take more time to apply new functionality). The payback is in subsequent use, through reduced time and energy and through increased functionality and/or reduced errors.

V. USE OF INFORMATION TECHNOLOGY IN PRODUCTIVITY IMPROVEMENT

The course addresses the following principles of productivity improvement:

• Reuse of structure and procedures	• Efficiency in data access	• Minimizing errors
• Reuse of data	• E-mail productivity	• Minimizing learning costs
• Efficiency in process activities	• Productivity in collaborative work	• Knowledge work management

REUSE OF STRUCTURE AND PROCEDURES

A major source of productivity is reuse. Most knowledge work activities and outputs require planning and design as well as execution. The planning of the work and design of the outputs can often reuse existing patterns and models. Templates, style sheets, designs, recorded macros, macros, stored queries, stored reports, custom lists (spreadsheets), and custom expansion of abbreviations (word processing) are examples of functions that support reuse of structure and procedures.

REUSE OF DATA

Data used in more than one application should not have to be re-entered. Examples of functions that support reuse of data are databases, import and export functions, various copying and pasting options, and linking and embedding.

EFFICIENCY IN PROCESS ACTIVITIES

Functions and features that support improved efficiency in performing common activities are outlining and table of contents (word processing), search and replace, built-in analytical functions, graphing and drawing functions, macros, and task management functions. Software packages allow options and preferences. These can be set to match task activities and improve efficiency and reduce errors. Appropriate use of menus, icons, and keystrokes reduces access time to functions.

EFFICIENCY IN DATA ACCESS

The time and effort to store and retrieve data can be improved by directory/file structures, search functions, search planning and stopping rules, and good database design. Conversion functions among packages and different file formats reduce time to transfer data.

E-MAIL PRODUCTIVITY

This task is identified separately because of its widespread use. Productivity can be improved by good folder structure and naming, use of archiving procedures, filtering mechanisms, directory structures and lists, and automating functions such as reply and confirmation of receipt.

PRODUCTIVITY IN COLLABORATIVE WORK

Communication and collaboration costs rise rapidly as team size increases. Use of collaborative work functions can reduce the overhead associated with teams. Examples are team communications support, scheduling and coordination functions, shared data access mechanisms, concurrent work software, collaborative authoring software, and meeting software.

MINIMIZING ERRORS

Errors significantly reduce productivity. Information technology can be used to reduce errors in knowledge work processes and data access. Examples are consistent and meaningful naming of files and folders, copying and pasting, importing and exporting, linking and embedding, automatic entry of phrases and names based on custom expansion in word processing and custom lists in spreadsheets, and use of input forms (with validation methods) for entering data. There are error-detecting functions in databases (such as the examine function and spreadsheet functions) and practices to reduce and detect errors (such as the spreadsheet audit function, the use of range names instead of cell addressing, and freezing panes so column or row headings are visible in large spreadsheets).

MINIMIZING LEARNING COSTS

A new software package, change, or upgrade has an associated learning time. If both software features and task process are changed, there is a loss of productivity in both software use and task activities. When software features are learned so that they are mainly automatic, attention can be directed primarily to the task. Process learning time can be reduced by compatible software versions, by online tutorials, and by online HELP facilities.

KNOWLEDGE WORK MANAGEMENT

Knowledge work is sensitive to motivational factors based on scheduling of work to provide motivation to complete, planning to reduce rework and lost opportunities, stopping rules to avoid unproductive expansion of work, search strategies, and feedback. A combination of organization, management, and information technology may be applied. Information overload can be reduced by appropriate data structures, search mechanisms, stopping rules, and data reduction functions.

VI. EXAMPLE SYLLABUS

Table 1 presents a flow of topics and exercises to achieve the objectives of IS'97.2. Each topic may take several class and/or lab sessions.

Table 1. Topics and Exercises for IS'97.2

Topic Number	Lecture/Discussion Topics	Problems on Analysis and Software Functions, Features, and User Practices
1	Understanding the nature of knowledge work. Optional: review of core information technology.	Task/activity analysis to analyze requirements for software packages.
2	How productivity improvement is achieved in knowledge work.	Reuse: templates for word processing, spreadsheets, and presentations. Style sheets for word processing and presentations.
3	Knowledge work management to improve productivity.	Software to schedule and track knowledge work. Software functions for switching among tasks.
4	Exploration/evaluation of value of technology in collaborative work.	Exploration of collaborative work software. Use of group work functions.
5	Analyze data and communication requirements. Tools and methods for communication productivity.	Using discovery approach to requirements analysis. Applying e-mail tools and methods.
6	Organization and management of individual files and directories.	Creating an efficient structure for directories and files. File search and indexing tools. Moving data between applications and sharing of data by applications: copying, importing, linking, embedding.
7	Access and report facilities of database package (for a database that already exists).	Forming and executing queries and reports. Analytical tools for data analysis. Importing and exporting to/from spreadsheets.
8	External data search using Internet. Productivity in search procedures and stopping rules.	Applying search methods and tools. Developing and applying stopping rules to constrain search time and effort.
9	Selecting options or preferences that improve productivity in use of packages. Reuse of sequences of instructions by use of recorded macros. Reducing time to access macros and other functions. Functions that reduce effort and reduce errors. Obtaining assistance from HELP facilities.	Tailoring and preference options in software packages. Recording macros for spreadsheets and word processing. Use of menus, icons, and keystrokes to reduce access time. Reducing effort and errors with autocorrect, fill, and similar functions Exploring efficient use of HELP.
10	Developing a procedure for reuse of a sequence of instructions where there are conditions and alternative	Extend a recorded macro by adding input box and message boxes Extend one or more recorded macros by

	procedures. Program control structures. Adding control structures to recorded macros.	adding decision logic looping and repetition until some condition is achieved.
11	Types of user interfaces. Design of user interfaces. Software development environments. Features of a typical user interface development tool.	Develop an interface for a simple end user application and program instructions for each object a user can select (using Visual Basic)
12	Evaluating or developing solutions with databases.	Develop flat file using database package, create input form, and input data. Develop and create database with two or more tables and enforce integrity.
13	Characteristics of robust, error resistant applications. Methods and functions for error control and error detection.	Evaluate robustness of an application. Design a spreadsheet that is very robust relative to errors. Use of validation rules in database input forms and database tables.
Optional	Technology infrastructure with emphasis on infrastructure for individual or small group.	Analyze, specify, and justify a personal information technology infrastructure. Formulating a productivity-based personal policy on upgrading of software.
Optional	Relationship of individual systems to organizational systems. Organizational support for individual and group productivity.	Evaluate cooperation with information technology function.
Project	Integration of topics and building of end user applications to improve productivity.	Use software packages to create application that demonstrates use of information technology to improve knowledge work productivity of an individual or small group.

VII. TEXTBOOK SUPPORT

The teaching approach combines discussion of concepts and methods with practical exercises. The concepts and methods provide long-lasting knowledge; the practical exercises give the student the ability to apply the knowledge. Details of current technology are skipped or de-emphasized, except as they are needed to clarify concepts and to support exercises.

The course requires three types of textbook support:

1. concepts and analysis,
2. software functions and features, and
3. exercises.

A book (Davis and Naumann, 1997) was prepared to provide concepts and analysis, an overview of relevant software functions, and exercises for the students.

In using the textbook, we found that most students need supplementary materials:

- A software manual describing how to use the functions and features emphasized in the course. An ideal solution would be a custom selection of items from one of the numerous how-to-do software manuals (especially those that cover some advanced features). An alternative is to identify HELP pages and examples in the documentation of the packages chosen.
- Walkthrough examples of use of software functions and features. These examples may be the basis for class demonstration, but students need to be able to observe and then repeat them on their own.
- The instructor must continually emphasize the connection between the productivity concepts and the functions and features being learned.

Problems and exercises are defined in Table 1. They range from very simple, low investment, and high payoff functions and features such as templates and simple recorded macros to more complex exercises such as adding logic to recorded macros and developing an end user application. Appendix B contains a comprehensive exercise that illustrates how to do a simple recorded macro for a spreadsheet and then extend the macro by adding logic. The exercise starts by defining a repeatable task, records the steps in the process, debugs the macro until it is right, edits the macro to add loop logic, and stores the resulting macro for future use.

VIII. ISSUES, PITFALLS, AND PROBLEMS IN COURSE DESIGN AND DELIVERY

The course should be a mix of concepts, analytical approaches, and software functions. Although concepts and analytical approaches have the highest potential for long term effect on behavior, they are learned best when they can be tied to specific software functions and features. Students are motivated in the course by learning new software functions. They need simple, concrete demonstrations and hands-on exercises. Principles are learned best when embedded in exercises; understanding comes from first doing and then articulating abstract concepts.

To spread out the assignments over the term, the software assignments must begin very early, even though many concepts have not been presented. This suggests that very simple, easy to apply functions (such as templates) be assigned first and more difficult functions be assigned later in the course.

We have taught the course as an elective over three years. The description for the course indicates it is for those with limited technical backgrounds. Section sizes have been 20 to 30. Interestingly, students with technical backgrounds in information systems but without the concepts and skills taught in the course do better than students with little information systems background. Students with limited backgrounds often have difficulty in learning the features taught in the course because they lack the underlying basic skills.

Experience suggests individual projects are useful in achieving course objectives. The issues include whether students should propose the project or the instructor should assign. Both approaches have worked for us.

When the course was first offered, we assumed that HELP facilities would be sufficient for presenting the details of features and functions. This turned out to not be so. HELP, even with examples, is often insufficient. Although most of the assignments deal with relatively simple software functions, students need a walkthrough demonstration. The instructions for features often contain assumptions or unspecified things that must be done. Completing the first exercise with new software features can be overly time consuming for students unless there are detailed instructions for a demonstration and students have done the demonstration themselves (or at least seen it done).

Database instruction was divided into two parts. The first made the assumption that the organization provided a database and the student had to learn to access it and retrieve from it. This part placed the focus on retrieval and report generation. Issues of database design were avoided. The second part addressed issues of building a simple end user database. Student assignments included building a database with at least two tables. The database building exercise was followed by discussion and exercises related to such topics as error control, use of data entry forms, and data validation.

Instruction in using software frequently omits any discussion of error control and practices that reduce error rates. In terms of productivity, errors are a high-cost type of rework and can have serious consequences if not detected and removed. Therefore, an important topic addressed the issue of error control. User practices, functions, and features to reduce input errors and detect errors were presented and applied in exercises.

End user application development is an optional part of the course Two types of end user applications are included: database software and a development language, Visual Basic. Students found the end user development projects interesting and instructive. However, the time available was not sufficient to do more than introduce the tools and do simple exercises.

The course worked well when taught in a room in which each student had a computer and could do simple exercises in real time as a function was presented. The second best alternative was a live demonstration in class with students having access to a computer lab after class. The course does not work well without a live demonstration. This point should be emphasized. Demonstrations and hands on experience are vital ingredients. With a laboratory that provides a computer for each student, demonstrations can walk through an example (with students not using their computers). The demonstration is followed by an example in which the students repeat the same type of exercise. This approach is successful because the students reinforce the learning from the demonstration and also because they help one another.

IX. SUMMARY

The IS'97.2 course, *Personal Productivity with Information Technology*, represents an innovative addition to the courses taught by the information systems faculty, both as part of the set of courses taken by IS majors and also as a service course. The course extends the elementary skills typically provided in a software toolkit course or modules. The emphasis is on using information technology to be more productive as a knowledge worker. The course explains knowledge work and how productivity is achieved. It presents ways to analyze requirements for knowledge work and provides practical, hands-on instruction in software functions and features that typically aid in improving productivity.

This tutorial has described the objectives of the course and assumptions about the student's prior computer and software knowledge. Approaches to improved productivity were summarized to give insight into the nature of the concepts for the course. Ways in which information technology can be used to achieve productivity improvements were presented along with examples of processes and software functions and features to be taught in the course. A sample syllabus of topics and exercises illustrate a possible structure for the course. The issue of textbook support was explored. Insights obtained from offering the course were noted.

EDITOR'S NOTE: This article was received on April 1,1999. It was with the authors approximately 2 months for revision. It was published on June 30, 1999. The article is based on a tutorial presentation delivered by the authors at the Fourth Americas Conference on Information Systems, August 14-16, 1998, Baltimore, Maryland

REFERENCES

Davis, G. B., J. T. Gorgone, J. D. Couger, D. L. Feinstein, H. E. Longenecker Jr. (1997), *IS'97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems,* Jointly Sponsored by the Association for Computing Machinery, Association for Information Systems, and Association of Information Technology Professionals.

Davis, G. B., and J. D. Naumann (1997), *Personal Productivity with Information Technology*, New York: McGraw-Hill.

Drucker, P. F. (1969), *The Age of Discontinuity*, New York: Harper & Row.

APPENDIX A

IS'97 AVAILABILITY AND IS'97.2 COURSE DESCRIPTION

AVAILABILITY

The IS'97 curriculum report (Davis et al., 1997), is available through the three participating organizations, in printed form from two of the organizations and accessible on the web in both pdf format and html from the third.

- Included in the ACM journal, *Database*, Vol. 28, No. 1, Winter 1997.

- Available as a report from the Association for Information Technology Professionals (AITP), (formerly DPMA). See their website for instructions on ordering**:** http://www.aitp.org**.**
- Accessible electronically (AIS) website: http://www.aisnet.org/**.** Select Resources and then choose either of the following:

a. The link to the portable document format (pdf) and download using Adobe Acrobat. The document is about 100 pages.

b. The link to the htm version and read using the web browser.

COURSE DESCRIPTION

IS'97.2 — Personal Productivity with IS Technology (Prerequisite: IS'97.PO)

Scope: This course enables students to improve their skills as knowledge workers through effective and efficient use of packaged software. It covers both individual and group work. The emphasis is on productivity concepts and how to achieve them through small systems.

Topics: End user systems versus organization systems; analysis of knowledge work and its requirements; knowledge work productivity concepts; software functionality to support personal and group productivity; organization and management of software and data; accessing organization data; accessing external data; selecting a computer solution; developing a macro program by doing; designing and implementing a user interface; developing a solution using database software; refining and extending individual and group information management activities.

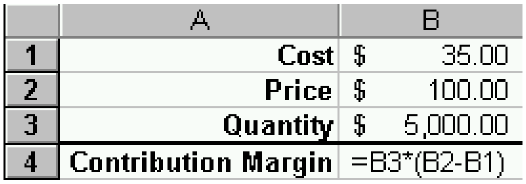
APPENDIX B

A STUDENT EXERCISE ILLUSTRATING THREE PRODUCTIVITY PRINCIPLES AND FEATURES

(SENSITIVITY ANALYSIS USING EXCEL 97)

This exercise covers three features that could be presented separately. Presenting them in the same exercise illustrates how various features can be used together to build a complete solution.

A sensitivity analysis consists of set of formulas that rely on one or more input values and then generate one or more results. For example, a person might create a worksheet to predict the contribution margin of a proposed new product. The model uses direct cost, price, and quantity as inputs and the contribution margin as the output (Figure 1).



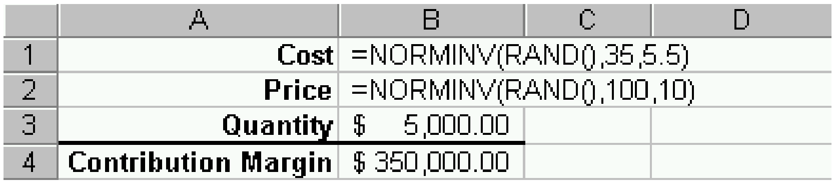
	A	B
1	Cost	$ 35.00
2	Price	$ 100.00
3	Quantity	$ 5,000.00
4	Contribution Margin	=B3*(B2-B1)

Figure 1. Simple Sensitivity Analysis

This worksheet allows a knowledge worker to quickly determine the expected contribution margin for a new product if the direct costs are 40 instead of 35, or if the quantity sold is only 500 instead of 5,000, or if the price needed to sell 5,000 is 85 instead of 100. This model works well for best-case and worst-case scenarios; however, the best and worst are not what we expect to happen. They do not represent our best guess of what contribution margin the new product will provide, nor do they indicate a level of confidence about the best guess.

Suppose our experience with price and cost estimates has been that 95 percent of the time the estimate is within 10 percent of the estimated price and that the deviation from the estimates are normally distributed. In this situation, we expect the cost to be $35; however, we are much more comfortable to assert that the price will be between $31.50 and $38.50. Likewise, we expect the price needed to sell 5,000 units will be between $90 and 110. How do we express our best guess at the contribution margin this product will generate?

One way to do this is by simulation. We can describe the cost and price parameters as functions that get their values from a particular distribution. We can recalculate the worksheet several times to see the different values generated by the formula for the contribution margin. Using Excel's norminv function, we can select a value from a normal distribution based on a relative position on the specified normal curve. If we combine the norminv function with the rand function, we can let Excel decide which part of the normal curve to use in selecting the cost and price for a particular observation of the projected contribution margin. Figure 2 shows the formulas used to randomly generate the cost and price for the simulation.

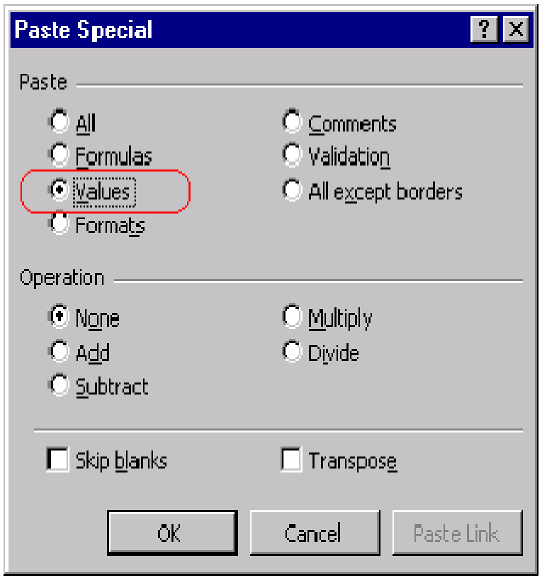


	A	B	C	D
1	**Cost**	=NORMINV(RAND(),35,5.5)		
2	**Price**	=NORMINV(RAND(),100,10)		
3	**Quantity**	$ 5,000.00		
4	**Contribution Margin**	$ 350,000.00		

Figure 2. Sensitivity Analysis Configured for Simulation

The next step is to recalculate the worksheet many times and to record the different values observed for the contribution margin. In order to do this, we need spreadsheet cells to store the observed data. In this example, column D is used for the observation data. After each recalculation (F9 forces a recalculation), we need to store the observed contribution margin. To do this in this example, the user must select the cell (B4) containing the result and select

“Copy” from the “Edit” menu. Move to cell D1 and select “Paste Special” from the “Edit” Menu. This will invoke the dialog box shown in Figure 3.



Paste Special

Paste

○ All ○ Comments
○ Formulas ○ Validation
● Values ○ All except borders
○ Formats

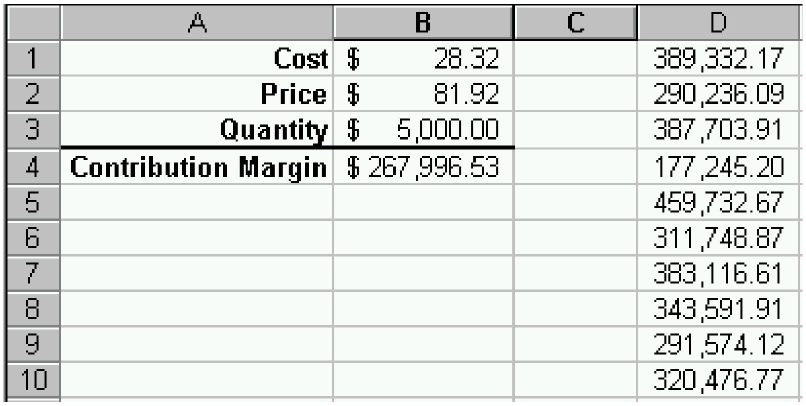
Operation

● None ○ Multiply
○ Add ○ Divide
○ Subtract

□ Skip blanks □ Transpose

Figure 3. “Paste Special” Dialog Box

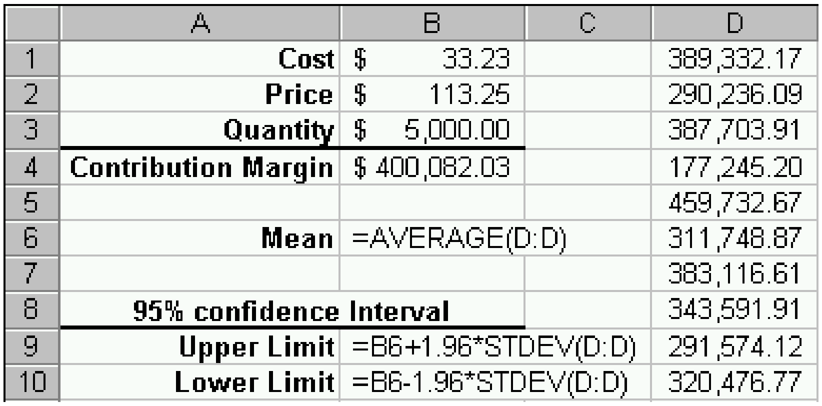
The user must select “Values” as highlighted in Figure 3, then click the “OK” button. This will paste the value generated by a particular combination of possible values for both price and cost. After the user has done this process 10 times the worksheet will have 10 values as shown in Figure 4.



	A	B	C	D
1	Cost	$ 28.32		389,332.17
2	Price	$ 81.92		290,236.09
3	Quantity	$ 5,000.00		387,703.91
4	Contribution Margin	$ 267,996.53		177,245.20
5				459,732.67
6				311,748.87
7				383,116.61
8				343,591.91
9				291,574.12
10				320,476.77

Figure 4. Simulation Results from 10 Observations

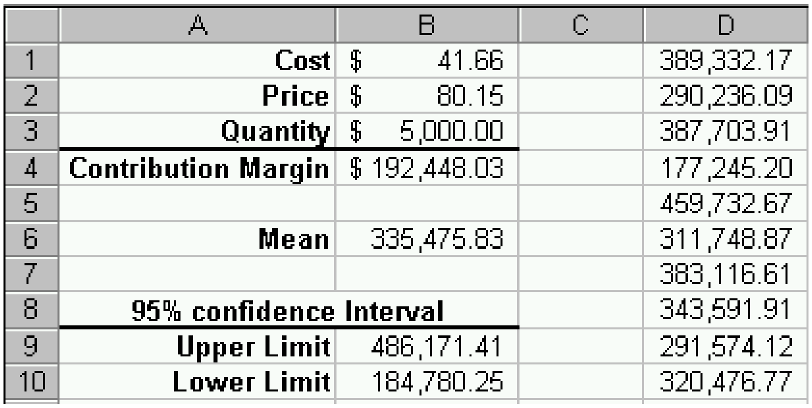
Each of these 10 observations represents a possible contribution margin for the proposed new product. Although the observations presented will not exactly match the observations of another user (because of the use of the random function), they should be similar. A quick review of these observations indicates that the highest value observed is over $459,000 and the lowest is less than $178,000. Although these values give us some idea of a range, a better way to describe this data is with its mean and confidence interval. We can calculate the mean by using the “average” function. Since a 95 percent confidence interval is 1.96 standard deviations on each side of the mean, we will calculate an upper and a lower boundary using the “stddev” function. These functions are shown in Figure 5.



	A	B	C	D
1	**Cost**	$ 33.23		389,332.17
2	**Price**	$ 113.25		290,236.09
3	**Quantity**	$ 5,000.00		387,703.91
4	**Contribution Margin**	$ 400,082.03		177,245.20
5				459,732.67
6	**Mean**	=AVERAGE(D:D)		311,748.87
7				383,116.61
8	**95% confidence Interval**			343,591.91
9	**Upper Limit**	=B6+1.96*STDEV(D:D)		291,574.12
10	**Lower Limit**	=B6-1.96*STDEV(D:D)		320,476.77

Figure 5. Formulas for Mean and Confidence Interval

The formulas shown in Figure 5 use (D:D) to indicate the entire column. As a result, as more observation are generated, they will automatically be taken into account in the mean and confidence interval calculations. Figure 6 shows the values calculated for mean and confidence interval for 10 observations of the contribution margin.



	A	B	C	D
1	**Cost**	$ 41.66		389,332.17
2	**Price**	$ 80.15		290,236.09
3	**Quantity**	$ 5,000.00		387,703.91
4	**Contribution Margin**	$ 192,448.03		177,245.20
5				459,732.67
6	**Mean**	335,475.83		311,748.87
7				383,116.61
8	**95% confidence Interval**			343,591.91
9	**Upper Limit**	486,171.41		291,574.12
10	**Lower Limit**	184,780.25		320,476.77

Figure 6. Values for Mean and Confidence Interval

With 10 potential outcomes, the expected contribution margin is about $335,000. The confidence interval is quite wide, being greater than $300,000. Since the standard deviation becomes smaller as the number of observations increases, the estimate is improved by making many more observations. Although we could add observations by continuing the process of recalculating the worksheet and then copying and pasting the values into the data column, we can automate that process, in part to speed up the process of recording various observations, but also to reduce the possibility of operator error.

In expanding the basic spreadsheet sensitivity analysis, three features of spreadsheets will be employed. They illustrate they way productivity may be improved both by increasing efficiency and by improving the usefulness of the solution. The three features to be implemented are as follows:

Feature	**Productivity principle**
Record Macro to select and store observations	Reuse of a procedure
Assign a keyboard shortcut to run macro	Efficiency (using keystroke preferences)
Add a loop to a recorded macro to more fully automate a procedure	Efficiency (automate loop) Effectiveness (make a more general, more useful application)

A recorded macro saves a set of keystrokes and/or menu commands that can later be invoked to perform a particular task. To record this macro, use the following steps:

1. Select cell A1. (This is just to make sure we are starting at the same place)

2. Begin recording a macro by choosing [Tools | Macro | Record New Macro] from the menu bar as shown in Figure 7.

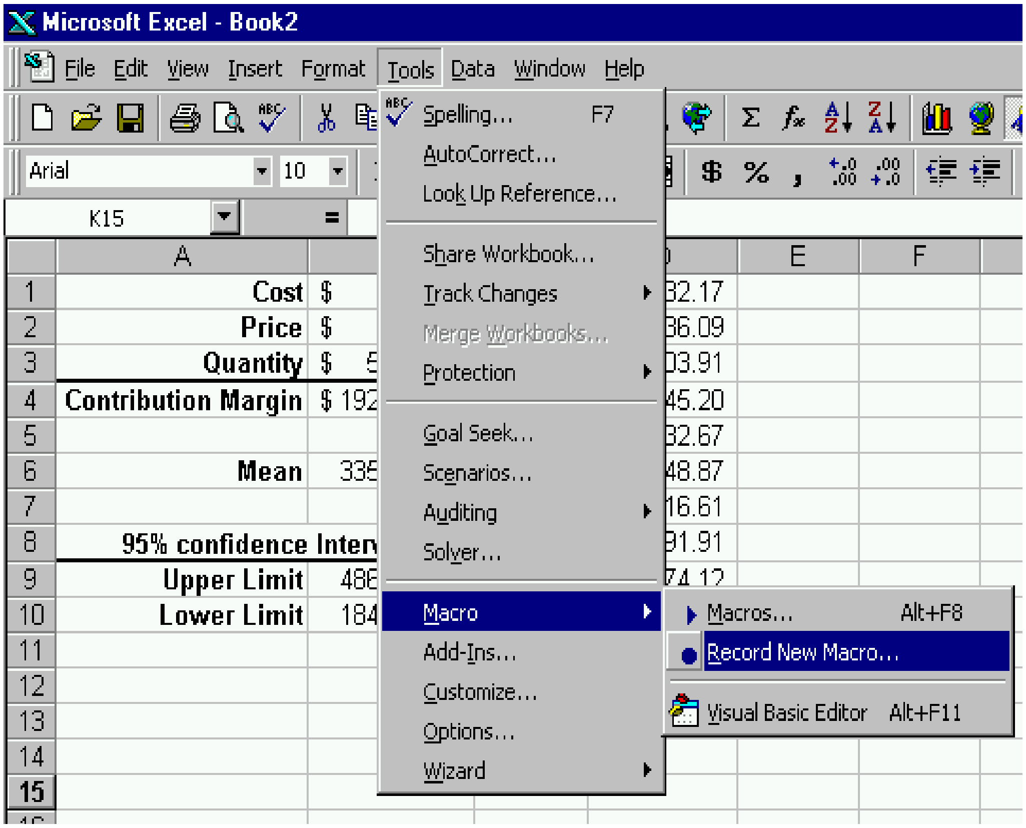
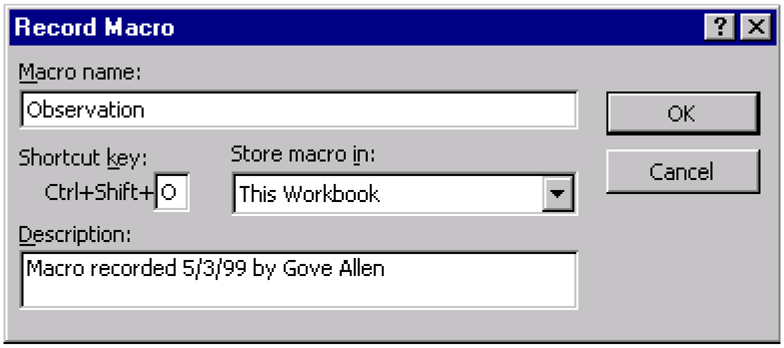


Figure 7. Recording a Macro

3. In the “Record Macro” dialog box, enter a name for the macro name and capital “O” as the short cut key, then click “OK” as shown in Figure 8. This step allows the macro to be executed either by selecting the name from the list of macros (by choosing [Tools | Macro | Macros] form the menu bar) or by pressing

ctrl+shift+O from the keyboard. (Assigning this macro to a keyboard shortcut is an example of increasing productivity by setting efficiency preferences.)



Macro name: Observation

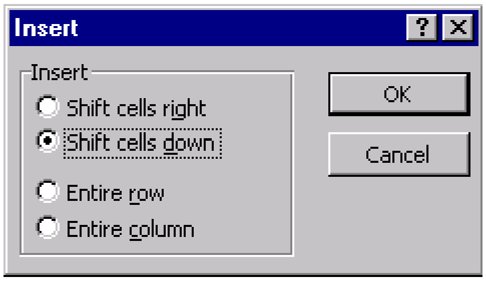
Shortcut key: Ctrl+Shift+O

Store macro in: This Workbook

Description: Macro recorded 5/3/99 by Gove Allen

Figure 8. Record Macro Dialog Box

4. All operations from this point until stop will be recorded into the macro. To make room in a list for each observation starting at D1, select cell D1 then choose [Insert | Cell] from the menu bar. The Insert dialog box is invoked, as shown in Figure 9. Select "Shift Cells Down" and then click "OK."



Insert

○ Shift cells right
● Shift cells down
○ Entire row
○ Entire column

Figure 9. Insert dialog box—Making Room for More Data

5. Press F9 on the keyboard to force the worksheet to recalculate

6. Copy the new observation of the contribution margin (from cell B6)

7. Select Cell D1 and paste the value of the newly copied contribution margin observation (Edit | Paste Special | Values)

8. The macro is now complete. Click the stop button (the button on the left in Figure 10) on the record macro toolbar. This tool bar appears while recording a macro.

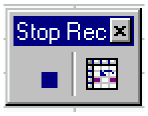
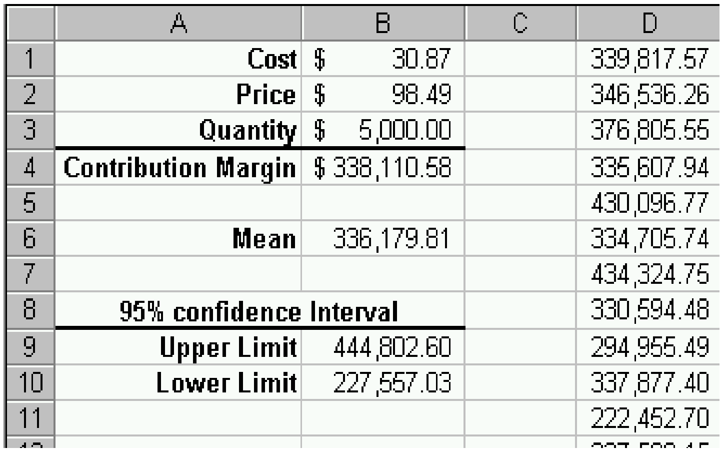


Figure 10: Ending the Macro

The user invokes (plays) the macro by pressing ctrl+shift+O to generate an observation and record it. By pressing the keys to play the macro 89 more times (for a total of 100 observations), the width of the confidence interval for the data is reduced from over $300,000 to under $220,000, as shown in Figure 11.

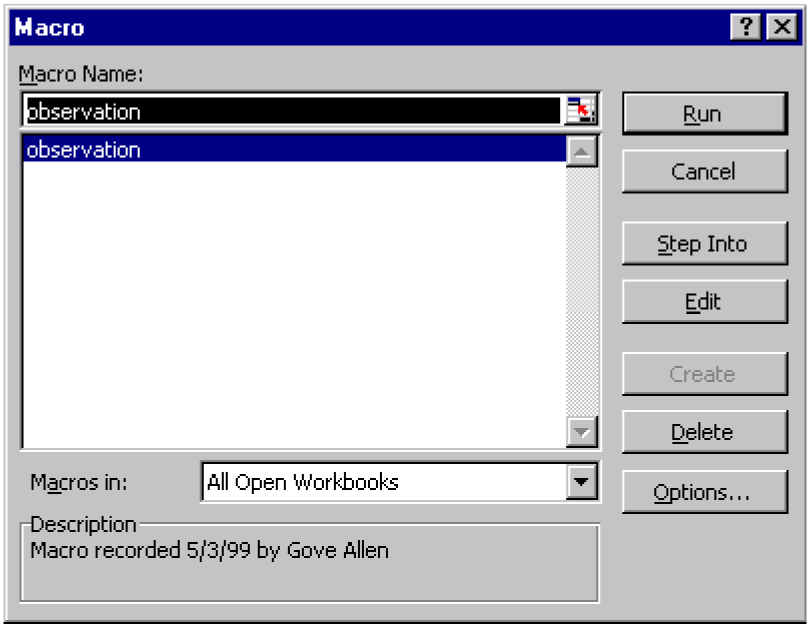


	A	B	C	D
1	Cost	$ 30.87		339,817.57
2	Price	$ 98.49		346,536.26
3	Quantity	$ 5,000.00		376,805.55
4	Contribution Margin	$ 338,110.58		335,607.94
5				430,096.77
6	Mean	336,179.81		334,705.74
7				434,324.75
8	95% confidence Interval			330,594.48
9	Upper Limit	444,802.60		294,955.49
10	Lower Limit	227,557.03		337,877.40
11				222,452.70

Figure 11. Confidence Interval after 100 Observations

If the user wishes to generate a large number of observations (such as 1000) it makes sense to modify the macro further. The modification will reduce the time needed to run the analysis and reduce possible errors in using the procedure; it also makes a more complete user-written application.

The modification is to edit the Visual Basic code generated by recording the macro. The user to selects [Tools | Macro | Macros] from the menu bar. This invokes the Macro dialog box as seen in Figure 12.



Macro Name: observation

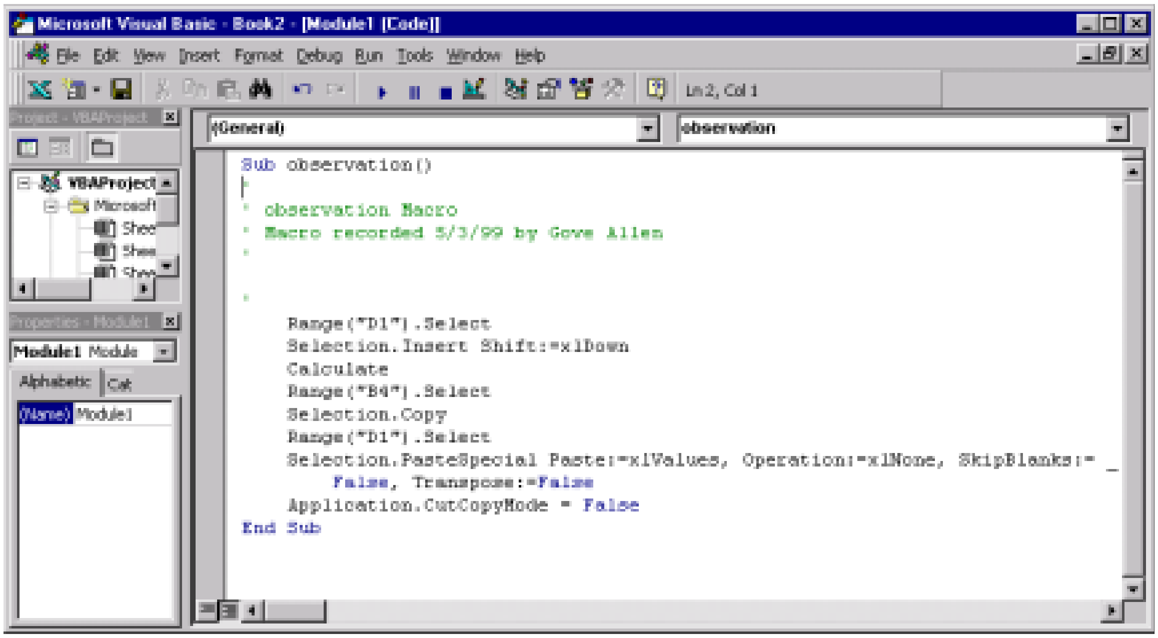
observation

Macros in: All Open Workbooks

Description
Macro recorded 5/3/99 by Gove Allen

Figure 12. Macro Dialog Box

From the macro dialog box, the user selects the macro named "Observation" and clicks the "Edit" button. The Visual Basic code for the macro appears in an editor as shown in Figure 13.

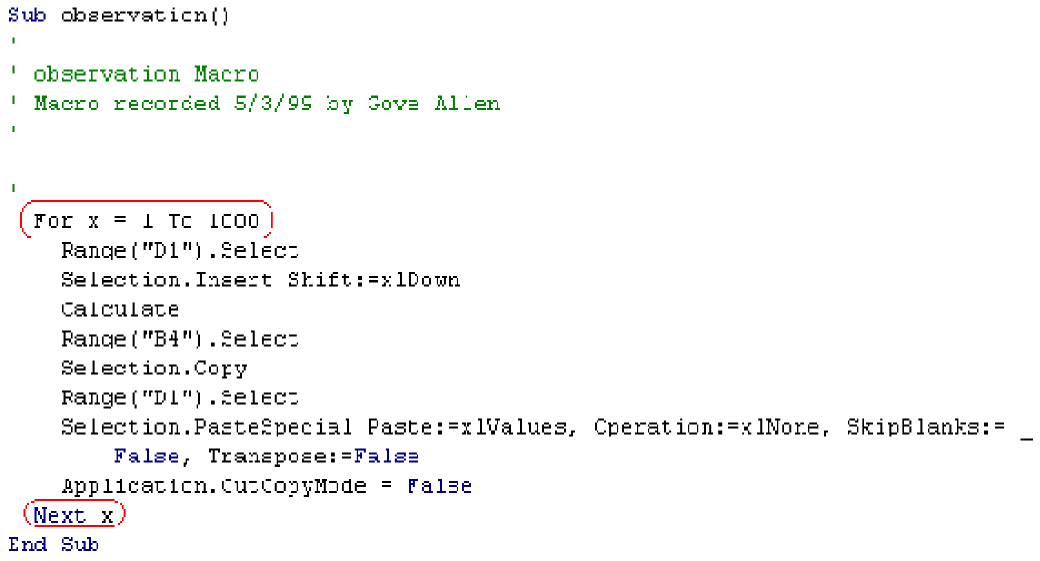


```
Sub observation()
'
' observation Macro
' Macro recorded 5/3/99 by Gove Allen
'

'
    Range("D1").Select
    Selection.Insert Shift:=xlDown
    Calculate
    Range("B4").Select
    Selection.Copy
    Range("D1").Select
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
    Application.CutCopyMode = False
End Sub
```

Figure 13. Visual Basic Editor and Code for Observation Macro

Using the Visual Basic editor, the user adds a loop around the recorded statements to make them repeat the desired number of times. Figure 14 shows a “For…Next loop” added to the code to make it iterate 1,000 times.



```
Sub observation()
'
' observation Macro
' Macro recorded 5/3/99 by Gove Allen
'

'
 For x = 1 To 1000
    Range("D1").Select
    Selection.Insert Shift:=xlDown
    Calculate
    Range("B4").Select
    Selection.Copy
    Range("D1").Select
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
    Application.CutCopyMode = False
 Next x
End Sub
```

Figure 14. For...Next Loop

When the macro with the loop is run, it will generate 1,000 new observations. In other words, editing the Visual Basic code recorded by the macro recorder, we have changed the set of instructions from a single recorded macro to the beginnings of an end user application.

ABOUT THE AUTHORS

Gordon B. Davis holds an endowed chair as Honeywell Professor of Management Information Systems in the Carlson School of Management at the University of Minnesota. He is a pioneer in the field and has been active in the major initiatives to establish the academic discipline. His current research interests are conceptual foundations for the field of information systems, management of the IS function, and personal productivity with information systems.

J. David Naumann is associate professor of Management Information Systems in the Carlson School of Management at the University of Minnesota. He is an alumnus of the University of Minnesota Professor Naumann's research focuses on the process of information systems development, and he has published articles in the leading MIS journals. He co-authored the textbook *Personal Productivity with Information Technology* (McGraw-Hill) with Gordon B. Davis. He is active nationally in MIS curriculum development, as a member of the IS Curriculum '97 Task Force sponsored by the three leading MIS professional organizations: ACM, DPMA and AITP. He designed and operates the IS Faculty Directory, a World Wide Web international database of 4,000 IS researchers. He teaches telecommunications at both the MBA and undergraduate levels. He also teaches applications development and systems analysis and design courses.

Gove N. Allen is a doctoral student in the Information and Decision Sciences Department, Carlson School of Management, University of Minnesota. He received a BS in Accounting and an MAcc with an emphasis in Information Systems from Brigham Young University in 1994. His primary teaching responsibilities and research interests are in the areas of information system development, logical database design, management of intellectual property, and user interface design. From 1993 to 1997, he operated a training and consulting firm specializing in relational database design and implementation and object-oriented client/server development environments, working with such companies as Sony, AT&T, Sprint, MCI, NASA, Kennedy Space Center, The Money Store, American Express, and Lincoln Life.

Communications of the Association for Information Systems