Developing Employment-Related Office Technology Skills

Judith J. Lambrecht
Dept of Work, Community and Family Education
University of Minnesota

October 1999

Project 98-04

National Center for Research in Vocational Education

Supported by
The Office of Vocational and Adult Education,
U.S. Department of Education

MDS 1199

Executive Summary

The purpose of this project was to develop guidelines for teaching employment-related office technology skills, particularly those offered in office technology, based on the practices in schools already judged to be exemplary in their program design and teaching practices. A key assumption in asking about effective teaching practices was that teachers and students would need to balance three elements in learning to use desktop software applications:

- How the technology operates
- Business concepts being applied
- Expectations of a given work setting

Teaching software effectively is becoming increasingly important. This is particularly true for general-purpose business software such as word processors, spreadsheets, databases, graphics presentation, and telecommunications. If almost all employed workers will need to use some portions of such software, clearly, large numbers of people need to learn to use general business applications software. Most importantly, they need to transfer their knowledge from the learning setting to business settings, and they need to be able to continue to learn new software as versions change and new software displaces old.

The dynamic nature of business technology makes questions about what to teach, content questions, prominent in the vocational education field. This is especially the case in the business education literature, since the need to teach computing technology is compelling for business occupations. Much attention has been given to what technology businesses are using and the employment competencies students need to work in such settings. Less attention has been given to how to teach technology skills effectively for employment purposes. This project focused on teaching practices.

This research allowed description of the teaching and learning practices in exemplary programs at mainly the postsecondary level from the view of students, teachers, and employers. These secondary and postsecondary programs were judged exemplary by state- and national-level professional staff because of their reputations for responsiveness to current employment needs, innovative programs, comprehensive office technology programs, and consistent student employment placement. These programs have been innovative in the variety of program scheduling options made available to students and their responsiveness to diverse student needs.

In five technical/community colleges and one technical high school, at least three students, at least three (or all) teachers, and three employers were interviewed over the course of a six-month time period. A total of 48 teachers, students and employers of program interns were interviewed. Each school was visited one to three times to observe classes and talk with these three groups. A critical-incident approach was used to identify

particularly noteworthy program aspects. Pilot-testing of the interview showed that it was easiest for teachers and students to think about challenging software aspects to teach/learn. These interviews were the basis for developing program development guidelines.

Two key findings were apparent from this research. One was the largely decontextualized teaching of introductory software use, even when business examples are the textbook frame of reference. The second was the dominance of systematic teaching practices over more discovery-oriented minimalist teaching practices. A third finding of this study was concern about the image of various employment fields, particularly the contrast between "office technology" and "information technology." This image affects whether programs will be offered, who will teach them, and how they will be taught.

With regard to the first finding, the question of how teachers and students balance the teaching/learning of software operation in conjunction with business content, the finding was that instruction in software operation dominates. Teachers saw themselves primarily as software instructors. This is especially true in the beginning stages of instruction. Business content becomes prominent at advanced courses that explicitly focus on office operations and the use of technology as a support tool. The primary employment use for technology was that of facilitating office communications. This meant that if there were any prerequisite knowledges or skills that students needed, they were keyboarding skills and basic written English communication skills.

With regard to the second key finding, the question about specific teaching practices, the teaching assumptions implicit in the most popular instructional materials showed the most prominent approach to be systematic, step-by-step processes that engage students in comprehensive software instruction. An alternative has been suggested by research, largely in industry settings, for a minimalist approach to teaching software. While it was expected that this approach would be observed in exemplary school settings, it was conspicuously reserved for the most advanced levels of instruction, if it was present at all. Possible explanations for strong preference for structured, systematic instructional approaches and the lack of awareness for minimalist techniques are included in the discussion of the findings.

According to the interviews with all three groups, students, teachers, and employers, the open-ended office tasks encountered in the advanced coursework were also likely to be encountered on the job. Realistic, work-like projects reinforced the "basic skills" that businesses say they expect schools to develop—computing skills, attitude and work ethic, written and oral communications, keyboarding, handling telephones and voice mail, using basic equipment such as copiers and fax machines, and handling mail services. Even as employers expect students to come to entry-level jobs with these skills, they also recognize that students will be learning specific work procedures once employed. The challenge for students as employees is to be responsive to these learning opportunities.

The ability to learn in an office setting appears to be dependent upon possessing the dispositional traits that are required for being part of a support staff. An important part of what students learn in Office Technology programs is the kind of work that they will actually be expected to do. Interpreting expectations and priorities and knowing how to ask questions were frequently part of the traits or competencies that could only be learned on the job. Learning such tacit skills is discussed in terms of Discourse learning, or moving from students' primary Discourse of schooling to the Discourse(s) of employment settings. An implication of thinking of technology-related education as being part of a Discourse is that the systematic approach to instruction is framed in the Discourse of the tool and its functionality. Instruction is *about* the tool. On the other hand, the minimalist approach is framed in the Discourse of the office. Moreover, instruction is *within* the Discourse, not about it. That is, the knowledge constructed is largely tacit and relates to how office workers, as a type of people, do things, as opposed to what the tools can do.

In summary, two different goals can be prominent in employment-related programs. One implicitly recognizes the situation of schooling. The other very explicitly focuses on the eventual employment goals. When the goal is developing employment-related office technology skills, balance is needed between gaining technology skills and understanding the eventual work settings in which such skills have meaning.

One goal can be software skill development. If so, then the skill-hierarchy model should prove helpful in interpreting the target skill level with various tools. Curriculum content and instructional materials are chosen accordingly. On the other hand, the goal of instruction is also movement out of the schooling Discourse and entry into the employed-worker Discourse—to become a certain kind of person. If this is the case, then the model of skill acquisition needs to be applied to fluency with the ways of being an employed worker, not simply to facility with the tools of the trade.

When success in a work setting is the goal, the Discourse of an employment setting should eventually become more dominant than the Discourse of schooling. This means the sooner the context and content of the employment field can become dominant in the learning setting, the better for allowing student participation in the employment Discourse. The instructional guidelines gleaned from exemplary office technology school settings suggest ways in which students might be assisted in their progress as they move from schooling Discourses into work-related Discourses for using technology.

Table of Contents

Developing Employment-related office technology skills

Chapter I - Introduction

A series of federal reports and daily news articles warn of shortages of information technology workers. It can be difficult to define just exactly what types of jobs are available and where the shortages exist--technology permeates work of all kinds. What is more, the specific computing technology available in today's work sites is constantly changing. Advancing technological capabilities and the need for businesses to remain competitive propel these changes. Even with the dynamic nature of the jobs and the on-going changes in the technology that potential employees need to know, schools must offer technology-related employment preparation. This project singles out a segment of technology-related jobs and asks how exemplary programs are preparing students for employment. The intent is to develop program guidelines that can be helpful to other secondary and postsecondary schools.

This chapter will provide an introduction to four conceptual aspects of this project:

- need to teach applications software skills,
- assumptions about what students need to know to use software effectively,
- current assumptions about teaching technology skills, and
- specific research questions raised in this study.

Chapter II reviews literature related to teaching end-user computing skills; Chapter III describes the research methodology; Chapter IV presents the findings from the critical incident interview analysis; Chapter V presents a discussion of findings and further research implications, and Chapter VI presents guidelines for effective teaching of technology skills based on the practices in the exemplary schools that were part of this project.

Need to Teach Application Software Skills

To narrow the scope of the study to a manageable, yet significant, use of workplace technology, those applications used most in office settings by "computer end users" are the focus of attention (Regan, 1996). These are commonly called "applications software" such as word processing, desktop publishing, spreadsheets, databases, presentation graphics, and telecommunications software. All of these software applications are used increasingly within networked, telecommunication-intensive business environments. Use of these applications, with diverse levels of expertise, is assumed to be part of the technology competencies currently required by at least 65 percent of the workforce and projected to be needed by 95 percent of workers by the year 2000 (Oblinger & Maruyama, 1996; Oblinger & Rush, 1997). Schiller (1999) corroborates these estimates using U.S. Dept of Commerce data:

> Between 1984 and 1993, the percentage of U.S. workers using computers doubled, from one-fourth to nearly one-half. In 1996, 7.4 million people worked in the U.S. information technology industry, while industries that were major users of information technology employed about half the workforce. (pp. 15-16)

The broad extent of use of personal computers and the consequent need for basic "computer literacy" can be illustrated by this data about computer use in the United States from *The Economist* (Strassmann, 1997). The United States in the middle 1990s lead the rest of the world in the number of personal computers per 100 people with a ratio of 35:100.

> The 1995 population of the US was 263 million, with a labor force of 123 million, of which 71 million were in information occupations. The ratio of 35 computers per 100 people suggests that a significant share of the installed base of computers must be in households. With 92 million computers installed, the US would have about half of the world's computers. This also suggests that most of the people in information occupations already have a computer for their business uses. (p. 229)

Strassmann (1997) identifies the office as the place where the microcomputer has had an especially large impact on how work is done:

> The overwhelming acceptance of the microcomputer by office workers is a phenomenon that hardly anyone foresaw. A business systems analyst, a financial controller, or a senior executive in the 1970s would find it inconceivable that, within fifteen years, approximately three-quarters of US office workers would operate their own computers at their own desks. Hardly anybody in the computer industry could imagine that individuals would possess exclusive access to the calculating power of a 1972 mainframe computer and the storage capacity of a 1965 major data center. (p. 223)

When these "people in information occupations" are looked at more closely, particularly those using computers in white-collar work, the US 1995 Bureau of the Census data describes 51 million of these people as using the kind of software of primary interest in this study: word processing spreadsheets, databases, desktop publishing, and communications (Sennett, 1998, p. 155). Sennett uses these data in his book, *The Corrosion of Character: The Personal Consequence of Work in the New Capitalism*, to support his argument that "the computer is now used in virtually all jobs, in many ways, by people of all ranks" (p. 24).

While workforce use of computing technologies includes a wide range of computing applications, executives in Fortune 500 companies have consistently identified the general-purpose computing applications listed above as the major ones needed by business students for employment (Arney, 1998; Zhao, 1996; Zhao, Ray, Dye, & David, 1998). When people talk about teaching people basic job skills, these are the basic computing skills that are generally implied—skills intended to transfer to a wide variety of employment settings to be used by entry-level workers as well as professionals doing their own document preparation and record keeping. Nardi and O'Day (1999) talk about such skills in the following fashion as they describe the training programs that are necessary for "the kind of catch-up we have to play when students do not have basic computer experience in schools." … "Training will include basic literacy, office skills, word processing, spreadsheets, and database management—skills needed in the marketplace" (p. 196).

The wide penetration of personal computers into homes and businesses means that schools have been compelled to think about how to engage all students in learning about computers and also about how to use computers to support instruction. As they have done this, schools have chosen to use the same types of productivity software identified above for employment, expanding the interest in teaching such software to a wide range of student audiences. Current data about software use in schools is an indication of the domination of applications software, as it is often called. A national survey of 655 United States K-12 schools in 1998 found that the top four software programs available on 80 percent or more of the computers in schools were word processing, spreadsheet, database, and graphics software (Anderson & Ronnkvist, 1999). This domination of "productivity software" use by wide audiences has an effect on the breadth of student groups to be served and type of instruction developed in any school setting, even if the programs are identified as specifically for employment.

Because employment requires effective use of computing applications, software training on these major packages is included in virtually all business education programs at the secondary and postsecondary levels. These employment preparation programs are frequently identified by such labels as "end-user computing," "desktop computing," "business applications software," "administrative office technology," "office information systems," and, most commonly, "office technology." This project sought to develop guidelines for effective instruction using desktop-computing preparation, by whatever label it might be identified.

The study began with several assumptions about what effective programs might be like. At the outset, examples of effective programs were selected by obtaining recommendations about secondary and postsecondary office technology programs that were know to be successful. "Success" meant success of students in obtaining employment related to their preparation. These programs were considered to be exemplary in that they were considered to be up to date, well run, and also innovative—trying new ideas and being models for others.

Given that exemplary programs were identified, it was anticipated that certain program characteristics might exist. A key question was whether these assumptions would be confirmed. It was possible these assumptions might not be confirmed as explicit, essential components of effective programs. In either event, examples would be provided of how exemplary schools engaged students in technology instruction. Since these assumptions about technology instruction were the starting point for the development of interview protocols and other data collection efforts, they are presented below.

Assumptions about What Students Need to Know

Students need initial computer literacy for basic information handling processes. Without some initial evidence of computer competency, students are likely to have difficulty being hired for entry-level jobs that require desktop computing skills. Such skills are the basis for being able to benefit from any on-the-job training opportunities available to extend these skills. Students need to learn to use technology in a general way that will allow transfer to a variety of on-the-job settings. Key assumptions have been made about what students need to know in order to use technology effectively. These assumptions, in turn, complement and lead to further assumptions about what facilitates the transfer of general technology competencies beyond the classroom. These assumptions are outlined below.

Assumptions about learning to use technology. Using technology well requires knowing more than merely how particular software tools operate, though this is a significant part of the learning expectations. Many software users today make minimal use of computing capabilities and features of popular packages (Landauer, 1996; Gibbs, 1997; Nardi, 1993). Effective instructional programs are able to engage students in learning a wide range of software features that have important employment applications.

Beyond knowing how software operates, any particular instance of effective technology use requires understanding two additional crucial dimensions. One dimension is understanding the human or business purposes served by the tool, or the subject-matter content necessary to understand why a particular technology is useful (Nardi, 1993, 1996; Nardi & O'Day, 1999). This means that knowing specific software features needs to be coupled with the content knowledge or domain in which the software meets some important need. Software is almost always used for some purpose other than getting the software to work. Software is not often viewed as a puzzle in and of itself. If users have no personal or work-related needs for a software feature, such as the table-of-contents feature of a word processor or the pivot table function of a spreadsheet, they are not likely to seek out these features to learn. Once introduced to a particular software feature, students are not likely to remember how it works unless they recognize that it does something personally useful—it meets a pressing need.

But, seeing the purpose for using particular software features is just the start. Content or domain knowledge, such as in the business field, is essential for using software for solving problems. These problems will be both well-structured problems and ill-structured. Well-structured problems are business needs that appear frequently and for which well-established procedures exist, such as using the mail-merge feature of a word processor. If a student or employee recognizes an occasion for which the mail-merge feature is a viable solution, the software itself will provide prompts on how to carry out the necessary procedural steps. While mastering these steps is not a trivial activity, when and how to use them is well documented.

Other business problems are less well structured. Many business problems might have software solutions that are not well established, such as using a lookup table in a spreadsheet for the novel solution of identifying employee rankings and doing salary calculations of a certain type. A logical “IF” statement might also be used to do the same thing. This kind of ill-structured problem presents the challenge for the student or employee both to understand the end goal and how to use software to get there. (For some people, this might be a well-structured problem, so it is not always easy to predict the nature of a problem as well- or ill-structured.) Computer users must determine what the business need is, or, in this example, what salary calculations are being requested, where the data come from, and what is arithmetically involved. After understanding the business problem, computer users then must select the software features needed to create an efficient solution.

The implication of asking students to use software to solve problems is that eventually they need to understand both what a particular software program can do and the business need that can take advantage of that capability. Scribner (1986) captured the importance of such subject matter knowledge for solving practical problems in this assertion:

> From earlier assumptions that problem solving can be understood in terms of “pure process,” a consensus has arisen that problem-solving procedures are bound up with the amount and organization of subject matter knowledge. Practical problem-solving research reinforces this view by disclosing the diverse forms of knowledge—strategy interactions and the complexity of the knowledge involved in even the simplest tasks. (pp. 26-27)

The contextual nature of problem-solving skills has been reaffirmed over the last decade. A recent report by Stasz and Brewer (1999) expresses the relationship between generic and context-depend skills this way:

> Work competencies, or "generic skills" as defined by SCANS and others, are thought to be broadly transferable across work settings, although they can take on different meanings in different contexts. Problem solving, for example, is a general term that represents a particular competency, but the process itself varies with different tasks or situations. Without attention to context, terms like "problem solving" or "communication" are rendered meaningless, especially as definitions of what should be taught in classrooms. (p. 8).

It is being suggested that two kinds of subject matter are needed to use software effectively to solve problems in employment settings: knowledge about how the software works, and the demands of practical problems in the work world for which the software might be useful.

Beyond knowing how software operates, a second dimension in addition to subject-matter knowledge needs to be considered for effective software use: an understanding of the broader social context in which the software is being used. This means interpreting the expectations of a particular social context, such as a business work site or another institutional setting, in which the technology is used as part of a job (Glick, 1995; Scribner, 1997; Strassmann, 1997). This assumption essentially draws on a key tenant of activity systems theory: that human individuality itself and the demonstration of human capabilities are achievable only in society (Nardi, 1996; Kuutti, 1996; Tolman, 1999). How a student or an employee decides what to do, and whether, in fact, computing software is to be part of a job task, depends on interpreting and answering several keys questions—many of which will be answered tacitly. These questions include:

- what the learner's role is in the context of a school or work setting,
- what is to be accomplished,
- which tools can be used,
- how the tools are to be used,
- how well work is to be done, or the work standards,
- who makes this decision, and
- who can be asked for assistance.

These questions imply that learners eventually need to understand their roles as part of an activity system (Jonassen & Rohrer-Murphy, 1999). Both schools and business offices consist of activity systems, although they may not be labeled and defined as such by students, teachers, or employers.

In summary, three key elements are involved when students learn to use technology. Using technology effectively is assumed to require an understanding of these three overlapping aspects of software use:

- How the technology operates
- Business concepts being applied
- Expectations of a given work setting

Assumptions about the transfer of learning. Work-related competencies developed in most school settings are to be used outside of school in ways that cannot be

anticipated with precision. Much of students' job success comes from their ability to transfer and adapt what they have learned in school to different settings—to continue to learn on the job. There continue to be concerns about fundamental "skills gaps" and mismatches between educational attainments and employment requirements. Even so, there is also growing agreement about two key concepts related to what students should be taught (Bailey, 1990; Bailey & Merritt, 1995; Glick, 1995; Hart-Landsberg, Braunger, Reder, & Cross, 1992; Merritt, 1996; Raizen, 1989; Stasz, McArthur, Lewis, & Ramsey, 1990; Stasz, Ramsey, Eden, DaVanzo, Farris, & Lewis, 1992; Stasz, Ramsey, Eden, Melamid, & Kaganoff, 1996; Thomas, Anderson, Getahun, & Cooke, 1992).
.

- There are likely to be generic skills and dispositions developed in schools that can transfer to a variety of out-of-school settings, including employment settings; and,

- These generic skills and work-related dispositions are not just features of individuals or specific jobs; rather, these skills are a feature of the workplace as a social system.

Generic skills and dispositions have been defined in a variety of ways, from the SCANS (Secretary's Commission on Achieving Necessary Skills, 1991) three-part foundational and five-part competencies listings to the Workplace Basics from the American Society for Training and Development (Carnevale, Gainer, & Meltzer, 1990). NCRVE research by Cathleen Stasz and others have identified generic skills falling into four broad categories, each of which can be more fully defined (Ramsey, Stasz, Ormseth, Eden, & Co, 1997; Stasz, et al, 1990; Stasz, et al, 1992; Stasz, et al, 1996; Stasz & Brewer, 1999):

- problem solving,
- teamwork,
- communications, and
- dispositions and attitudes.

In describing the "New Basic Skills" required for a middle-class wage, Murnane and Levy add to these general capabilities "the ability to use personal computers to carry out simple tasks like word processing" (1996, p. 32). Such a claim, as well as the emphasis on technology skills in the "New Work Skills" described by Resnick and Wirt (1996), coincides with the technology-focused instruction examined in this project.

The challenge for teachers who wish to develop these competencies on the part of students is to understand the ways in which these generic skills are intertwined with subject-matter content and the social context in which the skills are applied. While it is possible to talk about generic skills, it is probably not possible to teach generic skills directly. Learning generic skills needs the frame of reference of some subject matter and some social context of application.

These dimensions of the transfer of learning assumption are closely related to the previous set of assumptions about learning to use technology. Both subject matter content and social context are crucial aspects of what one learns to do with software tools. Activity systems theory was cited earlier as one source of theoretical support for these assumptions. Such concern for the context of using computing tools is not limited to activity systems theorists, however. Cognitive theorists have also made a similar claim for the importance of the social setting in affecting what is learned. As one example, conclusions from research about problems of the transfer of learning in cross-cultural settings (Ceci & Roazzi, 1994) makes this point succinctly and strongly:

> Our review of work that spans continents, social classes, and levels of formal education shows that the context in which learning occurs has an enormous influence on cognition, by serving to instantiate specific knowledge structure, by activating context-specific strategies, and by influencing the subject's interpretation of the task itself. Neither context nor cognition can be understood in isolation; they form an integrated system in which the cognitive skill in question becomes part of the context. To try to assess them separately is akin to trying to assess the beauty of a smile separately from the face it is part of. (p. 98)

If students' use of software is not to be limited to what they can do in a school setting, they need to understand the ways in which their in-school preparation relates to later work expectations. They need to know how to build upon their basic skills and understandings after they become employed so they can become real contributors to a work unit in business and industry. A deep level of understanding is necessary in order to recognize when the technology might be useful in a new setting--transfer of learning--and to understand the inter-relationship of different technologies and their various applications (Thomas, Anderson, Getahun, & Cooke, 1992). In short, students need to understand software well.

To accomplish this, vocational education teachers need to teach the basic operational capabilities of the technologies. They need to ensure that students use key technology features in a wide variety of applications (Bransford, Brown, & Cocking, 1999; Campione, Shapiro, & Brown, 1995; Spiro, Feltovich, Jacobson, & Coulson, 1996). What is more, teachers need to assist students in solving the types of ill-structured problems that characterize employment settings. Attaining deep levels of understanding and development of the cognitive flexibility necessary for ill-structured problem solution requires that teachers engage students in technology use that goes beyond merely getting the technology to work. The primary goal of this project was to find models of how effective teachers do this.

Assumptions about Teaching Technology Skills

Two sets of assumptions were made about the teaching practices that might be observed. First, classroom-teaching practices or methods, were open to investigation—it was not known how the actual practices would be characterized. Two major types of teaching approaches characterize computer training, a systematic approach and a discovery-oriented “minimalist” approach. These will be briefly introduced and developed more fully in the review of related literature. A second assumption was that an explicit link to work settings was necessary if questions were to be asked about students’ and teachers’ interpretations of work expectations. All schools included in this project, therefore, had formal internship programs. This rationale is presented below.

Teaching Practices or Methods for Computer Skills. Teachers have a choice between two markedly different teaching practices when providing applications software instruction. One might be called the well-structured approach, or "systematic" approach, in that the specific concepts to be learned are specified beforehand and explicit step-by-step instruction is provided to assist students in understanding and using particular software (Gagne, 1985; Gagne & Medsker,1996). This direct instruction approach has been common for both school-based technology instruction and industry-based computer training. Industry trainers have labeled this approach “The Sagamore Design Model,” though they do not explicitly identify the theoretical base implied in their teaching recommendations (Masie, 1989; see also Brandon, Clothier, Copeland, Crowell, Dodge, Maday, Masie, Perry, Slobodian, Woodie, Zemke, & Zenke,1996; and Clothier, 1996).

A contrasting approach is called "minimalist" in that it incorporates a discovery orientation to software learning based on the goals, current understandings, and expectations of the learners (Carroll, 1990, 1998). Implicit in this approach is the expectation that learners understand the needs of a particular problem, such as a business problem requiring computer solution. Few prescriptions are possible about how to design instruction, other than making available a wide variety of resources to support learning and providing prompting and guidance in response to student errors. This approach is, thus, supportive of the position that students learn through mistakes encountered.

A goal of this project is to explore the extent to which teachers recognize and use instructional approaches that might be characterized as either of these two types, systematic or minimalist. Each of these approaches will be described more fully in the review of related research.

Work-related internships. Since employment success in technology-related jobs is the goal, student experiences in work settings need to be included in the teaching practices. School programs that incorporated internships or cooperative-education placements in technology-related jobs allowed examination of such teaching practices. These work experiences also provided information about problems at work or possible transfer failures.

Some aspects of information technology may be better learned in employment settings than in school classrooms. Examining teachers', students', and employers' perspectives about technology-related learning experiences both in-school and on-the-job allowed some of these aspects to be identified.

Consideration of the workplace as an instructional site is necessary, in addition to business content and software use, when thinking about business employment using technology and assisting students to become successful in their chosen career fields. Neither understanding of subject-matter content nor knowing how to operate particular software is sufficient alone to support learning how to use technology in ways that support job performance. When the intent of in-school instruction is employment preparation, it is incumbent upon teachers to understand the ways in which the school environment differs from the work environment. Unless students are prepared for work-like environments, they may not be prepared to continue learning and to adapt their prior learning to new expectations and resource opportunities.

One model for describing the cyclical nature of learning to use technology is described in a NSF project by Dede (1996) as learning for doing, learning through doing, and learning from doing. Each level presupposes a higher level of complexity and ability on the part of learners to reflect on the purposes and consequences of their uses of technology. Stern (1992) describes a related idea of "doing by learning" for engaging in non-routine problem solving in work settings and possibly classrooms as well. A similar notion of learning through doing and from doing within the context of work settings is described by Hart-Landsberg, et al (1992) in their ethnographic study of employed secretaries. They provide a wide variety of examples of how doing a job means essentially learning a job.

The purpose of this project was to more fully understand how teachers in career-preparation programs, students in these business programs who were participating in related internships or cooperative-learning arrangements, and employer/supervisors of these internships together balance three elements:

- need to understand the purpose of their work--the content and skills required,
- capabilities of a given technology, and
- specific expectations of a given work settings

This latter area of the impact of the work setting cannot be fully explored without a more extensive ethnographic research approach than was used in this study. However, the inclusion of programs with internships or cooperative-learning arrangements allowed some exploration of the ways in which actual job settings enhance students' in-school learning experiences.

Research Questions

Business programs at both the secondary or postsecondary levels were examined to answer the following specific questions:

1. What is the content of career-preparation programs whose purpose is to prepare business students for technology use in employment settings?
   a) What representative business technologies are being taught?

b) What representative business applications illustrate technology use?
c) What assumptions are made about student backgrounds and prior subject-matter understandings that affect the content chosen and its sequencing?

2. What instructional strategies do career-preparation programs use to support the learning of applications software for business employment settings?
   a) What assumptions about learning are apparent?
   b) What are the sources of instructional materials and work-related problems for learning technology?
   c) How do assumptions about prior student background affect the use of structured, direct teaching practices in contrast to minimalism or constructivist practices for teaching software use?
   d) What types of instructional practices do students use to develop independent problem-solving capabilities?

3. What does it mean for teachers in career fields to balance and integrate
   (a) how the technology operates,
   (b) business concepts being applied, and
   (c) expectations of a given work setting.

4. How do students enrolled in information technology areas and also participating in internships or cooperative-education programs integrate understanding of
   (a) how the technology operates,
   (b) business concepts being applied, and
   (c) expectations of a given work setting.

5. Which aspects of business information technology use do teachers, students and employers perceive as better learned in employment settings in contrast to in-school experiences?

Chapter II - Related Research

While teaching technology use is a dominant concern for business teachers, the business education literature does not often probe the effectiveness of different instructional practices beyond general concerns about course structure and content selection. In the business education literature, asking what technology to teach is much more common than asking how to teach it. Recent articles have addressed questions about the need for broad employment competencies as well as about the uses of specific computing software (Alexander, 1996, 1998; Chalupa, 1997; Chalupa & Sormunen, 1996; Davis, 1997; Davis & Gonzenbach, 1995; Frueling, Kerin & Sebastian, 1997; Gonzenbach, 1998; Groneman & Buzzard, 1995; Jaderstrom, 1995; Lambrecht & Sheng, 1998; North & Worth, 1997; Nourse, 1997; Perry, 1998; Sormunen, Smith & Lane, 1996; Smith, Jones & Lane, 1997; Smith, Nelson & Mayer, 1996; Zhao, 1996; and Zhao, Ray, Dye & David, 1998). Content identification was also the purpose of a recent state-wide development effort. The State of Wisconsin completed a project directly related to Office Systems Technology program development (Wisconsin Technical College System,1998). The DACUM model of task analysis was used to identify the core competencies for graduates of Office Technology programs in the state. These core competencies fall into five major courses or broad areas:

- Administrative Procedures and Management
- Business Communications
- Business Technology
- Information Management
- Professional Business Behavior and Growth

The purpose of these several projects has been to provide direction for curriculum development at the level of general goal statements, sometimes specification of more detailed learning objectives, and for equipment and software selection rather than guidance about how to teach. In fact, McEwen (1996) has claimed "information on methods of teaching microcomputer applications is virtually nonexistent" (p. 15).

When teaching practices have been the focus of technology-related research, one approach has been to examine relationships between student success in microcomputer classes and other background factors, such as keyboarding skill, prior computer experience, and computer anxiety (Erthal, Wiggs & Huter, 1996; Hemby, 1997a, 1997b; Webler, 1992; Wiggs & Huter, 1995; and Wiggs & Erthal, 1998). LaBonty (1997) has looked at the broad structure of computing teaching practices when he examined spaced versus massed teaching schedules with either live or video instruction and their effect on student retention of skill. Distance learning for teaching computing skills is also reported by Johnson, Hill and Lankford (1998) and Bartel and Bartholome (1999). The validation of word processing testing instruments has also received attention (Bursey, 1998; Ostwald & Stulz, 1996; and Williams, 1996). A rationale for "how to teach" is not fully addressed in any of these articles.

More closely related to the issue in this study of minimalist versus systematic instructional practices, McDonald and Echternacht (1998) compared guided practice versus independent practice. They found that guided practice was more effective than independent practice for initial development of spreadsheet skills, but that independent practice was more effective for transfer of skills to an unfamiliar spreadsheet program. This effectiveness of independent practice for transfer of learning to new tasks parallels the findings of research examining minimalist instructional approaches (Carroll, 1990, 1998; Van der Meij, 1992; and Ramsey & Oatley, 1992), but McDonald and Echternacht do not cite this literature as conceptual support for what they designed. Both approaches were equally effective in teaching content knowledge about spreadsheets, and both approaches were recommended on the basis of student perceptions. The general conclusion was that "teaching with each student positioned at a computer with step-by-step instruction appears to result in higher student skill achievement than expecting students to perform the learning activities independently" (p. 20). But at the same time, McDonald and Echternacht (1998) concluded that

> In contrast, if the emphasis is on the student's ability to transfer skills obtained from a familiar program to an unfamiliar program, the use of independent-practice learning activities appears to be superior. … The teaching methodology employed needs to promote the greatest amount of transfer from the software employed in the classroom to unfamiliar programs students may face in the future. Students who work through software activities and problems independently appear to transfer those skills learned in the classroom more effectively. (p. 20)

Current methods books for business computing teachers (Lundgren, Lundgren & Mundrake, 1995; and Mundrake, 1998) follow the pattern of providing a primary focus on teaching specific types of software in a systematic manner. An unexamined assumption is that systematic, frequently student-paced, instructional practices are an appropriate choice. Industry trainers have labeled this approach "The Sagamore Design Model" (Masie, 1989; Brandon, et al, 1996 and Clothier, 1996), though they also do not explicitly identify the theoretical base implied in their teaching commendations.

The dominant information-processing basis for instruction in the field of business education is expressed more fully in Vawdrey's (1997) examination of the principles of learning in the 1997 NBEA Yearbook. When problem solving is an important instructional outcome, however, the systematic, step-by-step approach to instruction can cause problems. Lundgren, Lundgren and Mundrake (1995) make this statement about teaching software using the systematic, step-by-step approach:

> One of the major criticisms of self-paced material is the push-button nature of available resources. Students are told to press keys to perform functions at a given time. When students are confronted with "real-world" situations, they do not know what to do or why to do it. Tutorial and other self-paced materials may serve well as introduction lessons, but the test of students' abilities to adapt come from advanced applications that require critical thinking. (pp. 45-46)

Schmidt and Kirby (1995) call attention to the same risk of "cookbook" approaches to teaching software. Their comments imply the need to examine more closely the assumptions behind "push-button" instruction and something else, more "advanced," designed to support problem solving and critical thinking. The main goal of all instruction is transfer of software skills out of the classroom into the "real world." The chief concern in either school or industry training settings is to develop students' abilities to learn new software and deal with unexpected job requirements. It is important to ask what type of instruction is most likely to lead to these valued outcomes.

As suggested by the reference to "push button" versus "advanced" instruction, at least two different approaches exist for teaching computing. These approaches are labeled here as Systematic and Minimalist. The above statement by Lundgren, Lundgren, and Mundrake (1995) suggests that the different approaches might be used sequentially. However, proponents of both the Systematic and the Minimalist approaches claim them to be appropriate for novice learners, but for different reasons. Each approach makes different assumptions about how people learn. As a result, the teaching strategies and instructional materials associated with each approach are quite different. Research has shown the Minimalist Approach to be a stronger choice when transfer of learning to new problems and new software is desired, thus making it a teaching practice worth exploring (Carroll, 1990, 1998).

The conceptual basis for these two approaches will be briefly described. Table 1 summarizes and contrasts the characteristics, assumptions about learning, and teaching practices for both the Systematic and Minimalist approaches.

Systematic Computing Instruction

The basic assumptions about learning that undergird the Systematic Approach can be traced to the influential work of Robert Gagne (1985) in the Conditions of Learning and Theory of Instruction and, more recently Gagne and Medsker (1996) in The Conditions of Learning: Training Applications. Gagne's theory has been based on a behaviorist model (Joyce, Weil, & Showers, 1992) or, more recently, a cognitivist and information processing view of learning.

The diagram in Figure 1 illustrates the information-processing model of learning and shows the strong emphasis on individualistic, internal processing in response to stimuli from the external environment. This model is the basis for several widely used instructional design textbooks, such as those by Gagne, Briggs and Wager (1992), Dick and Carey (1996), and Seels and Glasgow (1990).

Table 1: Comparison of Instructional Approaches

Systematic Approach	Minimalist Approach
Characteristics	
Pre-established goals	Learning goals determined from authentic tasks
Identified prerequisites	On-going assessment of learner needs
Step by step sequenced instruction	Processes of learning modeled and coached for students with unscripted teacher responses.
Elimination of error	Use errors for instruction (see Methods, below)
Comprehensive coverage	Learners construct multiple perspectives or solutions for an issue or problem.
Emphasis on reading or tutorial pacing	Emphasis on learning by doing and exploring.
Feedback for correct responses	The criterion for success is transfer of learning.
Assumptions	
Learning causes an observable change in the learner.	Learning causes a change in perception and action potential.
Learning outcomes can be prespecified.	Specific content and outcomes cannot be prespecified, although a core knowledge domain may be specified.
Skills should be learned one at a time.	Skills are learned within social contexts.
Each new skill learned should build on previously acquired skills.	Learning focuses on the process of knowledge construction and development of reflexive awareness of that process.
Learning and knowledge are hierarchical in nature.	People construct knowledge through discussion and collaboration.
There are five types of learning: 1. verbal information, 2. intellectual skills, 3. cognitive strategies, 4. attitudes, and 5. motor skills (Gagne, 1985, pp. 47-48)	Types of learning cannot be identified independent of the content and context of learning.

Table 1: Comparison of Instructional Approaches
(continued)

Systematic Approach	Minimalist Approach
Methods	
Events of instruction: 1. Gaining attention 2. Informing learners of objective 3. Stimulating recall of prior learning 4. Presenting the content 5. Providing learning guidance 6. Eliciting performance 7. Providing feedback 8. Assessing performance 9. Enhancing retention and transfer (Gagne, 1985, pp. 246)	**Principle 1** - Choose an action oriented approach - provide an immediate opportunity to act - encourage and support exploration and innovation - respect the integrity of the learner's activity **Principle 2** - Anchor the tool in the task domain. - Select or design instructional activities that are real tasks. - The components of the instruction should reflect the task structure. **Principle 3** - Support error recognition and recovery. - Prevent mistakes whenever possible. - Provide error information when actions are error prone or when correction is difficult. - Provide error information that supports detection, diagnosis, and recovery. - Provide on-the-spot error information. **Principle 4 -** Support reading to do, study, and locate. - Be brief; don't spell out everything. - Provide closure for chapters. (Van der Meij & Carroll, 1998, p. 21)

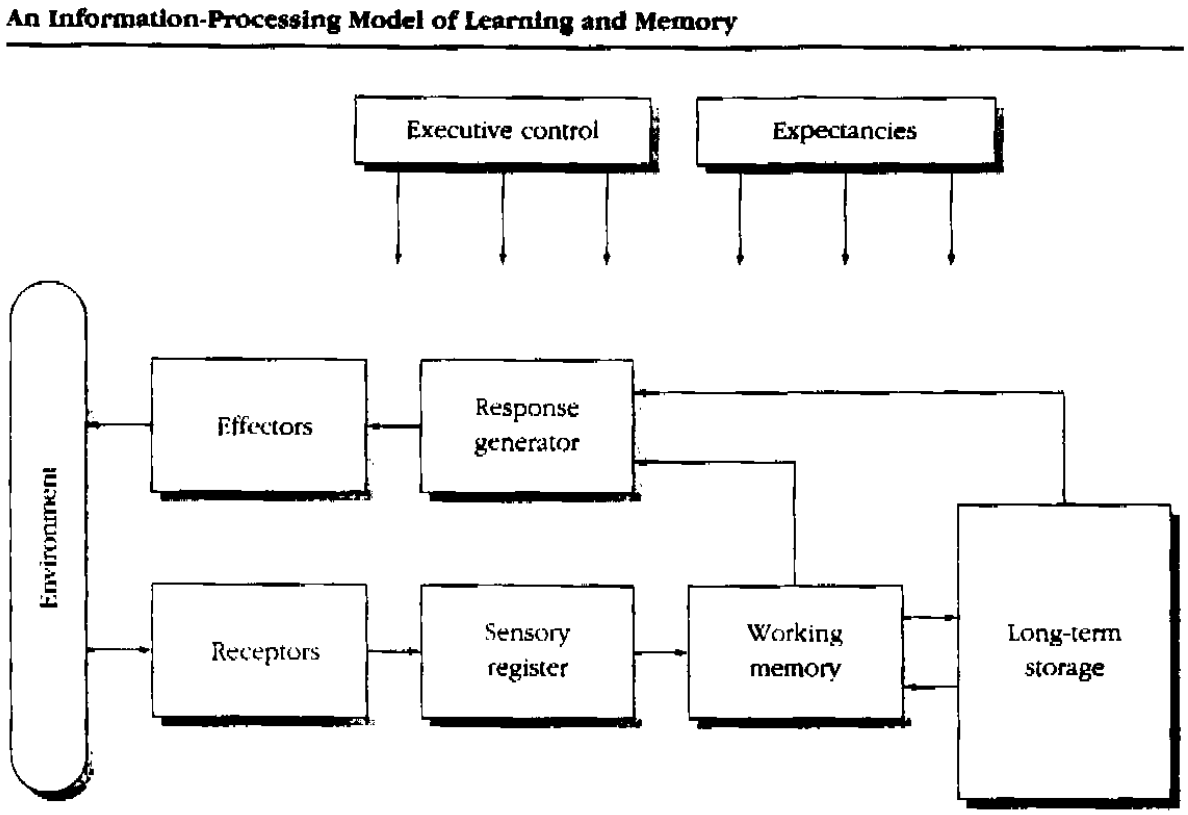

Figure 1: Information Processing Model (Gagne 1985, p. 71; Gagne & Medsker, 1996, p. 45)

To guide implementation of this model of learning, Gagne (1985) identified five types of learning and nine events of instruction to support the assumed internal processes, as listed in Table 1. Critique of any current computing applications textbook will allow all nine of these events to be identified. One of the most distinguishing features of current texts is their plentiful use of colorful screen captures to provide feedback to students about their successful completion of a series of steps. If students do not get a match, then they are asked to repeat the procedures until they do. When students have completed several exercises in a step-by-step manner, they are asked to complete open-ended problems in which the specific steps are not provided--problems more like those encountered on the job.

In the information-processing model, the assumptions about learning focus on the internal mental processing by the learner. Because of this concern for the structure of knowledge to be acquired by the learner, the most prominent aspect of systematic instruction is the content to be learned. This leads to a strong dependence on pre-specified learning outcomes, identification of these outcomes by learning type, and subsequent hierarchical task analysis to support the sequencing of teaching modules.

While the initial status of the learner is important (hence, the need for pre-assessment), and the needs and interests of students are hard to ignore, the primary focus is on what is to be learned and how this learning can be demonstrated by learner performance.

The chief benefits of the systematic approach for learning computing skills are comprehensive coverage of software features and the ability of students to work toward mastery independently in response to their current skill levels. This model is a familiar one to business educators and others who work with employment preparation or work-based training. Perhaps its well-established nature has prevented it from being labeled more directly as an object for research attention in contrast to other approaches. When such tests have been made, the Systematic Approach is not necessarily the one most easily defended.

The chief criticism of the systematic approach is lack of evidence for transfer of learning to new contexts, particularly contexts that require problem-solving skills. While problem solving is not precluded by Gagne's model, it is a higher-level skill that needs to be preceded by lower-level concept and skill development. Because problem solving may be postponed, some students may not reach this point, and, even more serious, might not acquire the deeper-level understandings and self-confidence needed to solve new problems. Response to these criticisms comes from the Minimalist Approach.

Minimalist Computing Instruction

The Minimalist Approach was developed through research conducted in the 1980s primarily with word processing software by John M. Carroll (1990, 1998) and publicized in the book The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill. This model is based on constructivist assumptions about learning and stems from attempts to deal with what Carroll identifies as the "paradox of learning": in order to learn, the student must interact with the system; but in order to interact with the system, the student must learn. This immediately raises the question of where to start. The minimalist answer is with the needs and interests of the student rather than the features of the software.

Several variations of constructivist assumptions could be identified, but across these several schools of thought about how learning occurs, some common assumptions can be identified (O'Connor, 1998). It should be noticed that, like information-processing theory, constructivism is a theory (some would say a philosophy) about learning, not a theory about teaching. But, assumptions about how learning takes place necessarily affect decisions about how to support this process through instruction (Duffy, Lowyck, Jonassen, & Welsh, 1993; Duffy & Jonassen, 1992; Carroll & Van der Meij, 1998; Mirel, 1998; and Redish, 1998). Table l lists constructivist assumptions about learning that are supportive of the Minimalist Approach to teaching computing skills.

Constructivist assumptions about learning are beginning to appear in the business education literature when the broad implications of technology on instruction are examined. Taylor and Jeffers (1994) have participated in this discussion by asking how technology is likely to change the teacher's role. They concluded that teachers would be in more learner-centered classrooms, encourage more active learning in teacher-guided environments, and generally work in more collaborative and cooperative teaching settings. They do not specifically address the question of what such settings might imply for teaching technology use itself.

Research, generally in industry settings, has repeatedly demonstrated the effectiveness of minimalist instruction (Carroll, 1990, 1998; Van der Meij, 1992; and Ramsey & Oatley, 1992). Van der Meij and Carroll (1998) have identified four principles for designing minimalist computing instructional settings; these are listed in Table 1.

Prominent in the Minimalist Approach, in contrast to the Systematic Approach, is concern for the complexity of software use. This means more attention is given to the social context of software learning and its subsequent use. Considerably more regard is given to the learner's perspective in the context of a work environment. Learning through error recovery is particularly prominent, given the great likelihood of computing problems and the usefulness of errors or "breakdowns" for calling learner's atttention to essential software rules and features. Such “breakdowns” are the kinds of experiences that eventually lead to greater understanding of both the technology and the work-related task at hand (Winograd & Flores, 1987; Schank & Cleary, 1995). Such "breakdowns" are instructional experiences to be noticed and taken advantage of by teachers and learners, not experiences to avoid by trying to ensure consistent student success.

A counterpart to the information-processing learning model for the Systematic Approach is the diagram in Figure 2 of software use within a work setting which could be said to depict a Minimalist setting (Flister, 1998). Notice the inclusion of several aspects of the social or organizational environment in contrast to the internal mental structure of the learner which characterized the information-processing model. The task to be completed using computing software by the worker/learner is influenced by several external social and techical forces. Both the organization in which the worker/learner participates and the teacher (possibly supervisor on the job) affect the interpretation of what the task is—what and how the task is to be accomplished. How the task is carried out using software is also affected by the software developers—those who designed the program itself, its functionality, those who designed the interface that the user sees to interact with the program, and those who designed the documentation (perhaps also those who designed the textbooks) that aid the worker/learner in getting the software to operate. No attempt is made in this model to describe the internal mental operations of the worker/learner.

While the social setting is complex, the learning prescriptions are quite lean--hence the term "minimalist." A hallmark of the Minimalist Approach as Carroll first developed it was the "Minimalist Manual." This was a collection of instructional sets of cards where verbiage was kept to a minimum and instruction could be pursued in any order without a prescribed sequence.

While systematic or guided teaching practices appear to dominate in actual classroom settings, there is recognition that minimalist instruction or independent practice is important for developing software problem-solving skills. Minimalist instruction is premised upon constructivist assumptions about learning, and "the search for transfer is a central mission in designing instruction for constructivist learning" (Mayer, 1999, p. 147). The main purpose of computer-related instruction is to support transfer of learning to new settings where different software and different types of problems may be encountered. The intent of this project was to see in what ways programs in exemplary institutions shared these conclusions.

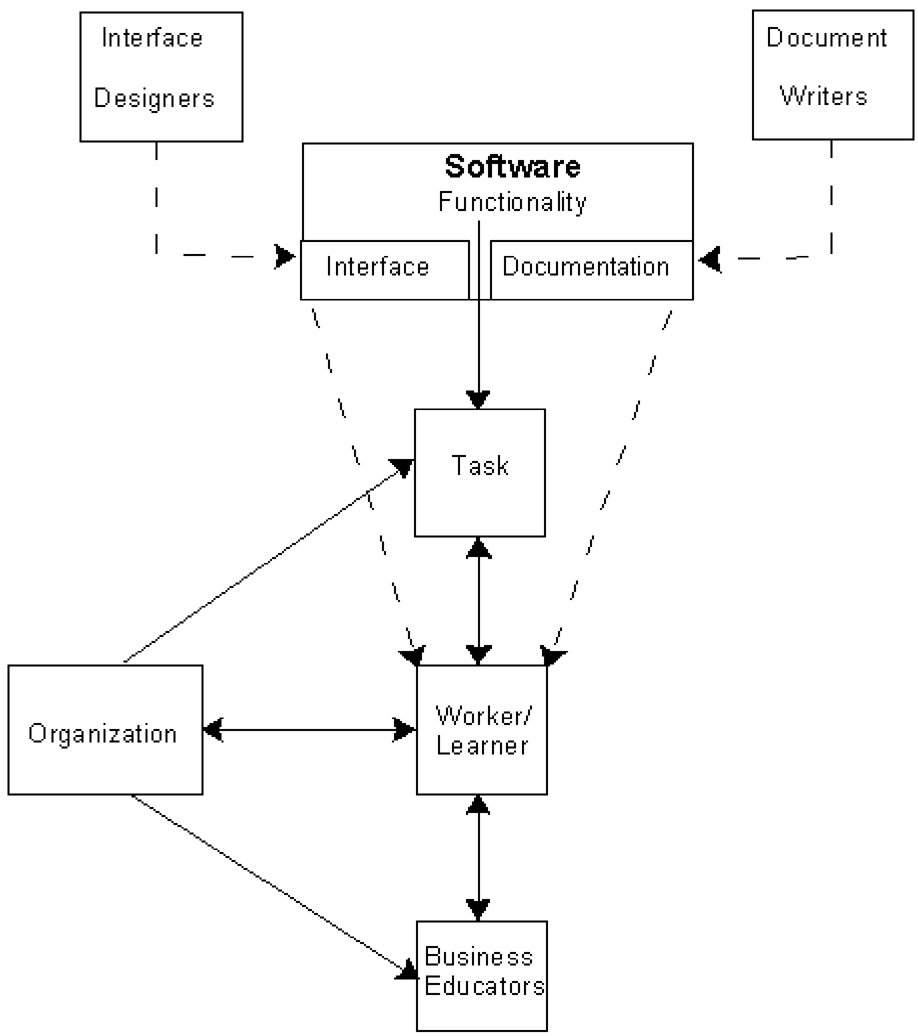

Figure 2: Software learning context (Adapted from Flister, 1998)

Chapter III – Procedures

The school and programs selected for participation in the study are described below. The critical incident interview protocol is described along with the results of pilot testing these procedures. The numbers of interviews completed are summarized according to type of school and interviewee. Finally, the analysis procedures for the interview data are described.

Site Selection

During the initial phase of the study, schools (secondary or postsecondary) were identified with internship or cooperative-education programs linked to business information technology instruction. School nominations were sought from persons in leadership positions in states identified in previous research as having reputations for strong vocational programs (Moss, Lambrecht, Jensrud, & Finch, 1994). These nominations were secured by letter and phone calls to state leadership staff who assisted in identifying schools considered the strongest in their states. Phone calls to state leadership staff and program department heads were more effective than mailings in obtaining specific leads to potential school participants.

"Strong" or "successful" schools were judged subjectively on the basis of their length of time in operation, the success of their students in obtaining employment, and general consideration that innovative practices in these schools had frequently been a model to others. Several of the schools had won awards for their exemplary programs or evidence of noteworthy growth and development within their regions. These nominations allowed for a selection of schools from urban, suburban, and rural settings at both levels of instruction, secondary and postsecondary. Five postsecondary technical colleges and one high school agreed to participate in this project. Participation meant providing time for on-site interviews with students and teachers and identification of local businesses that had employed student interns from their programs. The following summarizes the locations of these six schools, whose identities are confidential as a condition of their participation:

- High School – Suburban
- Technical College – Small Urban
- Technical College – Small Urban
- Technical College – Urban
- Technical College – Urban
- Technical College – Rural

All of the participating schools have solid reputations for having exemplary programs in the areas of Office Technology, Administrative Support, or Office Systems. The programs have consistently high employment placement rates for their students, though specific follow-up data were not available for the Office Technology programs. In all of the schools the Office Technology programs had strong linkages with their business communities through advisory committees and student internship programs.

Office Technology programs prepare students for support positions in businesses that have a variety of job titles. These include secretary, administrative assistant, clerical assistant, word processing operator/specialist, medical secretary, legal secretary, receptionist, and office computer specialist. The specific job titles vary widely from firm to firm and frequently have a hierarchy of levels. All require a high level of word processing expertise and associated written communication skills. In addition to word processing skills, all programs asked students to become competent in using the complete spectrum of office computing applications: spreadsheets, database, business graphics, and telecommunications, including e-mail and Internet use.

The following Table 2 summarizes the locations of the programs included in the study and the number of persons interviewed at each site. In all instances the program was identified as Office Technology, Administrative Support, or Office Systems. This delimitation was necessary because of time constraints and because other technology-intensive programs, such as Computer Information Systems or Network Systems Engineer, are quite different from the Office Technology program in their teaching practices and the types of students served. Some of these program differences will be described more fully when the findings are discussed.

While Office Technology programs cannot be considered representative of all instructional programs in which computers are prominent, office users of computers are noticeably dominant in the work place. It has been estimated that a majority of computer users are clerical workers (seven to ten million) and 80 percent of them are women. Because of the large number of clerical and other support workers in the US workforce, in fact, more women use computers than do men (Machung, 1988). In this study, most of the students interviewed were women (two were men), and all but two of the teachers were women. It also happened that only one of the employers interviewed was a man.

Table 2: Participating Schools by Location and Number of Interviewees

School Type	Student	Teacher	Employer	Total
High School – Suburban	3	1	1	5
Technical College – Small Urban	3	4	4	11
Technical College – Small Urban	5	3	2	10
Technical College – Urban	0	3	0	3
Technical College – Urban	5	7	2	14
Technical College – Rural	3	0	2	5
Total	19	18	11	48

While an attempt was made to talk with students, teachers, and employers at each school site, this was not possible. At one site, only one visit was scheduled and the entire morning was spent talking with teachers. Because school had ended for the year, students were not available. It was not possible to schedule a later return visit to talk with either students or employers, as desirable as this would have been. In a second school, three site visits were made; teachers and other school staff were helpful in allowing class observations and making conversations possible with several students. Employers were recommended for interviews. However, because the teachers were all teaching an overload of classes, it was not possible to talk with them outside of their classes. While much was learned at this site, interviews with the teachers were not captured either on tape or in shorthand. In the other four school instances, all three groups of interviewees were available.

Interview Guidelines

The following is the general program information that was sought from teachers, students, and employers. Some of this information was gained through introductory interview conversations; specific questions about teaching practices were answered through critical incident interviews.

> *Teachers*: rationale for business technology program organization with regard to outcomes anticipated for students, student prerequisites for program entry, sequencing of courses, instructional laboratory arrangements, choice of instructional materials and teaching strategies, student assessment practices, program integration with internship or cooperative settings, and program follow-up evaluation.
>
> *Students:* initial business technology program expectations, prior preparation for business program coursework, program experiences, perceived inter-relationship between business content coursework and technology-related instruction, perceived relationship between in-school and work-site expectations and learning opportunities, and personal judgment about learning outcomes.
>
> *Employer/Supervisors*: student preparation for available technology-related job assignments, and relationship between in-school business technology program and work requirements with particular focus on those aspects of instruction best provided by either in-school or work-site experiences.

The critical incident technique (CIT), a qualitative approach, employs the interview method to obtain "an in-depth analytical description of an intact cultural scene" (Borg & Gall, 1989, p.387). According to Gay and Diehl (1992), behavior occurs in a context, and an accurate understanding of the behavior requires understanding the context in which it occurs. For example, the culture of an organization can have a direct influence on the behavior of the employees. Therefore, having an understanding of that culture can lead to a better understanding of the employees' behavior. As a result, qualitative methodology is an appropriate method for understanding real-world job settings. The critical incident approach, in particular, is an appropriate tool that can be used to analyze jobs in the social context in which they occur. It is not, however, an adequate substitute for a more time-intensive, ethnographic case study.

John Flanagan (1954) developed the critical incident technique, and his now famous article, "The Critical Incident Technique," is considered by the Society of Industrial and Organizational Psychology to be the most frequently cited article in the field of industrial/organizational psychology (American Institutes for Research, 1998). The CIT is an outgrowth of studies done in the Aviation Psychology Program of the United States Army Air Force during World War II. Flanagan was faced with the problems of improving military flight training, bombing missions effectiveness, and combat leadership. On a large scale, he systematically asked trainees and veterans to describe exactly what they had done successfully and unsuccessfully with respect to a designated activity. Later, Flanagan formalized this process and defined it as a method of identifying critical job requirements. The process involves collecting factual stories or episodes about behaviors that are crucial in making the difference between doing a job effectively and ineffectively (Flanagan, 1954; Zemke & Kramlinger, 1982). The American Institute for Research (1998) defines CIT as a "set of procedures for systematically identifying behaviors that contribute to the success or failure of individuals or organizations in specific situations." According to Zemke and Kramlinger (1982), the critical incident technique is not an appropriate job analysis tool for every job. It is appropriate for jobs that have a flexible or indefinable number of correct ways to behave. This is probably an accurate description of teaching.

The structure of CIT involves: (1) developing plans and specifications for collecting factual incidents (e.g. Determine from whom the information is to be collected. Determine method of collections. Develop instructions about the collection.), (2) collecting episodes/critical incidents from knowledgeable individuals, (3) identifying themes in the critical incidents, (4) sorting the incidents into proposed content categories, and (5) interpreting and reporting. The data can be collected from observations or from viable self-reports, e.g. interviews. The classification and analysis of the critical incidents are the most difficult steps because the interpretations are more subjective than objective (Di Salvo, Nikkel & Monroe, 1989).

The interview procedures to be used in the study were approved by the Committee on the Use of Human Subjects in Research at the University of Minnesota. The approval letter from the Committee is in the Appendix, page 102. All students, teachers, and employers who participated were asked to sign the approved consent form (see Appendix, page 103) indicating that they had been informed of the purposes of the project and were participating willingly. Parental consent was obtained for the high school students.

Given the range of information that was being sought at each site, at least two site observations and interviews with teachers, students and employers were planned. Interview protocols were developed that asked teachers, students, and employers to focus on examples of critical learning experiences or work experiences as a way to explore important program or job setting aspects. These interview guidelines were tested in two Minnesota postsecondary schools to see if they were effective in eliciting rich program and classroom descriptions.

During pilot testing it was learned that faculty and students found it easier to talk about critical learning experiences if they were asked to think of a software feature that was challenging for them to teach/learn rather than asking about effective learning experiences in general. When a challenging topic or software feature was identified, it was then possible for interviewees to talk about the teaching practices that were effective in dealing with this problem. Employers were able to talk either about critical incidents related to learning opportunities for students on the job or special projects that student/interns were asked to complete.

Figures 3, 4 and 5 summarize the guiding critical incident interview questions that were asked of teachers, students, and employers.

Figure 3: Interview Protocol - Teacher

Introductory question: I want you to think of an occasion when your teaching had a noticeable impact on your students. This might be an occasion when students finally caught on to some difficult concept. Please describe to me the key elements of this teaching experience with enough detail so that they can be clearly understood by others. I will be asking some questions to assist you in telling your story.

1. Can you give a brief overview of the experience?
   (Setting, circumstances, hardware, software, content, methods, materials, timing)

2. Had you tried to teach the concept before? IF YES, how was this lesson different?

3. What background do you think students need to understand this concept?

   3a. Where do you think students learn this (background needed to understand concept)?

4. What makes this concept difficult for students?

5. How do you balance the need to teach software operation and business concepts?

6. How do you think students would use this concept/skill in a work setting?

7. How do you decide when students have mastered this concept/skill?

8. If I wanted to provide a similar learning experience for students, what would I need to know?
   (What are the key elements that made this teaching experience effective?)

Figure 4: Interview Protocol - Student

Introductory Question: I want you to think of an occasion when you were in a computing class and the activities had a noticeable impact on your learning and understanding. This might be an occasion when you finally caught on to a concept that you were having a hard time understanding. Please describe to me the key elements of this lesson with enough detail so that they can be clearly understood by others. I will be asking some questions to assist you in telling your story.

1. Can you give a brief overview of the experience?
   (Setting, circumstances, hardware, software, content, methods, materials, timing, outcome)

2. Had you tried to learn the concept before? IF YES, how was this lesson different?

3. What background do you think you needed to understand this concept?

   3a. How do you think you got this background?

4. What makes this concept particularly challenging?

5. How do you think you would use this concept or skill on the job?

6. How could you tell when you were doing it right or could understand the problem?

7. Do you think the standards used in class would be the same as an employer would use?

8. If I wanted to provide a similar learning experience for students, what would I need to know?
   (What are the key elements that made this learning experience effective?)

Figure 5: Interview Protocol – Employer

Introductory Question: I want you to think of an occasion when you had a student intern or newly hired graduate doing computer work. Think of an occasion when the student/new employee did an exceptional job OR had a computer problem they could not solve quickly on their own. Please describe to me the key elements of this experience with enough detail so that they can be clearly understood by others. I will be asking some questions to assist you in telling your story.

1. Can you give a brief overview of the experience?
   (Setting, circumstances, hardware, software, project, materials, timing, outcome)

2. What factors lead to success/caused the difficulty?

3. What background do you think affected the student's/new employee's performance on this job in your business?

4. Where do you think students/new employees should get this background?

5. What resources did/should have the student/new employee use/need to get this job done?

6. How did you decide when this project had been done well?

7. If I wanted to help students/new employees deal with work assignments/problems like this, what would I need to know?
   (What are the key elements of the work experience that made this intern/new hire effective?)

Interview Analysis

Interviews were carried out with teachers, students, and employers by the researcher. When possible, these interviews were recorded with the permission of the interviewee. When recording was not possible because of the spontaneity to engage in the interview, notes were taken in shorthand by the interviewer. Even when an interview was being recorded, shorthand notes were also kept. While the entire interview was not recorded when only shorthand was used, verbatim segments of responses were possible throughout the interview. This allowed the researcher to continue to carry on a conversation with the interviewee.

Interview transcripts were transcribed by the researcher with the assistance of two graduate students. The researcher transcribed all of the shorthand notes. One graduate assistant transcribed all taped interviews. A second graduate assistant read all of the interviews and transferred them into segments representing complete thoughts on a single question or topic. This allowed three people to read all the interviews and contribute to the development of interview themes, within the context of the original study questions. All transcribed interviews were broken into coded segments representing complete thought statements. After coding, all of the interview segments were transferred from word processing format into a spreadsheet for further analysis.

Transcripts of the un-coded interviews were mailed to the participating teachers and employers for their review. Students were not asked to review their transcripts because they were generally not easily contacted after the end of the school year. While several of the employers and teachers acknowledged receipt of the interview transcripts, none asked to make changes in what had been recorded.

The researcher did the initial coding of all interviews independently. One graduate student reviewed all coded interviews. A second graduate assistant reviewed an early portion of the coded interviews. When there was not agreement on a code, the decision was discussed in order to achieve consensus. If an existing code was not appropriate, additional themes were added when they were considered to be better descriptors of the interview segment. The transcribed interviews were coded according to themes that coincided with the major areas of questions. These seven broad areas of the themes are listed below. A complete listing of 105 specific themes within these broad categories is available in the Appendix, page 105.

Main Interview Themes

A. Needs/background for successful course participation or internship success
B. Student and teacher program expectations in relation to employment
C. Student learning problems in class or on the job
D. Program issues
E. Competencies best learned in school
F. Competencies best learned on the job
G. Effective teaching practices

The decision was made to create separate specific themes for the employer interviews even when students or teachers had mentioned the same idea. This would keep visible the circumstance that employers were responding to different questions from those raised with teachers and students. For example, though employers, teachers, and students all talked about teamwork and about written communication skills, employers were taking about these capabilities from the perspective of where these skills were best learned. Students and teachers talked about teamwork as a learning activity and about written communication as something they needed to know in order to engage in certain software-using tasks. While all three groups would probably agree that working in teams and having good written communication skills were important for employment success, and could and should be part of school-based learning activities, it seemed that greater clarity would be retained by not merging the themes. Students and teachers had a more similar frame of reference in talking about school experiences, and the same theme code was used when they talked about similar events.

In all interviews, the text that represented each theme was marked as a block. In order to permit counting and sorting of the themes, all interview text was transferred into a spreadsheet and the themes represented by numbers. These numbers precede each theme in the Appendix listing, page 105. By placing each block of text and its identifying number into a spreadsheet, pivot tables could be created to allow summary counting of each theme. Pivot tables made it possible to tell how many different themes were present—mentioned at least once—in a single interview. Pivot tables also allowed a tally of the total number of times a theme occurred within each group of interviewees. Finally, spreadsheet format allowed similarly coded themes to be sorted together, thus allowing examination of text from all interviews representing a single theme.

A theme may have occurred several times within an interview, but, for purposes of analysis, a theme was counted only once per interview. If a group of students or teachers were interviewed as a group, this was counted as one interview. Table 2 on page 29 represents the number of interviews used for this tally, except for the teachers in the one urban technical college where only teachers were interviewed. These three teachers were interviewed as a group over the space of four hours, and this event was used in the analysis as one interview. Because only teachers were interviewed at one school site, rather than also including students and employers, as was desired, it is important to notice that three persons were included. Because these three teachers were merged into one interview, the number of interviews from teachers reported in the succeeding tables is 16 rather an 18, as was shown in Table 2.

Chapter IV: Findings

The following section presents the summary of themes from the 46 interviews with teachers, students, and employers associated with Office Technology/Administrative Support/Office Systems programs in six schools. It was considered appropriate to consider all schools together rather than separate the one high school from the postsecondary sites. The high school was a regional vocational center to which students were bused from surrounding high schools. It shared these common characteristics with the postsecondary schools:

- All students had an explicit employment objective for their enrollment.
- Technology employment preparation included the same software applications.
- Instructional materials included the same textbooks.
- Instructional practices were very similar in that student self-pacing was a key feature of all programs.
- Internship opportunities were available within the program.

The numerical summary of the occurrence of the themes within the interviews will be presented first. Following this summary, the information from these themes will be used in Chapter V to address the main questions raised in this study.

Summary of Themes from Interviews

By placing each interview into a spreadsheet, it was possible through the use of pivot tables to tally the number of times each interview code appeared in each interview. This tally could, in turn, be used to combine all of the interviews for each group of interviewees, students, teachers, and employers, and sum the total number of times each interview theme was mentioned. A specific theme was counted as occurring once per interview, even though an interview might contain several statements coded for a single theme. Within the seven broad theme categories, 105 different themes were coded. Tables 3 through 5 which follow summarize these final tallies of the number of times a particular interview theme was mentioned, in rank order, by each of the three groups: students, teachers, and employers

Table 3 contains the rank ordering of themes identified in interviews with students. Out of the total 105 different themes, 64 themes appeared at least once in interviews with students. Table 3 contains the 25 themes that appeared in at least three different interviews. The decision to include only themes found in at least three interviews was made to reduce Table 3 to a manageable size. The table containing all 65 themes from students is in the Appendix on page 109. In Table 3, the letter label preceding each theme identifies the broad category of which a specific theme is a part.

Table 3: Total Times a Theme was Mentioned
Sorted by Students

Theme Category	Theme	Students N =19	Teachers N =16	Employers N=11
G	Independent practice/self-pacing	17	11	1
C	Specific software features	15	11	0
G	Student focus/student-teacher contact	15	15	2
B	Business requirements	13	10	0
B	Computer requirements	13	5	1
G	Instructional materials	13	12	1
G	Other students	9	5	0
A	Keyboarding	8	8	0
G	Feedback/constant evaluation	8	8	0
G	Open-ended problems/exercises	8	8	0
G	Employer/work involvement	7	2	3
G	Simulations	7	3	0
G	Structure/schedule for students	7	10	0
G	Student groups/teams	7	5	1
G	Business focus/"real-world" focus	6	11	1
G	Computer in several courses	6	3	1
G	Teacher demonstration	6	8	0
B	Mailability standards	5	2	0
A	Prior experience	5	1	0
A	Accounting skills	4	2	0
C	Computer phobia	4	5	3
A	Software skills	3	9	0
B	Newspaper ads	3	1	0
G	School-wide projects	3	1	0
F	Specific work procedures	3	0	7

Table 4 shows the frequency of themes sorted by their rank order from interviews with teachers. Of the 105 specific themes from all interviews, teachers mentioned 70 different themes. Of these, at least three teachers mentioned the following 36 themes. The decision to include only themes found in at least three interviews was made to reduce Table 4 to a manageable size. The table containing all 70 themes from teachers is in the Appendix on page 113. In Table 4, the letter label preceding each theme identifies the broad category of which the specific theme is a part.

Table 4: Total Times a Theme was Mentioned
Sorted by Teachers

Themes Category	**Theme**	**Teachers** N=16	**Employers** N=11	**Students** N=19
G	Student focus/student-teacher contact	15	2	15
G	Instructional materials	12	1	13
G	Independent practice/self-pacing	11	1	17
C	Specific software features	11	0	15
G	Business focus/"real-world" focus	11	1	6
B	Business requirements	10	0	13
G	Structure/schedule for students	10	0	7
A	Software skills	9	0	3
G	Course diversification (different type of course for different student audience)	9	1	1
G	Course accommodation (different types of students in same course)	8	0	0
A	Keyboarding	8	0	8
G	Feedback/constant evaluation	8	0	8
G	Open-ended problems/ exercises	8	0	8
G	Teacher demonstration	8	0	6
G	Faculty cohesion	6	0	0

Table 4: Total Times that a Theme was Mentioned
Sorted by Teachers
(Continued)

Themes Category	**Theme**	**Teachers** N=16	**Employers** N=11	**Students** N=19
D	Academic credit	5	0	0
B	Computer requirements	5	1	13
C	Computer phobia	5	3	4
G	Other students	5	0	9
G	Performance exams	5	0	0
G	Student groups/teams	5	1	7
G	Whole group discussion	5	1	0
D	Academic “turf”	4	0	2
A	Communication/written English	4	0	2
G	Support--tutorial	4	1	2
G	Concept/theory exams	4	0	0
G	Oral presentations	4	1	2
D	Program identity	4	1	0
E	Attitude/work ethic	3	8	2
G	Computer in several courses	3	1	6
A	Motivation/reason to be here	3	0	1
A	No business content	3	0	1
G	No concept/theory exams	3	0	0
C	Personal situation	3	2	2
G	Simulations	3	0	7
G	Support--technical	3	0	1

Table 5 shows the frequency of themes sorted by their rank order from interviews with employers. Of the 105 specific themes from all the interviews, employers mentioned 47 different themes. Of these, at least two employers mentioned the following 25 themes. The decision to include only those themes mentioned by at least two employers was made to reduce Table 5 to manageable size. The table containing all 47 themes from employers is in the Appendix on page 117. In Table 5, the letter label preceding each theme identifies the broad category of which the specific theme is a part.

Table 5: Total Times a Theme was Mentioned
Sorted by Employers

Theme Category	**Theme**	**Employers** N=11	**Teachers** N=16	**Students** N=19
E	Computer skills	10	0	2
E	Attitude/work ethic	8	3	2
F	How to get assistance	7	0	2
F	Interpreting expectations/priorities	7	0	1
F	Specific software	7	1	2
F	Specific work procedures	7	0	3
E	Communications/ written English	6	1	0
E	Keyboarding	6	0	0
F	Dependability	6	0	1
F	Business environment/ motives/ profit	5	0	0
E	Telephone/voice mail	4	0	1
F	Accepting work assignments	4	0	0
F	Attention to detail/ task orientation	4	0	1
F	Initiative	4	0	0
F	Specific computer systems	4	0	2

**Table 5: Total Times a Theme was Mentioned
Sorted by Employers
(Continued)**

Theme Category	**Theme**	**Employers** N=11	**Teachers** N=16	**Students** N=19
C	Computer phobia	3	5	4
G	Employer/work involvement	3	2	7
F	Teamwork	3	0	2
A	Communication/written English	3	1	0
C	Personal situations	2	3	2
G	Student focus/student-teacher contact	2	15	15
E	Copying/ faxing	2	0	0
E	Mailing procedures	2	0	1
F	Confidentiality	2	0	0
F	Telephone/voice mail	2	0	0

The discussion in the next chapter provides a fuller definition of the themes derived from the total of 46 interviews. These themes were used to provide answers to the main questions raised in this study. They also were the basis for the guidelines presented for developing employment-related office technology skills.

Chapter V: Discussion and Recommendations for Research

The following sections use the themes contained in 46 interviews from students, teachers, and employers to answer the main questions raised in this study. These themes are first discussed within the context of the major questions themselves. Next, the finding that systematic instructional practices were conspicuously more prominent in Office Technology programs than were minimalist teaching practices is specifically addressed. Explanations are explored for why this was the case in programs recognized for their successful practices. The issue is raised about how to balance different types of program goals for learners at different stages of software expertise.

In the process of talking with both teachers and students, another issue was noticed that has an impact on teaching practices—the image of Office Technology programs within postsecondary institutions. This issue was not part of the original questions of the study. However, it is hard to avoid asking about the place of Office Technology or Administrative Support preparation in a US economy that struggles with both a shortage of information technology workers and sexist job categories with markedly different salary levels. Recommendations for continued research are directed toward these two issues: the appropriate balance of different instructional practices, and understanding how best to organize and present, or, perhaps, to label, instruction for technology-related office support positions.

Lastly, guidelines are presented in Chapter VI for programs that are directed toward developing employment-related office technology skills. These guidelines are based on the practices observed to be working successfully in exemplary programs.

Discussion

The following questions provided direction for the critical incident interviews and are used below to organize the discussion of the themes found in these interviews:

1. What is the content of career-preparation programs whose purpose is to prepare business students for technology use in employment settings?
   a) What representative business technologies are being taught?
   b) What representative business applications illustrate technology use?
   c) What assumptions are made about student backgrounds and prior subject-matter understandings that affect the content chosen and its sequencing?

2. What instructional strategies do career-preparation programs incorporate to support learning of applications software for business employment settings?
   a) What assumptions about learning are apparent?
   b) What are the sources of instructional materials and work-related problems for learning technology?

c) How do assumptions about prior student background affect the use of structured, direct teaching practices in contrast to minimalism or constructivist practices for teaching software use?
d) What types of instructional practices do students use to develop independent problem-solving capabilities?

3. What does it mean for teachers and students in career fields to balance and integrate
   (a) how the technology operates,
   (b) business concepts being applied, and
   (c) expectations of a given work setting.

4. Which aspects of business information technology use do teachers, students and employers perceive as better learned in employment settings in contrast to in-school experiences?

1. Content of Career-Preparation Programs

The general content of Office Technology programs is distinctive from the content of other programs that also use technology and, most markedly, different from programs like Computer Information Systems in which technology use, programming, network administration, and user support are the primary goals of the program. Some implications of these differences will be discussed further when current issues about teaching technology are addressed.

Office Technology programs have three basic components: basic skills development, general business preparation, and office-specific preparation. The distinctive basic skills of the Office Technology area are keyboarding and written communication. While many general computing curricula give scant attention to teaching keyboarding—even though keyboarding skill markedly affects ease of computer use—the Office Technology area places high value on this skill and considers it prerequisite to further computer use. One or two semesters might be devoted to developing high levels of keyboarding skills, skills that will later be reinforced in word processing instruction and transcription instruction. Written communications are also developed par excellence in Office Technology programs—this may be the most distinctive attribute of well-prepared administrative support staff. Through courses such as Business Communication, Machine Transcription (also called Legal or Medical Transcription in these specialized areas), and Office Procedures, students engage in intensive reviews of grammar, punctuation, spelling, and English style. Students and teachers alike recognized the importance of these areas for business employment, and all singled out such books as the Gregg Reference Manual as their "bible" for preparing high-quality, mailable business documents.

As a second type of curricula area, most Office Technology programs, particularly those that were two years in length, asked students to take general courses related to business. These include such courses as accounting, economics, management, and marketing. These requirements vary widely from school to school, especially as schools differentiate between Associate of Arts degrees for college transfer and Associate of Applied Arts degrees that might not apply to college transfer. Nevertheless, recognition was given to the relevance of general business content—outside of the Office Technology courses themselves.

The third major area of coursework concentration related specifically to Office Technology specialization areas. These courses focused on office procedures, sometimes the procedures specific to medical and legal offices. When students were in such specialties, they also took concentrated courses in the terminology of the area and other courses that would assist them in understanding the field, such as Physiology in the Medical Office Assistant curriculum and Business Law in the Legal Office Assistant program. Even if they were not in a specialty, Office Technology students and their teachers placed high value on their procedures-related courses. These courses allowed students to apply previously learned technology skills to actual office tasks, with the focus on the office task itself and not necessarily on how to use the technology.

When students were asked about key aspects of their programs and how they learned to deal with problems they were encountering (or thought they would encounter) in office settings, Office Procedures courses were more prominent in their discussion than were the specific software courses. These were courses where students learned about the receptionist role, how to handle telephones, how to manage information files—both electronic and paper—and how to deal with difficult interpersonal problems, such as confidentiality on the job and working with difficult customers. These were also classes in which formal team projects were prominent. Students knew that their employment success was largely dependent on demonstrating very intangible interpersonal qualities—as well as using software and creating impeccable, mailable office documents.

Here are some comments from students and teachers that support their holistic perspective about their programs and the strong link they saw between overall program expectations and workplace demands.

Teachers:

> JJL: What works well?
>
> It is the whole package that you offer.

> They need formatting skills. They need grammar skills. They need software skills. So you'd still need the pieces. You'd have to show the students how to put it together. They don't understand how to put it together. We did that in Office Procedures. Both in Medical Office Procedures and in [particular teacher's] Office Procedures. Where you're showing them and you're telling them and having them work on real work.

Students:

This is so cool, I'm so excited about it. I still talk about it. In Medical Office Procedures we were playing receptionist things—we were simulating an office a lot of the time, and I had three days of my internship, and there was an emergency where one lady had to go get a child, and another lady had to leave and pick up checks. I was the receptionist for an hour, and I did it.

Because of all the background that I had. I did not mess anything up. But because of the background we had here, and those couple of days that I had, I could do that basic part of that job.

S: My supervisor said that if I had to describe my job in one sentence it is to make his life easier, period. And that's what we've learned. Everything we've learned is based on making the doctor's life as easy as possible.

JJL: How have you learned that? What did you do here that conveyed that message?

S: How to. In Office Procedures we learned a lot of that. How to sort through his mail. Like what he's going to want to see right away, what you can take of for him without him even having to look at it. Writing letters for him.

Scheduling appointments for him. Triage. Deciding what, who needs to be worked in and who can wait. We learned a lot about "What's wrong with your son, and how long has he had this pain? And has the doctor seen him before?" You don't even put that through to the doctor. You just take care of it, or you put that through to the nurse, and let her take care of it. It's basically protecting him and doing all the stuff that he doesn't have to take the time to come and do.

JJL: [Discussing the problem of a database mail-merge project.] So how did you negotiate working together? What happens that lets you do that successfully?

S: How do you work together successfully? It's personalities, I think. Personalities allow you to work together. Especially if you have one person that's really outgoing, and another person who's shy, then the outgoing one is going to take over.

JJL: Do the classes do anything to help you deal with those issues?

S: Yes, there was one class [on integrated projects] that I was in, but it was four of us in a team. … And [another instructor] did that also in her Electronic Office class. You were assigned different roles for each of the projects.

S: Part of it was gathering information from other individuals, and then just the process of doing it. We sat together as a team and said, "OK, do you know how to do this? Do you remember this? And now what would we do?" And one would know and the other wouldn't. And when you go into an office to work, it's a team environment. They don't say that you can't ask any questions. That's so important to have that team environment. And I think that it builds confidence in other students, too.

2. Instructional Strategies

In order to learn what teaching strategies were working well, students and teachers were asked to think about areas of software learning and teaching that were challenging. These then were the focus of conversation to determine what teaching practices were particularly effective. The topics that presented learning/teaching problems will be summarized first. Then the key instructional practices will be described: student pacing, the instructional materials, student-teacher contact, structure through scheduling, and course diversification and accommodation.

Software topics causing problems. With a few exceptions, both students and teachers mentioned themes related to teaching practices more often than themes in any other category. This is to be expected from the nature of the critical incidents that students and teachers were asked to identify. The second most common theme mentioned by at least two-thirds of both students and teachers was that of problems with specific software features. This, again, is the result of the type of critical incidents interviewees were asked to think of—those related to difficulties in learning software. Students encountered a variety of problems when using computing software, and their common problems were of these general types, mentioned by both students and teachers:

- Using the operating system and file management procedures
- Doing mail-merges in word processing
- Setting up tables and columnar documents in word processing
- Managing headers and footers in word processing
- Controlling the appearance of charts in a spreadsheet
- Dealing with mathematical formulas in spreadsheets
- Creating queries in databases

These problems cannot be considered representative of the major types of problems students have when learning software, but they are ones that most business teachers will recognize as common problems. These are the recurring kinds of problems that make certain kinds of teaching practices necessary and effective.

In addition to software-related problems, the major focus of this study, students and teachers volunteered other types of learning problems. These were related to the two basic skills areas of written communication and mathematics. Students identified language skills, particularly the special terminology of the medical and legal areas, as challenges in their transcription courses. These courses asked students to apply their word processing skills. It was apparent that the software was not as much of a challenge as the content of the business documents being prepared, especially from voice dictation. The need for integrating the teaching of communication skills within the context of their application has the rich potential of being met within Office Technology programs because students see directly the importance of such skills to meet workplace expectations. This need has been reinforced by other reports that argue for integrating communication skills into workplace contexts (Boiarsky, 1997; Stasz & Brewer, 1999).

Here are comments from students who saw the relevance of their in-school communication instruction, in particular the use of English style reference manuals, on the job:

Students:

> I brought my Gregg Reference Manual in so I could have it there. I mean we run into the same problem that we did the first year. "Effect and Affect". Yesterday it came up again. The difference, and what you should do, and I had it there, I knew it, and it was nice. I said "It's here if you want to look at it". It's a nice reference to have on hand.

> JJL: Let's look at the work that you do because you're telling me interesting things about the Internet and software. When you do your work [in the classroom], …, when you get it back, do you think that the standards that you're asked to meet, the quality, are the same that you'd meet in an office?
>
> S: Yes. All of our documents have to be professional. They have to be without misspellings, they have to be proper punctuation, spacing, everything has to be up. He goes through it and if it's not [correct], he'll circle it, and show us. These words should have been corrected; these words should have had a break in it. [He tells us] if there're too many spaces here or if there're the wrong amount of spaces between the lines, if you're doing a form letter.

The area of mathematics as a learning problem was raised both by students using spreadsheets and by employers who were, in turn, asking students to use spreadsheets. The frequency of these problems, however, was less than the concern for written communication skills. Neither the theme of "math phobia" as a problem or the need for "math skills" in order to solve software problems appeared in the top-ranked themes in Tables 3-5. While this area was not prominent in this study, the challenge of assisting students to transfer basic mathematics skills from formal classrooms to actual practice—even to another classroom—is not a new one (Mathematical Sciences Education Board, 1998; Nunes, Schliemann, & Carraher, 1993; and Stasz & Brewer, 1999). The need to teach or to review mathematics skills within the context of learning software skills becomes particularly apparent when teaching spreadsheets or databases. Office Technology teachers who considered themselves experts in spreadsheet use did not consider themselves to be business math teachers. Giving time to teaching basic math concepts would reduce time devoted to software features.

In these excerpts, two teachers expressed the challenge of teaching math concepts in a database and a spreadsheet course:

Two Teachers:

JJL: Where would the formulas come from? Would they be given to the students, or would they have to understand them?

T: Yes. It might be both ways. It depends on the assignment. There's never just one way. A lot of the software for us, it's either given to the students, or we give it to them, as opposed to the Accounting students, where they have to figure it all out.

JJL: Why do you think it's a good idea to separate those two things?

T: Yes, because I'm not teaching Math.

T: Yes. That's right. And do you know how long it takes to explain to students...

JJL: Yes.

T: Here's your unit price. They wanted the updated price that was an increase of four percent. We just did this in Access. It takes forever to explain to these students why you have to multiply by 1.04 and not .04. And if they multiply by .04, then why they only got a few cents instead of this plus this. And I do go through and explain this. But if every time we have to use a formula, I had to go through and have to explain the whole mathematical process, I wouldn't get through half the stuff I do in a chapter.

I can tell where the problems are by the questions they ask. The program that gives them the most problems is Lotus. They have problems with math. Their numeric concepts are weak. There was a student in the first session who was having trouble, and I pulled out a $20 bill and talked about change. Dividing things in pieces. Multiplying to find taxes. Adding the tax back on. What would happen in the store? I have to do that on a regular basis in this class.

As will be discussed in relation to balancing the teaching of software features versus business concepts, in this case, business math concepts, the balance is generally shifted in favor of teaching the software features.

Student pacing. The teaching practice common in all of the office technology programs, and which both students and teachers identified as important to their success, was student self-pacing of computer task completion. All but two students mentioned this aspect of their program experiences, and 11 of the 16 teachers considered this an important aspect of their teaching practices.

Instructional materials. To allow self-paced student work, instructional materials were chosen which had the chief characteristic of being ones students could use independently outside of class and in computer labs. Sometimes these materials are accompanied by CDROM versions of the tutorials in the textbook. Both types of materials contained step-by-step instructions with colorful examples of what the computer screen should look like when procedures were done correctly. Pictures and diagrams were mentioned by students as being important to their understanding of what to do and whether their procedures worked.

Teachers regarded pictures and diagrams in textbooks as essential to minimize reading requirements and provide crucial feedback to students. Teachers valued textbooks that had clear, brief explanations. They wanted more than just button-pushing instructional materials, however. Teachers wanted clearly highlighted, succinct hints for using software, the kinds of tips that they themselves might present in class demonstrations. In some cases a written textbook was supplemented with CDROM or video materials to illustrate software features and provide small simulations of their use. Beyond the software features being learned, teachers thought that independent use of instructional materials was a necessary experience in and of itself—it was the way students would continue to learn in the future.

Here are excerpts illustrating the way several teachers felt about the instructional materials:

Four Teachers:

> Then it [the textbook] gives them some hands-on practice. Let's try to use the tree command for example, can you see the tree? And it has great screen shots within the book so whenever it's talking about a concept, it shows the student what would be happening on the screen. Then, it takes them through and has the student actually do the commands, and shows them what their screen should look like, and if their screen doesn't look like that, then the student knows that they should go back and find out why it didn't work like they thought it should.
>
> In DOS we always struggle with the very technical issues of there's a space here, there's a slash, this is a forward slash, not a backward slash. You know, the things that just frustrate the socks off of students. And we know that these are things that you just have to cope with. But it's not something that students like to have to deal with. So seeing many examples is what the book provides.

> And the reason that I like the [particular] book is that it gives very specific instructions. And the first time that you go through something, it tells you one hundred percent of how you do it. And as you go through the book it keeps repeating things. And when she brings it back the next time, she doesn't tell you precisely how to do it. She just tells you to do it. Knowing that if you have to, you can go back to page ten and see the steps of how to do it. And hopefully the further we get down the pike, we don't have to tell you all the steps and you just do it on your own.

> Learning from the CDs--it optimizes the learning for the student when they figure out when their good learning time is. Mornings. Evenings. Issues at home. This is freedom for this adaptability.

> This matches the way students are going to be learning in the workplace. They will be in front of a VCR. The instruction will be computer-based and individualized. They will be learning from videos. They need to know how to learn from these processes. They are actually learning how to learn. They are learning time scheduling. They are setting their own goals. They are learning from something other than a live person.

Here are some typical comments from students about their textbooks. This excerpt came from an interview with two students together.

Students:

> S: Now the book happens to be very, very good.
>
> S: In the book. The book is very good, it's very informative, it tells you, if you make a mistake if you get this result. It tells you go back and do this other thing. And maybe you did something like this. And then you go back to the book and...
>
> S: It's got the pictures. It also has the text, but also if you read everything in it before you do the exercises, you have an idea of what you're doing. You can always go back to it again.

While the step-by-step instruction was important, teachers also wanted culminating exercises or projects that asked students to apply the software concepts just presented. Students, as well, saw the open-ended exercises as being necessary to demonstrate their understanding. More advanced students actually looked upon the step-by-step instruction of beginning textbooks as creating a handicap for them if they did not try some actual, realistic problems independently.

Here are a few representative comments from teachers and students to illustrate their thoughts about open-ended problems:

Two Teachers:

> They have to understand the concepts of how to do it, and then after they have done a few where they have gone through step-by-step then the logical conclusion is OK here's a problem. Can you put something together now that you've done a couple that they've told you how to do?

> And the second year that I taught we had consolidated the labs. We taught in a lecture and discussion format, and then let them go into the labs and work on their own. And although we were available during certain times to help them, they might be working on projects when we weren't in attendance. And the students the second year learned far better and far faster than the first year.

Two Students:

> The book is very step-by-step. Basically you can read and follow along. I think it is good. Sometime you have a tendency to read and follow the instructions and not know what you are doing. The "on your own" problems are better. It makes me know what I have learned and what I need to go back and review.

> If something doesn't work, why doesn't it work? Is it the computer or is it you? It [projects] also allows you to use creativity with multi-media, HTML, PowerPoint, desktop publishing. It's just not cut and dry. I think that most people would benefit from classes that incorporate creativity. I think that starting out classes with little projects that you can incorporate into one big project.
>
> You could go to others who were more expert on one of the pieces of software, and that was good. Project oriented--teams on projects were good.

Student-teacher contact. The most frequently mentioned theme for the teachers and the third highest theme for students was the student focus within the programs or the personal contact between teachers and students. A chief benefit of student pacing of instruction through the use of very systematic instructional materials was that teachers were able to be flexible and respond to a wide variety of student needs. While self-pacing and the instructional materials are only two examples of this responsiveness to student needs, these were two key elements in freeing the teacher to work with individual students. It would be hard to overemphasize the importance to teachers of this contact with students.

Here are some illustrative comments from four different teacher interviews:

> By using this material, the instructors would spend less time talking about the software with the students and more time helping the students.

> The Help Desk person [student, within the class] keeps a log of the help they give. The book we use is complicated. If I see a tendency on the Help Desk sheet that students are having trouble with macros--176A10 task--I will give a demonstration for a group. They gather around any computer. Much of this is done on an individualized basis.

> Our area, OT, works well. We keep close track with our students. We take attendance. We contact them if they are not here. Teachers keep track of what the students are doing so they don't fall through the cracks. We always have an instructor here on duty. This is very, very important.

> From a group of teachers together:
>
> There is a lot of one-on-one instruction.
>
> If a person is at the end of the row, I pull them out to the side.
>
> They end up being individualized a lot.
>
> We have some wonderful teachers.
>
> It is the way business educators are trained.

Likewise, students were aware of the expertise of their teachers. Students were dependent upon teachers' availability to assist with problems of a wide variety of types—software problems primarily, but personal problems as well. Student/teacher contact was important as one of the sources of feedback and evaluation, a theme mentioned by eight students and teachers. Here are three students' comments about self-pacing:

> The pace and the teacher. Going at our own pace. The teacher works with us one-on-one most of the time because we're all in different parts. In different chapters, in different programs.

> You are learning on your own, but the teachers is there if you are stuck. They ask you to go through the book and find out how to reason it out. If you are stuck, the teachers are there to help you out.

> I have never been in a situation where there were people who were more supportive. I mean the support here has been wonderful. It's not so bad going back to school when you're twenty or thirty—it's no big deal, but once you pass forty, I thought I'm going to walk in and people were going to laugh at me or think that I am a teacher or something.

Structure through scheduling. While the ability to work at one's own pace to learn software was a characteristic of all the programs, so was the element of structure, or control over a minimum pace that needed to be maintained. No program was totally open-ended. While one school allowed students to begin computer-related coursework at the beginning of any week, there was a fixed amount of time available in which to complete course requirements. All other programs allowed students to work on exercises and projects at their own pace, going faster than the course schedule if the student wished.

Except for one school in which several Office Technology courses were taught in an open computer lab, all courses met regularly for teacher demonstrations and class discussions.

Further, tests were administered at regular intervals, and students were closely monitored, in all settings, to make sure their progress met key deadlines. Not all teachers agreed that concept/theory tests were an important part of teaching software use, but all agreed that performance tests were essential to the final evaluation of using applications software. This was true even though only five teachers identified performance tests as part of critical incidents from their teaching practices. All of the programs included performance testing among their assessment devices.

Course diversification and accommodation. The importance of teacher responsiveness to students' needs showed up not only in specific instances of one-to-one contact, but also in the way courses as a whole were organized. Half of the teachers interviewed talked about practices that have been labeled with the themes of "course diversification" and "course accommodation." The term "diversification" was used for a variety of course meeting-time arrangements and content differences according to different students' needs. The term "accommodation" has been used for including a variety of students within single courses in order to make particular courses viable—large enough enrollment to be offered. The major types of course arrangements observed in the six exemplary schools of this study were the following:

- Semester-length courses
- Open-lab courses
- Fixed-hour length courses
- Short, intensive workshops

Semester-length courses were part of degree or diploma programs, the type of courses that might also transfer to a four-year collegiate institution. All schools had such courses. This might be considered the "typical" course arrangement in public school settings for the teaching of technology skills.

Computer labs were present in all schools, but some used them more intensively than others for regular teaching engagements. One school offered introductory technology preparation solely within the context of a large open computer lab. Other schools that had "traditional" classroom-based courses also had arrangements for teaching computing classes in "flex labs." Students chose their times of attendance in these computing labs, after initial enrollment in a course that allowed weekly starting dates. Group presentations of materials or student group discussions were not easy to conduct, and students worked individually with teachers who were available at all times. Teachers felt strongly that the open computer lab or flex lab should be seen as a teaching lab for the Office Technology program, not just an open study area.

Even in schools that made extensive use of open computer labs or flex labs for teaching, not all courses in the Office Technology program were offered through these labs, not even all of the software-intensive courses. When technology program guidelines are presented, comments will be made about the type of courses that seemed to work best in the open lab.

Fixed-hour courses were generally offered for students who were not pursuing degrees. These students would not be transferring their coursework to a four-year school. In many instances, these students were obtaining employment preparation through the assistance of special state or federally funding programs, or through customized training arrangements for "displaced" workers. These courses were sometimes referred to as "vocational." The funding arrangements for these courses set the hour requirements that determined how long such courses could meet and what type of content was appropriate, thus the term "fixed-hour course."

When a course met the hour requirements of a "regular" semester-based course, the "vocational" fixed-hour course and the "degree" or "diploma" courses might be combined. This practice has been labeled "course accommodation." Combining groups of students increased enrollment for some courses such that the class could actually be held. This meant, however, that students with a wide variety of interests, goals, and prior experiences with business content and computers were all in one class. Such courses were successful because teachers were responsive to individual students through the self-pacing mechanism described above. Such courses were also successful because of the textbooks or other instructional materials chosen. These well-written and amply illustrated materials allowed independent student progress until assistance was available from the teacher, a tutorial assistant, or another student.

Lastly, several schools were responsive to community needs and specific business needs for *short, intensive computer workshops*. These courses are quite numerous because of public demand for computing skills. They are very much like industry-based computing training—of short duration to fit into the work schedules of people who cannot leave their work settings for the ten- to eighteen-week periods of the regular academic calendar. The courses are offered for four- or eight-hour time periods, and sometimes multiples of these half-day or whole-day events. The instructional is well structured through teacher whole-group presentation and very systematic.

Little student independence is allowed in intensive workshops, as the main objective is to present a comprehensive overview of particular software features. Tests are not generally given; student accomplishments are gauged by their immediate satisfaction with the experience and their judgment that they learned software features likely to be useful elsewhere. Teachers and students in these types of courses were not the primary audience of this study. Intensive computer workshops were similar to courses observed in this study in the structure of the text materials. Though the materials were necessarily briefer, the systematic model of computer instruction dominated.

3. Balancing and Integrating Software Operation, Business Content, and Employment Expectations

One of the assumptions of this project was that teachers and students would need to balance the teaching and learning of software use, business content, and office expectations. This balancing was done, essentially, by eliminating the need for prior business content knowledge. Introductory courses use business content as examples, but knowledge of business concepts is not necessary for understanding the software features. Implicit in textbook examples is the message that business uses the software in a particular way, but business standards for software use are not the focus of instruction either. Students have little opportunity in beginning software courses to bring their own problems to the learning of software. The systematic approach for teaching software, not conspicuously a minimalist approach, was inherent in the very comprehensive and well structured textbooks.

While introductory courses inevitably included either personal-use or business examples, it was possible for students to use the text materials and not know very much about business content per se. The focus was primarily on software features. The examples used were intended to be self-explanatory, familiar business topics that were presented in such procedural detail that knowledge about the business topic itself was not needed nor directly taught. For example, problems could include extensive analysis of stock market data without any discussion or elaboration about the stock market itself or particular stocks, unless a teacher would so choose. The purpose of the textbook problem using a business example was to illustrate the software feature—not to teach the business concept.

If students needed particular background in order to successfully engage in learning the new software concepts, that background was of two kinds: keyboarding skills and previously developed basic computing skills. Accounting skills were also recognized as related to using spreadsheets. The following excerpts from student and teacher interviews indicate the background they considered important to successful resolution of their software learning/teaching problems.

Three Teacher Interviews:

> The most important courses for students if they are going to be in our program (Office Technology) are Keyboarding Applications I and II. We read all of the work that they turn in these courses.

> Keyboarding, Word. Advanced WP is covered in both programs (WP and Word). Spreadsheet skills are needed, either Lotus or Excel.

> JJL, talking with a group of teachers: Do students need any business content to take the software courses that you're teaching?
>
> T: No.
> T: No.
> T: No, in Accounting they also have computerized spreadsheets, and then they can use the Lotus and the Excel. They have to take our course first, and then they can put all of their Accounting attention on Accounting.

Four Students:

> I have to have keyboarding; I can tell that I need to work on the skill. It is required. So I'm doing the first block.

> When you come into this class you had to know how to type. You had to have taken a class. That's the only requirement to get into this class. You have to have some kind of typing class--you have to know how to more or less touch type.

> All of the previous software that I had taken was a must. Then it was beneficial to be able to utilize that software that I had learned in that class.

> A lot of the Accounting work is what I needed here for the spreadsheets; I think spreadsheets are going to benefit me. I enjoy figuring out doing charts and percentages and having a visual there for meetings.

Office Technology teachers saw their primary role in beginning software courses as that of "software teachers." However, once beyond the beginning courses, teachers were keenly aware of office expectations and standards of document mailability that their students needed to master. However, if students did not take the more advanced, specialized courses, it would be possible to know how use a software package, but not have direct participation in either class or internship experiences that would place that software tool within an actual work context. Even without specific business contexts for instruction, students in the beginning classes were there because they knew that general desktop software was an unavoidable employment requirement in a variety of business jobs.

Students were keenly aware of the kinds of business employment requirements they needed to meet, particularly the general imperative to be able to use a computer for office applications. Teachers also mentioned the immediate business community and their advisory committees as confirming the importance of their course offerings. Students were aware the document "mailability" standards they were being asked to meet in their classes were also ones that businesses required. Several students suggested, however, that business standards varied; and they were being asked to meet high standards in their schoolwork. They learned about these business requirements from newspaper ads and their general contacts with friends and their communities.

Here are interview excerpts that illustrate both teachers' and students' expectations from program involvement and the rationale for the program decisions.

Three Teachers:

> I've seen too many students that I hired [when this teacher was employed in business] that couldn't function without an excessive amount of direction. And they're useless in a work setting. I don't care how fast they can type. I don't care if they know every punctuation rule, and I don't care if they know WordPerfect inside out and backwards, if I have to draw them pictures and tell them every single step to take, I might as well do it myself.

> And the reason that we did that [choose specific software] was because of the demand in [our city]. And when we talked back at the beginning about what we did and why we did it, it's because of the demand in [our city]. And [our city] was so WordPerfect and Lotus oriented. So we wanted to make sure that when they [students] left they knew how to operate whatever they needed to.

> We are very dependent on advisory committees. If we are switching materials, we get them involved. Temporary agencies. Big companies. These people are on the advisory committees.

Four Students:

> I think they [school and industry expectations] matched really well. All my instructors are very in tune with what companies will look for in business today. Computer skills. Team building skills. Computing. Communication skills.

> You have the deadlines [in school and at work]. And you have piles of work. I've worked as a secretary in [a particular business], and it gets pretty frustrating. Cause you're all by yourself--there's no group there. It's just a secretary, and I don't know it's just a lot more difficult there than it is here working by yourself. And then you have someone that you can go to. You can't go running to your boss every minute saying "Hey, I had a problem with this." So you have to figure it out on your own. It's kind of frustrating.

> Yeah [we are being asked to meet business standards]. But, I was talking to my aunt. She is a transcriptionist. And in our transcription class everything had to be almost perfect. And my aunt transcribes, and she says that she's not very good with grammar and things like that. And she says that it doesn't really...it's a big deal, but it's not that big of a deal.

> I know I need to know about computers to get a job. I didn't have the office skills I needed to get a job after the computer operator program. I'm doing factory work now that is very repetitious. I think office work can be that way if you have to do this (keying) all day. But, I think I will get a better job.

4. Sites for Learning about Information Technology

Employers were the group of interviewees who were asked most directly about the preferred learning site for employment-related office technology skills. Table 5 on page 42 gives the rank ordering of interview themes identifying either the school or work setting as the place to learn various skills or employment competencies. It is apparent in Table 5 that a mix of both “hard” software skills and “soft” interpersonal skills is to be gained in both learning sites if students are to enter an employment setting and be successful. Students need to come to office employment settings with a good base of computing skills—not an easily defined set of skills.

But, as nebulous as the “basic computing skill” category may be, students need to possess sufficient skill to learn new software on the job and to apply basic software functions to actual office tasks. The application of software skills to open-ended office tasks was begun in school. Both teachers and students recognized open-ended problems in the textbook as important for understanding software use. Even more, such problems were encountered in the advanced Office Procedures-types of courses. At the advanced level, software use was integrated with large projects, projects that were also likely to be encountered on the job. These school- and work-based projects included these kinds of tasks:

- responding to requests from customers and other employees,
- developing large-scale reports from a variety of information sources,
- preparing travel plans,
- verifying membership qualifications for corporate boards,
- creating corporate meeting agenda, plus the resulting minutes and reports, and
- keeping track of work schedules using software.

Projects like these reinforce the “basic skills” that businesses say they expect schools to develop—computing skills, attitude and the work ethic, written and oral communications, keyboarding, handling telephones and voice mail, using basic equipment such as copiers and fax machines, and handling mail services. Even as employers expect students to come to entry-level jobs with these skills, they also recognize that students will be learning specific work procedures once employed—and the challenge is to be responsive to these learning opportunities.

The ability to learn in an office setting appears to be dependent upon possessing the dispositional traits that are required for being part of a support staff. An important part of what students learn in Office Technology programs is the kind of work that they will actually be expected to do. Knowing this before entering an office setting prepares them to accept the kinds of assignments they are given—sometimes repetitive and not very challenging tasks that are important to the functioning of an office. Being able to accept these tasks and do them well is, in turn, dependent upon knowing how to interpret expectations and priorities and how to ask questions, and of whom. These later two traits—interpreting expectations and priorities and knowing how to ask questions, were frequently part of the traits or competencies that could only be learned on the job.

These interview excerpts from employers provide some insight into what these traits mean on the job.

- Interpreting expectations about how to do work, how to anticipate scheduling preferences:

> On another project, there was a deadline and we had to mail and generate letters that were personalized for very particular people. She [client] wanted the paper clip on the papers just so. She [student intern] worked well with that lady. My high school student did very well with her. They did very well. It got to be that they [documents] were coming off the printer--she pulled up a chair at the printer to sit down and I was stuffing the letters. I asked her if she would take over the printing. I didn't expect her to do the stuffing. She pulled up a chair. I was surprised that she was sitting. She was OK with that. You could do it faster if you were standing. I don't know how you create this ability. (JJL: to notice a better way to do something.)

> I can't assign more than just day-to-day things. For example, an intern that we have now in Human Resources, the individual strictly schedules interviews. The interviews are not just with the recruiters here, but with managers, with team members, and then there's some testing. So really having to work with all the different people who are coming to this intern and saying, I can't do it this day, or that day, and understanding how this has to work with everyone. And she has to be able to work with all these different levels of individuals who have all of these different priorities.

- Knowing where to go to get information; in this case, not the software reference manual!

> And if we see someone pick up the manual, it's kind of a joke that they don't really know what they're doing if they pick up the manual. For most of us, it is learned by doing. There are some tutorials that we run through on a CD-ROM that can do some nice things. But for the most part, here's the basics and we sit down and show them some things, and just let them sit down and say go play with it. We encourage play and make mistakes. Because you learn from your mistakes.

> The other thing that I want that is hard to teach in the classroom is the soft people skills. Saying good morning to a customer, the telephone, the attitude is a big one. The critical thinking. Knowing when this is beyond me. Knowing when to ask questions. Asking the right questions. When to take that next step of initiative and step forward.

These excerpts highlight independent learning traits. Knowing what information may be needed and how to get it are the kinds of traits that the Minimalist Approach to teaching, a discovery-learning approach, tries to cultivate. Teachers know that these traits are crucial to employment success. Yet they use, at least initially in their programs, systematic teaching practices that are not known to promote independence. It is time to ask why might this be the case.

Why so Little Evidence of Minimalist Approaches?

Previous research has shown transfer of learning of computing skills from training sites to employment settings to be a more likely outcome of minimalist instructional approaches than of systematic skill instruction (Carroll, 1990 and 1998). However, current observations in exemplary technical school settings have shown most programs to have primarily a systematic orientation in their instructional materials selection and instructional practices. Virtually all of the programs provided introductory and intermediate-level computer training using systematically oriented instructional approaches.

It is not clear whether minimalist recommendations for teaching software might be incorporated in some less conspicuous way within this textbook-dominated software instruction. Some instances were noticed where this was the case. On the other hand, it may be that minimalist learning practices should be postponed to advanced levels of instruction because certain student characteristics make early use of such constructivist teaching practices unacceptable (Ertmer & Newby, 1993; Jonassen, Mayes, & McAleese, 1993). There were several instances in the interviews with both teachers and students where the weaknesses of systematic, step-by-step instructional materials were evident. There were also instances of teaching practice that showed a teacher preference for features that might be considered "minimalist." Here are some examples of a movement away from a well-structured framework:

Comments critical of systematic materials:

Four Teachers:

> Unfortunately, many of the textbooks that I've worked with in that category tend to be so oversimplified. "Now push this key" and then show you a picture of what happened. That's OK, as long as it's being supported by "Why did I push that key", and what am I'm trying to accomplish, and not just that I go through the cookbook, and finish the project, and I have no idea of why I was doing any of this.

> I hate them when we use them in the classroom and we pretend that we're also learning how we're going to use this tool. Then they're too superficial and they're too…they don't give them enough, and because I pushed the buttons, then I'm all done. No, you're not done, you learned how to push all the buttons in the lab, but now you're going to come into the classroom and we're going to do something with it and you'd better remember what buttons you pushed. But not only that, but you'd better have figured out that I pushed this button because this is the result that I want to get from it.

> I have to correct errors in the books--the machine calculation book had some errors in the formulas. But, by and large, the books are clear. Students think they are doing the steps, but they may not understand why they did it.

> They take you through a whole chapter and give you steps, but if you have to quit in the middle of the chapter, you would have trouble. It is hard to stop in the middle. There are a lot of mistakes.

Two Students:

> [The textbook is] very step-by-step. Basically you can read and follow along. I think it is good. Sometimes you have a tendency to read and follow the instructions and not know what you are doing. The "on your own" problems are better. It makes me know what I have learned and what I need to go back and review.

> I just didn't think that it explained everything well enough. They expected you to either know what they were trying to say or you thought that...it was just really hard for you to understand. I didn't follow the directions. That was the book that I didn't like. I just didn't follow them exactly the right way, or I didn't understand.

Three Students:

- On step-by-step instruction that was computer- rather than textbook based:

> That was another thing about the [brand name materials]. At the end we had a simulation test, and it was all hands-on, and the computer says do this, and there are six ways to do it, and you choose one, and it will say you're wrong. This is the way we were looking for.

> They have pictures of the screens. It's just that they spend a lot of time on the bells and whistles, and not enough time on the...they will spend twenty minutes on fonts, when you can't set a tab. (Laughs). And if you knew the number of people in this school that can't set tabs, and I think it's more important in business to be able to set tabs than to add a new font to your computer.

> We are now working on applications of Access and Excel in a working environment. We don't use the [brand name materials] anymore. In those materials, projects were done step-by-step. If you have problems--you don't have much of a brain. It is nice not to have the steps given to us. Help references have been obtained from the WWW in this seminar.

Advanced students were more likely to notice the weaknesses of well-structured lessons. However, teachers and students generally liked the step-by-step materials, as was illustrated by earlier comments. Within these books, however, it may be possible to identify elements that had minimalist characteristics, and teachers singled out some of the following:

Two Teachers:

> The "Introductory Book" has good illustrations. The step-by-step is in smaller sections. Everything is labeled.

> In the Enhanced book, only the "Cases and Places" exercises are used.

Notice the preference for wanting the instructions in smaller sections—to prevent the problems of getting lost in the steps, having problems, and having to start over. In addition, the smaller sections were considered more useful for future reference—like a "minimalist manual" or regular reference manual that would allow directions for a specific purpose. Both teachers and students valued the illustrations because they provided crucial feedback about command use. It is also possible, as the second excerpt above illustrates, to skip the step-by-step instructions and proceed directly to open-ended problems.

The next excerpts illustrate the value that teachers saw in allowing students to try software features on their own—even when they had problems—because they learned more and increased their confidence.

Two Teachers:

> Once they get past the fear I'm going to break it, and try enough different kinds of software, that cumulative knowledge seems to kick them over the top into the confidence or the "I can do this" attitude. I have several different things that I can try; let's see which one of them will work. Eventually one will. And that's what learning computers is…I don't know a better way to do it. Try it. Those of us who see something new, and get excited and try it, and go back later to do the tutorial, come from that with "this is fun," and "I have enough knowledge that I can go back and probably figure out how it works," it seems easier for them. I don't know if you're right.

> And I think that the best learning the students do is when they fall flat on their faces and have to struggle through two hours of why doesn't this work? Having that instant teacher come over here and fix this problem--the first two years that I taught I was in a classroom setting and the students had the computers right in front of them and their lab time was with me right there with them. This was not really very helpful—it was a crutch.

The question that still needs to be raised is why does initial software instruction in exemplary schools not make more conspicuous and earlier application of minimalist teaching practices that other research has shown to be more effective than systematic instruction for developing problem-solving skills? Why are teaching practices not being used that other research and learning theory suggest would be more supportive than systematic methods for promoting transfer of software skills to different settings, particularly employment settings different from the training site?

Two factors seem be key in explaining the preference for using systematic instructional materials and practices for teaching technology skills: level of software expertise and characteristics of the social environment.

Levels of Software Expertise. Certain learner characteristics may create the need by beginners for what may appear to be systematic instruction for learning computing tools. As part of their critique of the goals of artificial intelligence research, Dreyfus and Dreyfus (1986) developed a five-stage model of skill acquisition that may be useful in understanding the effectiveness of office technology programs. Hackos and Stevens (1997) use this same model for developing standards for online communication and instruction, particularly instruction tightly facilitated by computing tools.

Students enrolled in formal education to learn computing skills may best be characterized as being at the lower stages of development. The implication, if one accepts this model, is that systematic instruction might be more appropriate to their needs. However, if one also accepts the assumption that few computer users progress beyond the first two stages, as Hackos and Stevens (1997) maintain, many instructional goals would remain unfulfilled. Awareness of these five levels may assist in interpreting the types of learners for whom different learning settings are more appropriate. These five stages are summarized in Figure 6.

Figure 6
Five Stages of Development for Computer Users

Novice computer users:
- have no previous experience
- experience concern about their ability to succeed
- don't want to learn, only accomplish a goal
- don't know how to respond to mistakes
- are vulnerable to confusion

Advanced beginner computer users:
- try tasks on their own
- have difficulty troubleshooting
- want information fast

Competent computer users:
- develop conceptual models
- troubleshoot problems on their own
- seek out expert user advice

Proficient computer users:
- want to understand the larger conceptual framework
- frustrated by oversimplified information
- correct previous poor task performance
- learn from the experience of others

Expert computers users
- are primary sources of knowledge and information
- continually look for better methods

Dreyfus & Dreyfus, 1986; Hackos & Stevens, 1997

With regard to the importance of the business context for learning software, Hackos and Stevens (1997) make these assumptions about Novices:

> In terms of job-related tasks, novices need tasks that they can successfully complete without considerable prior knowledge of the domain. If they have considerable domain knowledge or the domain is complex, they still want to be able to accomplish something quickly and then move into more complex tasks. (p. 33)

Advanced beginners have had sufficient experience in real situations, "…practical experience in concrete situations with meaningful elements…. (Dreyfus & Dreyfus, 1986, p. 22)", to begin to pick up situational elements that are similar to past experiences. "The vast majority of all users remain advanced beginners, performing the tasks they need and learning new tasks as the need arises, but never acquiring a more broad-based, conceptual understanding of the task environment" (Hackos & Stevens, 1997, p. 36).

Competent computer users are a smaller group of computer users whose actions involve seeking out and solving problems. They are likely to become involved in performing tasks that require more complex chains of actions and decisions. They appreciate the interrelationships among tasks to the extent that they can more easily get themselves out of trouble and correct errors. This description matches the expectations frequently expressed by employers for employees who are independent workers and learners and who can be said to exemplify such terms as "initiative" and "resourcefulness" (Stasz, McArthur, Lewis, & Ramsey, 1990; Smith, Jones, & Lane, 1997; Carnevale, Gainer, & Meltzer, 1990; Raizen, 1989; Secretary's Commission on Achieving Necessary Skills, 1991).

The Proficient computer users have sufficient experience to begin recognizing whole constellations of situational elements that, through experience, the user has associated with sets of actions that may lead to the desired outcome (Keller & Keller, 1996). Choosing among them is still a conscious act. "Proficient performers have moved beyond simple task performance and basic troubleshooting. These users need a conceptual framework to understand how the entire system works so that they can discover solutions to complex problems by extrapolating from the framework how things are supposed to or actually work" (Hackos & Stevens, 1997, p. 42). Providing instruction for learners at this level is not easy because of the complexity of the problems potentially available for solution.

The Expert computer user represents no more than 1 to 5 percent of users (Hackos & Stevens, 1997). This person is so experienced that action becomes non-conscious. They have a comprehensive, often intuitive and inarticulate knowledge of how a process or product functions. The expert commands a large set of situational element configurations that shape choices. These persons may or may not be good teachers, and for themselves, they rely mostly on self-instruction.

If most learners in formal educational settings are likely to be at Novice or Advanced Beginner stages of computing skill, this may exert pressure to provide systematic, very supportive instruction. Equally important as the level of learner skill in influencing instructional choices is the school environment. As social structures, schools create expectations between both students and teachers about what their roles are, what their goals ought to be, and what counts as success.

Contrasting Social Environments. The choice of systematic instructional approaches by office technology teachers is undoubtedly affected by their judgments about what works best for their students. What works best must also work within a given school environment. Schools are distinctive in that courses get completed within calendar-based frameworks; courses that must start and stop at certain points in time. Course content needs to be "packaged" to fit within this time constraint. Further, it must be possible, with large numbers of students, to account for what has been accomplished within the time-constraints of a single course or program. This means that the program outcomes cannot be ambiguous or given largely to the control of the student. Completing instruction directed toward pre-specified competencies or program outcomes, and doing so within a pre-specified time period, creates a school culture that may minimize the freedom that can be extended to novice software learners. Most beginning software students will very likely be at the “Novice” stage of learning, and what and how they learn needs to “fit” within the school setting.

All action, including schooling actions, is situated (Lave, 1988). Further, the situation of schooling is not the same as the situation of an employment site. Gee has developed the notion of Discourse as the situating context (Gee, Hull, & Lankshear, 1996; Gee, 1992; Gee, 1998). "A Discourse is composed of ways of talking, listening, reading, writing, acting, interacting, believing, valuing, and using tools and objects, in particular settings and at specific times, so as to display or to recognize a particular social identity (1996, p.10)". In this view, learning is situated and interpreted in the multiple Discourses to which the learner belongs—the school being a key one. The Discourse(s) represented by the employment goals also affect learning. These are the new materials, tools, and ways of thinking that the learner is confronting in employment-related programs. Early on, the Discourse of the new skill is so unfamiliar that the learner approaches the situation by way of other Discourse(s) in which the learner is expert, which in the case of school-based instruction may likely be the Discourse of *schooling.* The schooling Discourse is likely to imply characteristic ways of thinking. Some of these include

- the teacher as expert,
- teacher as pouring knowledge into the heads of students (Carroll, 1990),
- "what do I have to do to get a good grade?",
- fitting appropriately into the peer culture while learning, and
- certification of learning, rather than proficiency, as the final outcome.

All of these expectations may make the systematic approach for teaching both natural and successful. The challenge for teachers is how to assist students in their movement from the Discourse of schooling to the Discourse of employment.

Whatever a student's primary Discourse, certain assumptions can be made about the nature of the learning process. Learning requires variation in a relevance framework; learning requires unexpected events to occur (Marton & Booth, 1997). Learning is facilitated when outcomes are different from learners' initial expectations. The notion of the *zone of proximal development* attributed to Vygotksy (Salomon & Perkins, 1998; O'Connor, 1998) extends the concept of learning to include support from the environment and other persons. The *zone of proximal development* is the social space wherein learning occurs when a learner engages in activity that he or she can more or less successfully attempt with the help of a more knowledgeable other. The ideas of Discourse developed by Gee connect to these ideas about learning in that Discourse(s) provide the relevance framework that defines what variation is worth attending to. Other members and artifacts in the social practice defined by the Discourse provide the more knowledgeable "other," building a scaffold to the current skill level of the learner.

Further, Gee would argue that the outcome of learning is better understood as becoming a kind of person rather than gaining a particular skill. "Discourses create, produce and reproduce opportunities for people to be and recognize certain *kinds of people"* (Hacking 1986, 1994) [Gee, et al, 1996, p.10, emphasis in original]. These ideas are consistent with the literature on apprenticeship; see for example Rogoff (1990) and Rogoff and Lave (1984). The idea of participation in a Discourse provides a different conception of the goal of schooling. It is clear that the content of instruction should consequently not be limited to the functionality of the tools. The work of office administration consists largely of solving problems with tools. The problems involve communication, record keeping, accounting, and application of regulation and policy to particular, emergent situations. The tools (desktop computers) include ways of thinking about these kinds of problems.

An implication of thinking of technology-related education as being part of a Discourse is that the systematic approach to instruction is framed in the Discourse of the tool and its functionality. Instruction is *about* the tool. On the other hand, the minimalist approach is framed in the Discourse of the office. Moreover, instruction is *within* the Discourse, not about it. That is, the knowledge constructed is largely tacit and relates to how office workers, as a type of people, do things, as opposed to what the tools can do.

These idealized conceptions of instruction can be applied to whole programs. An office technology program may as a whole exhibit a systematic approach, with decontextualized courses about specific tools that are then integrated in later courses and applied to business-like problems. Alternatively, a whole office technology program may be minimalist, by being about business problem solving, with skill development with the appropriate tools emerging from the problem solving process.

It may also be that systematic and minimalist ways of teaching can also be found in the smallest day-to-day activities of students and teachers, independently of whether the overall program is systematic or minimalist in character. Who works out the solution to an unexpected software problem in the lab? Who does the keyboarding (drives the computer) to handle the problem? How long are students permitted to suffer with the ambiguity of a problem before help is rendered? What sources of help exist? A minimalist approach to these mundane activities may foster a deeper understanding (Marton & Booth, 1997) and a willing-to-fail stance (Dweck & Leggett, 1988). Both a deeper level of understanding and an attitude of risk taking can support transfer of learning by students to different situations, in particular, from the school setting to the work setting (Thomas, Anderson, Getahun, & Cooke, 1992).

In summary, two different goals can be prominent in employment-related programs. One implicitly recognizes the situation of schooling. The other very explicitly focuses on the eventual employment goals. When the goal is developing employment-related office technology skills, balance is needed between gaining technology skills and understanding the eventual work settings in which such skills have meaning.

One goal can be software skill development. If so, then the Dreyfus and Dreyfus (1986) skill-hierarchy model should prove helpful in interpreting the target skill level with various tools. Curriculum content and instructional materials are chosen accordingly. On the other hand, a second goal of instruction is movement out of the schooling Discourse and entry into the employed-worker Discourse—to become a certain kind of person. If this is the case, then the Dreyfus and Dreyfus model of skill acquisition needs to be applied to fluency with the ways of being an employed worker, not simply to facility with the tools of the trade.

Since success in a work setting is the goal, the Discourse of an employment setting should eventually become more dominant than the Discourse of schooling. This means the sooner the context and content of the employment field can become dominant in the learning setting, the better for allowing student participation in the employment Discourse. The instructional guidelines gleaned from exemplary office technology school settings will suggest ways in which students might be assisted in their progress as they move from schooling Discourses into work-related Discourses for using technology.

Issue of Program Identity

While examination of program identity was not an initial purpose of this study, it became apparent in conversations with teachers that at least three kinds of influences might be responsible for image problems in programs teaching introductory computing skills. One was the identification of some courses as “vocational,” “customized,” or something other than regular collegiate-level coursework. A second stemmed from the dominance of a self-pacing orientation in computer-intensive instruction. The third was related to enrollment changes matching job-market changes away from careers that carry sexist labels, such as “secretarial” and even “office.” This identity problem has lead to questions, perhaps disagreements, about which faculty group is best qualified to teach the basic computing skills required for employment. The following are illustrations of these three influences.

Course “diversification.” The first source stems from the effort of schools to engage in the “course diversification” described earlier, or to provide a wide variety of course lengths and enrollment options to meet different student’s needs. Students who learned computing skills in courses that were not regarded as “counting” for transfer credit did not think that their preparation was sufficiently rigorous or respectable. Here are some comments reflecting this concern.

Student:

- Concern for "transferable credit:"

> But, I just feel that a vocational certificate is not going...I was tested and told that I have a high IQ, and this and that, and I just feel that a vocational certificate, yeah, it might get me in on the ground floor somewhere and then it's up to me.
>
> But, I just feel that to really get somewhere that you need an A.A., an A.S., a B. A. or whatever, just a degree. But I'm satisfied; this is my last class. This and an income tax accounting class, and uh, I should pass both of them and I'll get my two certificates. I'm just not satisfied with my schooling. OK. I feel I can do better, I feel the school can do better. And I feel that I need more that if at forty years old I am going to start again and actually go someplace.

Two Teachers:

- Differentiating what should count for transfer credit:

> Maintaining a database in Access--no credit. Command language in Access should get credit. Going beyond routine user functions. You need the user functions first.
>
> No college credit for doing purely practical things. Learning to manipulate three different databases might mean credit. You will know much less in depth. More conceptual to get college credit.

> Introduction to Computers course was kept in CIS [Computer Information Systems]. Intro. to Microcomputers. [Specific course number mentioned]. It transfers to the University.

Student self-pacing. Second, if introductory computing classes were taught in open labs or with flexibility, student-paced patterns, they were also sometimes not regarded as academically rigorous. The content was judged as "skill-based" with little cognitive or problem-solving content. "Academic content" was sometimes regarded as requiring a lecture-format with in-depth technical content, such as might be considered part of programming instruction. In fact, "technical training" was often regarded as different from "computer training." A primary difference was a lecture format and instructional focus more directly on the internal operations of computing systems (Clothier, 1996). Desktop computing was thought to be characterized by more well-defined, routine skills. When this assumption was held, reservations could be raised about whether regular, licensed teachers were needed in computing labs, or whether unlicensed tutors or technical aids could satisfy students' needs for assistance. Here is one teacher's expression of frustration about differences in teaching philosophies across program areas, made more apparent when several program areas began to share common computer labs:

> We had a meeting; we were frustrated with the lab. Relocating was done three years ago. We liked the lab we had before. It was just us. No CIS. No Accounting. No walk-ins from any program. It was just our people [Office Technology]. We are able to operate with more discipline and more efficiency.
>
> We are different from CIS and accounting. While we are self-paced, we don't expect the students to be entirely on their own. We schedule hours for them and we take attendance.
>
> CIS has been under the policy that is self-directed--not just self-paced. Here are your materials and here are my office hours. It may be three hours a week. They [CIS faculty] are not as available [as OT faculty] for students. The aids start doing more. It borders on teaching. They don't care if the students do all the work in the last week. The CIS people are saying that they don't need certain things.

Nature of office employment. Thirdly, the Office Technology area as a whole has suffered from enrollment declines within the last decade because of perceptions held about the field of clerical support as a whole. Some people think the field is disappearing because others now assume many of the tasks commonly performed by secretaries and other administrative assistants. Sometimes the tasks have been automated and taken over directly by computers, such as the use of voice mail. In other instances, professionals and executives do the work themselves, or provide their own support, since they do not have access to a secretary (Berryman & Bailey, 1992; Carnevale & Rose, 1998; and Rifkin, 1995). Together with the perception that office jobs are shrinking (though this is still a large employment field), salaries are less attractive than those available for information technology jobs directly linked to technical support, systems development, and network maintenance (Office of Technology Policy, 1997).

These perceptions have been reinforced by current reports evaluating past vocational education efforts as unsuccessful, such as that by Grubb (1997) maintaining, "The most serious problem is that vocational education in the high school has carved out as its domain those entry-level occupations that are relatively low-paid and low status—secretary, …" (p. 41). If this is true at the high school level, postsecondary level administrative support programs may be equally suspect. In examining employment opportunities in the 21st century, Wirth (1992) also has made the claim that while there are now and will continue to be a large number of jobs for secretaries, clerks and computer operators in administrative support, "other than the computer operators, most of these categories require only modest skills" (p. 159).

In contrast, other researchers have labeled office education as a field to be kept separate from the general indictment of secondary-level vocational education. In part, Kliebard (1992) has agreed with Grubb, "As a policy that would bring needed job skills into the workplace and greater opportunity for the mass of the school population, vocational education can almost certainly be counted a failure" (p.198). But he added, "There are exceptions. Some evidence indicates that women were able to use commercial courses in order to make the transition to white-collar occupations" (p. 201). This transition, of course, was into jobs with high concentrations of women and relatively lower salaries compared to other professional while-collar jobs.

Raizen (1989) has made a similar judgment as Kliebard about the success of vocational education for office employment: "Also, despite the generally discouraging picture regarding job entry, success has been high in some selected occupations, particularly in the clerical and secretarial field" (p. 29). Her explanation for this success is enlightening in that it focuses on both "basic skills" and the nature of the work setting, reinforcing several of the findings of this study. Her analysis might even be said to allude to the commonalities between the Discourse of secretarial/clerical training and the Discourse(s) of office settings themselves:

> And what distinguishes the successful ones? Two conjectures might explain the success of vocational education programs in the clerical and secretarial fields. First, the skills that are taught (both technical and general) are useful in many different job settings: typing, word processing, answering the telephone courteously and informatively, setting up filing systems, anticipating the supervisor's needs, setting priorities among assignments, and running an efficient office. Second, school itself represents an appropriate model for the kinds of structure in which many clerical and secretarial personnel work—hierarchical and bureaucratized. The school setting in which the teacher is supervised by the principal, who is supervised by the administrator, who has policy set by the school board and the state, represents a hierarchical structure that determines not only what the students are to do and how they are to behave, but also how they are to be judged. Similarly, a secretary or clerk works for a supervisor who has several layers of supervisors above him or her, and these layers structure the secretary's tasks in a way similar to the way that school structures tasks for the student with neither the secretary nor the student having much power to influence the nature of the work. Also, learning to please the teacher is a good way of learning how to please one's supervisor, and learning to get along with one's fellow students at the same time as one competes with them is good preparation for relating to one's fellow workers later on. (p. 29)

While one might debate the status and skill requirements in the administrative support field, there is no doubt that teachers in this field face an identity crisis. The word "secretary" cannot be used to label the program. And, yet the skills that secretaries have generally possessed are in high demand. This is how one teacher expressed the dilemma.

Something that's happened here, and is fun to watch--when I first started here, I had waiting lists. I just had enrollment. No problem. We had educated people. And then, of course, [things changed and] women had much more opportunity, and so we didn't get those students, and so the enrollments been going down, down, down. But something else that happened here. We had a program called Office Mid-Management. It had a medium enrollment. Nothing much. All men, all men.

They magically changed it to Business Mid-Management. They changed one word.

JJL: So they'd be in the same class?

Yes. And it works very well.

JJL: It was called Office Mid-Management?

Office Mid-Management. It was changed to Business Mid-Management. The enrollment has just skyrocketed, and it took all our people. People that used to come to our programs. Their parents wanted to brag that their daughter is in Business Mid-Management at [school name]. They didn't want to brag that their daughter is in Administrative Assistant or Secretarial.

"Oh, you're paying to have your daughter learn to be a secretary? Why would you pay for that? Anybody can do that!" the image is so poor, that our students don't want the poor image, so they go to Business Mid-Management. But the irony of the whole thing is that they graduate, and guess what their jobs are?

JJL: Secretarial.

T: Yes.

The field of Office Technology—by whatever label may be more appropriate—might be rightfully identified as one in which "invisible work" crucial to a business's existence is undervalued (Nardi & Engestrom, 1998). Work can become invisible when it contains large components of interpersonal elements, communication skills, and competencies considered to be generally available in the population. Judgments about "little skill" overlook the underlying awareness, perhaps tacit knowledge, possessed by experienced support staff about expectations within a given work setting. Office staff who understand the history of an organization and the importance of certain relationships are able to perform tasks in a way not possible by less acculturated employees. Such essential understandings may be the very skills that disappear when businesses themselves eliminate support positions or move to greater reliance on temporary staff (Sennett, 1998). Such is the dilemma of educational programs—how to defend the offering of instructional programs when rapid changes are occurring in work settings themselves.

There is no doubt that office employees need to possess, among several competencies, high levels of computing skill, especially in word processing. The question is not whether these skills should be taught, but rather whether students will be enrolled in the advanced Office Technology specific courses where such skills are directly linked to office practices. Further, there is a question of who should teach the introductory computing courses required by all students, courses with a very weak link, if any, to any previous business competencies.

The following statements from a group of Computer Information Systems (CIS) faculty express the overlap existing in several schools between the teaching interests and teaching capabilities of CIS and Office Technology (OT) faculty:

> Microcomputers started here in CIS. Office Technology is office careers. Business English, Business Math. Things used in the office.
>
> We (CIS) turned the [Office] applications over to them [OT]. We didn't want to absorb it or see it broken up. We gave them the computer courses to teach.
>
> Most students [who take CIS courses] do not know how to type. The [computer performance] tests are timed, but they are such that one character in 5 seconds is the speed. This is different from Office Technology.
>
> [CIS instructor] teaches an introduction [to microcomputers]. [He] wants to teach the Web course. He will teach Javascript. OT would not do this. CIS would teach how to develop a web site.

In a different school, here is a comment from an Office Technology teacher about enrollment trends:

> …Because we're seeing less and less students entering our program each year, and if we didn't have software to teach, we wouldn't have jobs.

And in yet another school, a comment from an Office Technology teacher about which programs can and should teach software applications:

> It is a challenge to differentiate the computing classes offered by Marketing, Accounting, CIS, and Office Technology. There is a proposal to divide them as follows:
>
> Marketing: PPT
> Accounting: SS
> Comp. Info. Systems: DB
> Office Technology: Keyboarding and WP
>
> This was not acceptable to the OT faculty because they want to keep the business office applications prominent in the teaching of the software. All four areas are teaching these courses.

One of the exemplary programs included in this study has actually made itself distinctive by extending instruction beyond traditional office support areas into end-user support and multi-media development. They did this while still remaining separate from technical programs that are more formally developed to these fields, such as Computer Information Systems and Graphic Design. The following is an edited discussion of this program with a teacher and employer both present. It shows the kinds of efforts that are being taken to redefine a support area, Office Technology, that is in transition.

T: And there are three major points in our program development. One was when we made the decision to add an Office Systems program. The other was when there were hard decisions made a few years ago about what the industry needed and what was different from what they would get as an administrative secretary. And this past year we have made decisions about credit changes and blending the emphasis areas.

And what were we going to do with the Internship this year? And in my heart, I love the internship. But finding internships now is getting to be a little bit of a trick because of the policies that companies have if they have anyone on a lay-off or if they have any policy negotiations going on with people waiting to come back. We can't get people in to do work.

JJL [per previous examples]: So Interpersonal Communications is so important that you can't give it up and you'll get it in General Ed. And the internship is the other way--you're going to integrate it into the advanced classes.

T: Right. I think that we're going to find that it will work better that way. Because many of the internships that came from [the college person], who's our internship coordinator and works with the employers, end up being too much general office work and less software support kinds of things, and that isn't where they need their experience.

JJL: What do you think that the programs ought to be called, [employer], when people like you think about the jobs? What program title do you think would be the best preparation?

E: I think that Office Systems Specialist is a good one. We don't call them secretaries anymore; we call our people classified staff.

JJL: Classified Staff?

E: Classified Staff.

JJL: Because they are non-exempt?

E: Yeah. And I don't know. Office Systems Specialist.

T: Historically when someone asks me about what is an Office Systems Specialist graduate, what do they do? I draw them a five-ring diagram, and we laughingly call it a five-ring circus. The truth is that the students can go in different directions.

If they choose to put their efforts all in Desktop Publishing and Graphics kind of things, then that becomes a natural direction for them.

If they go more in the direction of Presentation and Web Design and Development, that can become a direction.

If they enjoy the Networking and Cabling, Installation and Maintenance, then that defines a track for them. What's interesting in the last few years is that many of our Office Specialist students have been paid by companies to be their Networking Specialists.

JJL: What's your fifth circle?

T: The fifth circle is the one that goes more into User Support and Network Training. I mentioned too that several graduates that have gone that direction. There is another group of administrative assistants that is working in managing secretaries. The area that is wonderment to us is the area of the Webmaster, not just web developer, but the web manager and network development. You know that you can get people that can't catch a typo and can do the networking, or the people that can catch all the typos, and can't do an FTP, so we're looking at that, the piece that's more the user support and corporate training. Our surveys are saying this needs to be expanded into a helpdesk offering. Those are all the directions that we see right now.

In summary, two key findings of this research are the largely decontextualized teaching of introductory software use, even when business examples are the textbook frame of reference, and the dominance of systematic teaching practices over more discovery-oriented minimalist teaching practices. A third finding of this study is concern about the image of various employment fields, particularly the contrast between “office technology” and “information technology.” This image affects whether programs will be offered, who will teach them, and how they will be taught. Given these three outcomes, three different avenues of further research are suggested below.

Research Recommendations

Three avenues for future research are implied by these circumstances—a strong preference for introductory software instruction requiring few or no prerequisite skills or content knowledge, to be taught in a direct, systematic manner, with the content having an uncertain career focus.

1. Investigation of the possibility of introducing problem-, project- or case-based software sooner than is now done in order to engage students in more realistic uses of software.
2. Investigation of the use of online, computer-supported learning settings for teaching software skills.
3. Investigation of the nature of office and technical support occupations to determine actual job needs, career paths, and appropriate program labels for instructional settings.

Problem-Based Learning. Project-based learning is being recommended more generally as a way to ask for active problem solving on the part of students in a way that integrates relevant content areas, basic skills, and the context of a work setting. In current instructional settings, if students do not choose to take more advanced technology-related coursework, they might not ever be asked to bring their own business or personal problems to technology use. This means that many students may not become engaged in work that asks them to independently develop problem solutions. If students do not participate in internships, they will be particularly deprived of an opportunity to confront the difference between a school-based Discourse for learning and the Discourses of a work setting.

While the current research sites provided insights into what certain teaching practices have been chosen, additional research is needed to more fully explore the nature of learning sites where more minimalist approaches are used. Schools that are responding to the increased interest in case-based learning and project-based learning are the likely places to look (Duffy & Cunningham, 1996; Schank, Berman & Macpherson, 1999; Schank & Cleary, 1995; and Vickers, 1996). Advocates of scenario-based or project-based learning claim involvement in realistic activities is essential to gaining the types of understanding that will transfer outside of school settings. A key premise is that learning is facilitated when the outcomes of realistic activities are different from what learners expected—*expectation failure*. On the job, such learning can be costly. In a school setting, solving problems can be controlled to build on what students already know.

Businesses have begun to realize the benefit of simulated learning settings, and computing skills are being taught with commercial computer-based software that asks students to engage in case-based learning (Schank, 1997). While merging the terminology of *simulations, cases, projects,* and *problems* overlooks key distinctions, as if they represented the same kind of instructional model, they are all quite different (Blumenfeld, Soloway, Marx, Krajcik, Guzdial, & Palincsar, 1991; Jonassen, 1999; Kolodner, 1993; and Kolodner, Schwarz, Barkai, Levy-Neumand, Tcherni, & Turbovsk, 1997). The main point is to embed students' learning in a work setting in a way that asks for engagement in both tool use and thinking about the purpose for its use. Key questions are *when* to introduce more realistic problems into instruction in software use, and *how to balance* the need for both contextual examples of software use with the understanding of general software functionality.

Online learning tools for software. As has been suggested with respect to case- and problem-based learning, much of the basic instruction about how to use software tools effectively is becoming available in a computer-based format. One of the schools participating in this project had already tried such software with varying degrees of satisfaction. Major commercial vendors offer a variety of computer-based instructional options for learning software use. More schools are likely to explore these options as they attempt to deal with the regular changes in software versions and the accompanying need to purchase new textbooks. Such textbooks, with their plentiful use of color and illustrations, are expensive. Schools are giving serious evaluative attention to the option of purchasing access to on-line instructional materials that can be more easily updated than paper textbooks. This is especially the case when the school rather than the student buys the textbooks. The effectiveness of such software in traditional school settings needs to be assessed both as a stand-alone system and in conjunction with other classroom-based or problem-based activities.

Related to the adoption of computer-based instructional tools is the desire by some schools to use such software for online evaluation of students. Several schools in this project were offering students the option of certificates to verify their learning accomplishments and provide a credential that was valued by potential employers. While individual schools created some of these certificates, major software vendors also offer certificates. Such "commercial" certificates frequently require that testing be done with online software. The validity of such testing in relation to actual on-the-job accomplishments has yet to be demonstrated.

Nature of office computer use. As schools attempt to provide career direction for students, they need to regularly reassess actual employment needs. Previous attempts to provide this essential updating of the state of the field have depended in large part upon task analytic approaches that generally result in lists of behaviors for curriculum development (Lambrecht & Sheng, 1998). If teachers are to have a fuller understanding of the work settings in which office staff do their work, more holistic approaches to curriculum development and work standards may be more effective (Bailey & Merritt, 1995; Merritt, 1996). These include work analysis approaches, called "professional" approaches, which may be considered more ethnographic in their study of work settings. These more holistic approaches make an attempt to understand the broader work environment, not just the job tasks assigned to a particular job title.

Such professional approaches, in contrast to task analytic approaches, have the potential to capture more of the subtleties of how work is organized and accomplished. The primary difference between the task analytic and professional approaches is that an attempt is made with the professional approach to situate the tasks and worker competencies within a work setting. Hart-Landsberg, et al. (1992) provide an example of how such workplace analyses might be carried out through case-based methodology. They have characterized the process of learning to work by actually doing work as "learning the ropes" within a specific job context.

The concept of sensitivity to working relationships has been a prominent finding in this study. Both employer interviews and the characterization of office work by Raizen (1989) conveyed the importance of "reading the work environment." Students and teachers need better understandings of what it means to interpret expectations and assess priorities, of what it means to come to understand the business culture, the profit motive, the need for standards of a certain type, and interpreting standards which may change depending on the circumstances. This is not a new insight for business educators. But what this language actually means in real work settings is yet to be fully developed. In particular, teachers need models of how such cultural understandings might be conveyed to students prior to their participation in an actual work setting.

Questions should also be raised about the circumstances of office careers—their potential for providing worthwhile and satisfying employment. When work is looked at more holistically, different "information ecologies" may be identified (Nardi & O'Day, 1999). The concept of "information ecologies" means that "different networks of relationships, values, and motivations" (p. 30) are exhibited within settings that are also dependent upon computing tools. With an understanding of what various office "information ecologies" look like, both positive and negative, students and teachers can then discuss the kinds of options students have. It will likely be the case that students with a full portfolio of well-developed skills can have more choices for satisfying employment settings.

Administrative support and other positions that may be characterized as "computer end-users" will continue to be affected by technological advances. A positive expectation is that computer-based devices will continue to assume repetitive and/or predictable tasks, allowing people to pursue more engaging and yet unknown kinds of work. This engaging work is likely to require greater understanding of technology, good judgment about how to use such technology effectively within business settings, and, thus, more options and need for career-related education.

As software continues to become easier to use, it is also likely to become more specialized, well suited to specific tasks (Norman, 1998). Task-specific technology is also likely to become easier to learn and less the object of instruction in and of itself. If software will become more specialized and easier to learn, the challenge for business employees in a variety of support positions will be to understand the business purposes of their employment so that they can choose appropriate technology tools and use them effectively. Research is needed to understand the nature of these business work settings.

Learning more about what computer end-users do, the variety of ways in which support staff will continue to use technology, and then presenting these career options to students in an attractive way, will lead to the continued vitality of technology-related programs. Greater understanding about the broad field of "end-user computing" has the potential to re-vitalize programs now labeled Office Systems and Office Technology that are struggling with the question of how best to promote or change the "office" part of their identity.

Chapter VI: Program Guidelines

Guidelines for effective employment-related office technology skills instruction can be drawn from the exemplary programs that participated in this study. While the basic structure of these guidelines is that of systematic instruction, an attempt has been made to capture those elements of minimalist teaching practices that were observed within these exemplary programs. When technology instruction is designed for groups characterized as Novices or even Advanced Beginners, the need for structure dominates, at least initially, over more open, discovery-oriented teaching approaches. But, movement should be made to open-ended problems as so as students have experienced success with seeing how software operates.

Guidelines are provided first for overall program structure, followed by specific instructional practices.

I. Provision for Different Student Goals and Capabilities

A. Responsiveness to Student Career Goals

1. The Office Technology field is broad, and students prepared for business employment as support staff should know that they can obtain work in a variety of businesses.

2. Two areas of specialization continue to be in demand within the broad area of Office Technology/Administrative Support: Medical and Legal Office Assistant Programs.

3. Areas of career growth that match the employment interests of office support staff are those of end-user support and training, desktop publishing, web page design and development, and perhaps network support and maintenance.

B. Scheduling options.

1. *Semester-long courses.* Courses fitting the regular school scheduling pattern, generally semester- or quarter-length courses, allow greater depth of software instruction. They also fit into the structure of degree-granting programs that some students desire either as an employment credential or for transfer to another institution.

2. *Fixed-hour length courses*. When students are returning to school with funding support from previous employers or government agencies, the courses they complete must match constraints set by the funding agency. A common constraint is that the software skills learned must be in courses of a fixed-hour

length. Schools have been able to accommodate this need in two ways: by creating special "vocational" or "customized" training programs for specific software, or by incorporating these time constraints within their regularly scheduled courses. Separate courses can be justified when the number of students makes them efficient to offer.

3. *Computer workshops*. Courses designed for employed persons who wish specific software skill frequently face the constraint of accommodating work schedules. Employers have generally been willing to allow employees to take four-hour or eight-hour leaves for intensive software training. This leads to creation of yet another sequence of software courses for short intensive workshops.

   A risk of such workshops is that of not knowing what students actually accomplish, since formal achievement evaluation is seldom possible within the very short course time period. Student satisfaction is generally the criterion of whether such courses are effective. The fact that such workshops continue to be offered suggests that they do meet important student needs.

4. *Open computing labs*. Flexible, individualized instruction within open computing labs or “flex” labs may be more convenient for students whose schedules do not allow them to attend class regularly. Such individualized progress can also allow students to complete courses in shorter periods of time.

C. Provision for Student Self-Pacing

1. Even when students meet together in regularly scheduled courses, provision should be made to allow student self-pacing, rather than asking students to proceed through course materials in unison.

2. Whether a "flex lab" exists along side of semester-length, regularly scheduled class, or whether the "regular" program also is designed for individualized instruction, certain student characteristics are essential if they are to benefit from less course structure. Students need to be strongly goal focused and committed to a personal deadline if they are to complete an individualized program. Even regular assistance from a teacher and follow-up messages about attendance and course progress will not compensate for lack of personal self-discipline. However, because of student commitments to employment and family demands that prevent regular class attendance, a "flex lab" or individualized schedule is exactly what many students need.

3. Technology-related skills can effectively be taught on an individualized basis through open computer labs or “flex” labs when these program characteristics

are present:

- ✓ Specific student competency outcomes can be identified and evaluation instruments prepared so that agreement exists about what students are to learn and when they have met program goals.

- ✓ Course objectives and related assessment instruments are sufficiently objective and unambiguous that several different instructors will interpret students' work in the same way.

- ✓ Good communication and respect exists among faculty such that a cohesive and consistent image is presented to students about program expectations.

- ✓ Efficient record keeping systems are available to monitor student attendance and progress.

- ✓ High-level school administrative support is consistently available to provide staffing of the lab with an adequate number of qualified teachers.

- ✓ Both technical and tutorial support staff are readily available.

- ✓ Instructional materials are designed for independent student use. (See item below about textbooks under Instructional Practices.)

- ✓ Independent student progress is reasonable, in contrast to group projects, as a way of learning specific software skills. This means students need to independently master a sequence of instructions about software functions, or a hierarchy of skills, before proceeding to more advanced work.

- ✓ In the Office Technology area, the above characteristics are likely to occur when teaching these courses: introductory and intermediate software application courses, keyboarding and machine calculation.

- ✓ In the Office Technology area, courses generally not taught effectively in open computing labs include: advanced computing courses that are more project-based, machine transcription, business communication, information management, professional development and human relations courses, office procedures, or the internship seminar.

II. Instructional Practices

A. Characteristics of effective applications software textbooks

1. Clear procedural and conceptual explanations
2. Short exercises within lessons illustrating software features
3. Minimal reading requirements
4. Clear, colorful diagrams to illustrate software features
5. Detailed illustrations to provide exercise feedback
6. Check-test exercises to highlight key concepts and procedures
7. Progressively more complex exercises. "Complex" means that less explanation is provided about what is to be done or the procedures for doing it.
8. Accompanying testing materials in the form of both concept tests and performance tests.

The chief benefit to teachers from systematic instructional materials is that comprehensive presentation of software features can be assured, including presentation and illustration of "basic skills" needed to use computers successfully. Because students have written access (and sometimes audio-visual presentation as well) to this information, the teacher is free to assist individual students with the unexpected, but very common, problems of using software.

B. Scheduling to provide structure, even with student self-pacing

1. Require that key assignments be handed in.
2. Require that tests be taken on set days.
3. Consider what additional assignments need to be "handed in" to assure student participation in acquiring necessary software understanding.

Some teachers think that giving performance tests is sufficient to assure that individual students have attained course goals. However, other teachers think that students miss essential concepts if they are allowed to skip exercises because the exercises are not handed in. Depending upon the maturity level of students, more structure may be needed to assist student learning. Unless students have personal goals and prior background that support the self-discipline needed to notice the detailed features of computing software, structure in the form of course assignments and specific due dates can help.

C. Provision of opportunities for students to work together

1. Allow "lab time" within scheduled courses and permit student conversation.
2. Allow "open lab" time outside of scheduled courses that gives students access to common groups of computers.
3. Assign "peer tutors" for students needing extra assistance. In some schools these are paid positions for students within regular courses.
4. Create joint projects that allow students to make decisions about how to complete large, complex assignments. This works effectively with more advanced students.

Students benefit markedly by being able to obtain immediate help from their peers. They also can learn a considerable amount about how software works by explaining software procedures to others. Students gain considerable self-satisfaction from being regarded as capable sources of assistance. The practice of asking for and providing assistance with software problems appears regularly in employment settings. Therefore, students can benefit from learning in school the value of, and the appropriate use of, collegial resources.

D. Ensuring access to instructional support

1. Allow independent work during class periods when an instructor is available.
2. Staff "open labs" with qualified teachers who can provide specific software instruction.
3. Identify "peer tutors" who can provide assistance in "open labs." The challenge with this approach is to provide instruction for the "peer tutors" so that they do not assume the role of telling students what to do instead of helping them solve problems.

The purpose of instructional support is not only to provide guidance to students as they use software features independently; it is an important source of feedback. Instructional support, like textbook illustrations and problem-solution models, provides feedback about the accuracy of students’ work and the clarity of their thinking.

E. Completion of large-scale, open-ended problems

Students need the experience of completing problems that have the following characteristics:

1. They are realistic problems from business offices.
2. They are unstructured to the extent that the solution is not self-evident.
3. They generally have a variety of possible solutions.

4. They allow a variety of ways of accomplishing the preferred solution.
5. They include opportunities to use a variety of software programs.
6. They require that business standards be met in the recommended solutions.
7. A variety of problems are available to allow students to reuse key software functions within different problem contexts.

More advanced students realize the traps that come with systematic instruction. Students can complete step-by-step exercises without fully understanding the implications of different software features. Too often, it is possible to complete textbook exercises and focus solely on the formatting, editing, or syntactical features of business documents without thinking about the business activity of which the document is a part.

Project-oriented advanced classes and internships allow students to think about software problems from the perspective of the business purpose being served. Students can then synthesize a variety of skills learned in different courses in a way not possible in courses that focus on a single software application.

III. Evaluation Practices

A. Performance tests—the only valid measure of software use

1. Ensure inclusion of representative software features.
2. Include teacher-prepared data sets to minimize data entry by students.
3. Use realistic problems so that students will see them as genuine and as matching workplace expectations.
4. When possible, allow students to complete entire exams so that all problems can be attempted.
5. Consider keeping track of student completion time in order to incorporate speed as well as quality in test scores.
6. Consider continuing test administration over more than one class period, if extra time is needed for realistic problems.

Most teachers augmented the performance tests that came with the textbooks with their own performance tests.

B. "Open-book" tests

1. Open-book exams encourage a focus on understanding of software functionality rather than the memorization of terminology or keystrokes.
2. Access to resources such as reference manuals, help screen, and software tutorials parallels the actual use of software in the work place.

3. Open-book exams encourage students to become familiar with using reference manuals and other resources.

C. Concept or theory tests

1. Concept or theory tests allow for a broader coverage of software topics than performance tests alone.
2. Concept or theory tests encourage students to focus attention on terminology.
3. Concept or theory tests allow students to identify the rationale for their thinking in ways that might not be possible with performance tests alone.
4. Using concept tests prior to performance tests can ensure that students understand time-consuming, wasteful, or risky operations before actually encountering problems during a performance exam.

Not all teachers think that concept or theory exams are as important as performance exams for assessing software use. Those who do value concept tests argue that the focus on software concepts and terminology ensures later performance success. Understanding terminology is necessary to use procedures effectively, use "help" menus and paper indexes, and to communicate with others about software questions. Some teachers allow these tests to be "open book" just as performance tests. When reading is difficult for students, "open-book" exams provide an incentive to become familiar with the textbook and other resources.

A general admonition for teaching applications software, implicit throughout these guidelines, is to match the instructional settings and the evaluation settings as much as possible to those that students will experience outside of school. This means making a conspicuous link between in-school practices and workplace practices.

Such a link makes it possible to talk about truly developing employment-related office technology skills. To reiterate ideas developed earlier, when success in a work setting is the goal, the Discourse of an employment setting should eventually become more dominant than the Discourse of schooling. This means the sooner the context and content of the employment field can become dominant in the learning setting, the better for allowing student participation in the employment Discourse.

REFERENCES

Alexander, M. W. (1996). Use of presentation graphics software by administrative support personnel: Implications for business educators. The Delta Pi Epsilon Journal, 38(3), 139-155.

Alexander, M. W. (1998). Uses of presentation graphics technologies in business and academic. OSRA Systems Research Journal, 16(1), 32-38.

American Institutes for Research (AIR). (April, 1998). The Critical Incident Technique. AIR Web Page. Available: www.air.org/airweb/about/critical.html

Anderson, R. E., & Ronnkvist, A. (1999, June). The presence of computer in American schools. (Teaching, Learning and Computing: 1998 National Survey, Report #2). Irvine, CA and Minneapolis, MN: Center for Research on Information Technology and Organizations. Available at WWW URL: http:///www.crito.uci.edu/TLC/findings/Internet-Use/startpage.htm

Arney, J. B. (1998). A comparison of computer skills needed by business administration graduates as perceived by colleges and universities and human resource managers in Indiana. NABTE Review, 25, 44-48.

Ascher, C. (1994). Cooperative education as a strategy for school-to-work transition. CenterFocus No. 3. [OnLine]. Available at WWW URL: http://vocserve.berkeley.edu/CenterFocus/CF3.html

Bailey, T. (1990, May). Changes in the nature and structure of work: Implications for skill requirements and skill formation. MDS-007. Berkeley, CA: National Center for Research in Vocational Education.

Bailey, T., & Merritt, D. (December, 1995). Making sense of industry-based skill standards. MDS-777. Berkeley, CA: National Center for Research in Vocational Education. [Online]. Available WWW URL: http://ncrve.berkeley.edu/MDS-777/Front.html

Bartel, K., & Bartholome, L. W. (1999). University distance and on-campus learners: A comparison of beginning microcomputer applications students. NABTE Review, 26, 46-50.

Berryman, S., & Bailey, T. (1992). The double helix of education and the economy. New York, NY: Institute on Education and the Economy, Teachers College.

Blumenfeld, P.C., Soloway, E., Marx, R.W., Krajcik, J.S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. Educational Psychologist, 26(3 & 4), 369-398.

Boiarsky, C. R. (1997). The art of workplace English: A curriculum for all students. Portsmouth, NH: Boynton/Cook Publishers.

Borg, W. R., & Gall, M. D. (1989). Educational research: An introduction. (5th ed.). New York, NY: Longman.

Brandon, B., Clothier, P., Copeland, S., Crowell, P., Dodge, B., Maday, P., Masie, E., Perry, G., Slobodian, G., Woodie, C., Zemke, R., & Zenke, S. (1996). The computer trainer's personal training guide. Indianapolis, IN: Que Education and Training.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (1999). How people learn: Brain, mind, experience, and school. Washington, D.C.: National Academy Press.

Bursey, T. L. (1998). A creative way to assess word processing skills. Business Education Forum, 52(3), 32-33, 54.

Campione, J. C., Shapiro, A. M., & Brown, A. L. (1995). Forms of transfer in a community of learners: Flexible learning and understanding. In A. McKeough, J. Lupart, & A. Marini (Eds.), Teaching for transfer: Fostering generalization in learning (pp. 35-68). Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.

Carnevale, A. P., Gainer, L.J., & Meltzer, A. S. (1990). Workplace basics: The essential skills employers want. San Francisco: Jossey-Bass Publisher.

Carnevale, A. P., & Rose, S. J. (1998). Education for what? The new office economy. Princeton, NJ: Educational Testing Service.

Carroll, J. M. (1990). The Nurnberg funnel: Designing minimalist instruction for practical computer skill. Cambridge, MA: The MIT Press.

Carroll, J. M. (Ed.). (1998). Minimalism beyond the Nurnberg funnel. Cambridge, MA: The MIT Press.

Carroll, J. M., & Van der Meij, H. (1998). Ten misconceptions about minimalism. In J. M. Carroll, (Ed.), Minimalism beyond the Nurnberg funnel. Cambridge, MA: The MIT Press, pp. 54-90.

Ceci, S. J., & Roazzi, A. (1994). The effects of context on cognition: Postcards from Brazil. In R. J. Sternberg & R. K. Wagner (Eds.), Mind in context: Interactionist perspectives on human intelligence (pp. 74-104). New York, NY: Cambridge University Press.

Chalupa, M. R. (1997). The status of presentation software usage in the workplace. OSRA Systems Research Journal, 15(1), 33-41.

Chalupa, M. R., & Sormunen, C. (1996). The status of presentation software and graphic design training. The Delta Pi Epsilon Journal, 38(4), 222-236.

Clothier, P. (1996). The complete computer trainer. New York, NY: McGraw-Hill .

Copeland, S. (1996). Course development. In B. Brandon, et al (Eds.), The computer trainer's personal training guide (pp. 111-128). Indianapolis, IN: Que Education & Training.

Davis, D. C. (1997). The Internet and electronic communications systems in business and industry. Business Education Forum, 51(4), 11-14.

Davis, D., & Gonzenbach, N. (1995). Trends in information and image management. The Delta Pi Epsilon Journal, 37(3), 128-141.

Dede, C. (1996). Integrating learning and working. NSF Educational Technology Workshop: Setting a Research Agenda for Computer Science in Educational Technology. [OnLine]. Available at WWW URL: http://www.cc.gatech.edu/gvu/edtech/nsfws/integrated.html

Dick, W., & Carey, L. (1996). The systematic design of instruction (4th Ed.). New York, NY: Harper Collins College Publishers.

Di Salvo, V. S., Nikkel, E., & Monroe, C. (1989). Theory and practice: A field investigation and identification of group members' perceptions of problems facing natural work group. Small Group Behavior, 20(4), 551-567.

Dreyfus, H. L. & Dreyfus, S. E. (1986). Mind over machine. New York, NY: The Free Press.

Duffy, T. M., & Cunningham, D. J. (1996). Constructivism: Implications for the design and delivery of instruction. In D. H. Jonassen (Ed.), Handbook of Research for Educational Communications and Technology, Chapter 7, pp. 170-198. New York, NY: Simon & Schuster Macmillan.

Duffy, T. M., & Jonassen, D. H. , (Eds.). (1992). Constructivism and the technology of instruction: A conversation. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Duffy, T. M., Lowyck, J., Jonassen, D. H., & Welsh, T. M. (Eds.). (1993). Designing environments for constructive learning. New York: NY: Springer-Verlag.

Dweck, C. S., & Leggett, E. (1988). A social-cognitive approach to motivation and personality. Psychological Review, 95, 256-273.

Erthal, M. J., Wiggs, L. H., & Huter. L. (1996). Factors that contribute to student success in beginning microcomputer classes. NABTE Review. 23, 47-52.

Ertmer, P. A. & Newby, T. J. (1993). Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective. Performance Improvement Quarterly, 6(4), 50-72.

Farkas, D. K., & Williams, T. R. (1990). John Carrolls' *Nurnberg Funnel* and Minimalist Documentation. IEEE Transactions in Professional Communication, 33(4), 182-187.

Flanagan, J. C. (1954). The Critical Incident Technique. Psychological Bulletin, 51(4), 327-358.

Flister, S. P. (1998). Ways of thinking about data. Doctoral dissertation. University of Minnesota, St. Paul.

Frueling, R. T., Kerin, J. D., & Sebastian, D. (1997). The information technology age and its impact on business and business education curriculums. The Delta Pi Epsilon Journal, 39(2), 64-75.

Gagne, R. M. (1985). The conditions of learning and theory of instruction (4th Ed.). New York, NY: Holt, Rinehart and Winston.

Gagne, R. M., Briggs, L. J., & Wager, W. W. (1992). Principles of instructional design (4th Ed.). Fort Worth, TX: Harcourt Brace Jovanovich.

Gagne, R. M., & Medsker, K. L. (1996). The conditions of learning: Training applications. Forth Worth, TX: Harcourt Brace College.

Gay, L. R., & Diehl, P. L. (1992). Research methods for business and management. New York, NY: Macmillan Publishing Company.

Gee, J. P. (1998). Language, learning and latecomers: Discourses in education. Paper prepared for a talk at the University of Minnesota.

Gee, J. P., Hull, G., & Lankshear, C. (1996). The new work order: Behind the language of the new capitalism. Boulder,CO: Westview/Harper Collins.

Gee, J. P. (1992). The social mind: Language, ideology, and social practice. New York, NY: Bergin and Garvey.

Gibbs, W. W. (July, 1997). Taking computer to task. Scientific American. [OnLine]. Available at WWW URL: http://www.sciam.com/0797/trends.html

Glick, J. (1995). Intellectual and manual labor: Implications for developmental theory. In L. Martin, K. Nelson, & E. Tobach (Eds.), Sociocultural psychology: Theory and practice of doing and knowing, pp. 357-382. New York, NY: Cambridge University Press.

Gonzenbach, N. M. (1998). Developing an information systems curriculum with input from business and industry. Office Systems Research Journal, 16(1), 9-14.

Groneman, N., & Buzzard, J. (1995). Competencies needed by computer educators. NABTE Review, 22, 15-19.

Grubb, W. N. (1997). Cited in L. Olson, The school-to-work revolution: How employers and educators are joining forces to prepare tomorrow's skilled workforce. Reading, MA: Perseus Books.

Hackos, J. T., & Stevens, D. M. (1997). Standards for online communication. New York, NY: John Wiley & Sons, Inc.

Hart-Landsberg, S., Braunger, J., Reder, S., & Cross, M. M. (1992, December). Learning the ropes: The social construction of work-based learning. MDS-413. Berkeley, CA: National Center for Research in Vocational Education.

Hellman, R. (1992). Comprehensive user education for successful end-user computing. Information Techno1ogy & People 6(1), 65-84.

Hemby, K. V. (1997a). Effects of keyboarding skill on self-reported computer anxiety among traditional versus nontraditional college students. The Delta Pi Epsilon Journal, 39(1), 24-38.

Hemby, K. V. (1997b). Teaching adults in the business education classroom--Computer anxiety and its impact on instructional methods. Business Education Forum, 52(1), 36-38.

Hignite, M. A., & Echternacht, L. J. (1994). Computer literacy levels of prospective business teachers: A comparative study. NABTE Review. 21, 35-38.

Jaderstrom, S. (1995). Technology's impact on computer and business curricula. In N. J. Groneman & K. C. Kaser (Eds.), Technology in the classroom. National

Business Education Yearbook, No. 33 (pp. 1-9). Reston, VA: National Business Education Association.

Johnson, J. E., Hill, M., & Lankford, W. (1998). Teaching computer skills via distance learning. Business Education Forum, 52(3), 39-42.

Jonassen, D. H. (1992). Evaluating constructivistic learning. In T. M. Duffy & D. H. Jonassen, (Eds.). Constructivism and the technology of instruction: A conversation (pp. 137-148). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Jonassen, D. H. (1999). Designing constructivist learning environments. In C. M. Reigeluth (Ed.), Instructional-design theories and models, Volume II (pp. 215-239). Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.

Jonassen, D. H., Mayes, T., & McAleese, R. (1993). A manifesto for a constructivist approach to use of technology in higher education. In T. M. Duffy, J. Lowyck, D. H. Jonassen, & T. M. Welsh, (Eds.). Designing environments for constructive learning. New York: NY: Springer-Verlag, pp. 231-248.

Jonassen, D. H., & Rohrer-Murphy, L. (1999). Activity theory as a framework for designing constructivist learning environments. Educational Technology Research and Development, 47(1), 61-79.

Joyce, B., Weil, M., Showers, B. (1992). Models of Teaching (4th ed). Boston, MA: Allyn & Bacon.

Keller, C. M., & Keller, J. D. (1996). Cognition and tool use: The blacksmith at work. New York, NY: Cambridge University Press.

Kliebard, H. M. (1992). Forging the American curriculum: Essays in curriculum history and theory. New York, NY: Routledge.

Kolodner, J. (1993). Case-based reasoning. San Mateo, Calif.: Morgan Kaufmann,

Kolodner, J. L., Schwarz, B., Barkai, R. D., Levy-Neumand, E., Tcherni, A., & Turbovsk, A. (1997). Roles of a case library as a collaborative tool for fostering argumentation. In R. Hall, N. Miyake, & N. Enyedy (Eds.), Proceedings of 1997 computer support for collaborative learning (CSCL97) (pp. 150-156). Hillsdale, N.J.: Erlbaum. Available at WWW URL: http://www.oise.utoronto.ca/cscl/noframe

Kuutti, K. (1996). Activity theory as a potential framework for human-computer interaction research. In B. A. Nardi, (Ed.), Context and consciousness: Activity theory and human-computer interaction (pp. 17-44). Cambridge, MA: The MIT Press.

LaBonty, D. (1997). A comparison of spaced or massed scheduling and live or video instruction in a microcomputer applications course. NABTE Review, 24, 35-39.

Lambrecht, J. J., & Sheng, L. (1998). Job task analysis approaches for microcomputer application curriculum development. In Conference Proceedings of the National Delta Pi Epsilon Research Conference, Louisville, Kentucky, November 19-20, 1998.

Landauer, T. K. (1996). The trouble with computers: Usefulness, usability, and productivity. Cambridge, MA: MIT Press.

Lave, J. (1988). Cognition in practice: Mind, mathematics and culture in everyday life. New York, NY: Cambridge University Press.

Lundgren, C. A., Lundgren, T. D., & Mundrake, G. A., (1995). Teaching computer applications. Little Rock, AR: Delta Pi Epsilon.

Machung, A. (1988). 'Who needs a personality to talk to a machine?': Communication in the automated office. In C. Kramarea (Ed.), Technology and women's voices: Keeping in touch. New York, NY: Routledge & Kegan Paul.

Marton, F., & Booth, S. (1997). Learning and awareness. Mahwah, NJ: Lawrence Erlbaum Associates.

Masie, E. (1989). The computer training handbook: Strategies for helping people to learn technology (2nd ed.). Minneapolis, MN: Lakewood Books.

Mathematical Sciences Education Board, Center for Science, Mathematics, and Engineering Education, National Research Council. (1998). High school mathematics at work: Essays and examples for the education of all students. Washington, D.C.: National Academy Press.

Mayer, R. E. (1999). Designing instruction for constructivist learning. In C. M. Reigeluth (Ed.), Instructional-design theories and models, Volume II (pp. 141-239). Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.

McDonald, M. L., & Echternacht, L. J. (1998). Teaching microcomputer software applications (electronic spreadsheets): Guided practice vs. independent practice. NABTE Review, 25, 16-20.

McEwen, B. C. (1996). Teaching microcomputer software skills. Business Education Forum, 50(4), 15-19.

Merritt, D. (April, 1996). A conceptual framework for industry-based skill standards. Centerfocus, No. 11. [OnLine.] Available at WWW URL: http://ncrve.berkeley.edu/CenterFocus/CF11.html

Mirel, B. (1998). Minimalism for complex tasks. In J. M. Carroll, (Ed.), Minimalism beyond the Nurnberg funnel. Cambridge, MA: The MIT Press, pp. 179-218.

Moss, J., Jr., Lambrecht, J. J., Jensrud, Q., & Finch, C. R., (1994, September). Leader Attributes Inventory Manual. MDS-730. Berkeley, CA: National Center for Research in Vocational Education.

Mundrake, G. A. (1998). Fifteen learning activities for a multimedia class. Business Education Forum, 52(4), 40-41.

Murnane, R. J., & Levy, F. (1996). Teaching the new basic skills: Principles for educating children to thrive in a challenging economy. New York, NY: Martin Kessler Books, The Free Press.

Nardi, B. A. (1993). A small matter of programming. Cambridge, MA: The MIT Press.

Nardi, B. A. (1996). Activity theory and human-computer interaction. In B. A. Nardi, (Ed.), Context and consciousness: Activity theory and human-computer interaction (pp. 7-16). Cambridge, MA: The MIT Press.

Nardi, B. A., & O'Day, V. L. (1999). Information ecologies: Using technology with heart. Cambridge, MA: The MIT Press.

Nardi, B., & Engestrom, Y. (1998, May). The issue of invisible work. Computer-Supported Cooperative Work: A Journal of Collaborative Computing, 3, 1-2.

Norman, D. A. (1998). The invisible computer: Why good products fail, the personal computer is so complex, and information appliances are the solution. Cambridge, MA: The MIT Press.

North, A. B., & Worth, W. E. (1997). Workplace competencies: Trends in advertised entry-level technology, interpersonal, and basic communications job skills, 1992-1995. OSRA Systems Research Journal, 15(1), 1-6.

Nourse, M. E. (1997). Look what arrived in the e-mail. Business Education Forum, 52(2), 45-47.

Nunes, T., Schliemann, A. D., & Carraher, D. W. (1993). Street mathematics and school mathematics. New York, NY: Cambridge University Press.

Oblinger, D. G. & Maruyama, M. K. (1996). Distributed learning. CAUSE Professional Paper Series, #14. Boulder, CO: CAUSE.

Oblinger, D. G., & Rush, S. C. (Editors). (1997). The learning revolution: The challenge of information technology in the academy. Bolton, MA: Anker Publishing company, Inc.

O'Connor, M. C. (1998). Can we trace the "efficacy of social constructivism"? In P. D. Pearson & A Iran-Nejad (Eds.), Review of Research in Education, No. 23. Washington, D.C.: American Educational Research Association, pp. 25-72.

Office of Technology Policy. (1997). America's new deficit: The shortage of information technology workers. Washington, D.C.: Office of Technology Policy, U.S. Dept. of Commerce. Available at URL: http://www.ta.doc.gov/otpolicy/reports.htm#USTPS

Ostwald, T., & Stulz, K. (1996). Using criteria-based evaluation in computer applications courses. Business Education Forum, 50(3), 33-35.

Perry, W. (1998). What software skills do employers want their employees to possess? Business Education Forum, 52(3), 35-38.

Ramsey, J. E., & Oatley, K. (1992). Designing minimal computer manuals from scratch. Instructional Science, 21, 85-98

Raizen, S. A. (1989, December). Reforming education for work: A cognitive science perspective. MDS-024. Berkeley, CA: National Center for Research in Vocational Education.

Ramsey, K., Stasz, C., Ormseth, T., Eden, R., & Co, J. (1997, September). Designing classrooms that work: Teacher training guide. MDS-963. Berkeley, CA: National Center for Research in Vocational Education, University of California at Berkeley.

Redish, J. (1998). Minimalism in technical communication: Some issues to consider. In J. M. Carroll, (Ed.), Minimalism beyond the Nurnberg funnel (pp. 219-245). Cambridge, MA: The MIT Press.

Regan, E. A. (1996, Spring). Organization and end-user information systems: Defining the field. Office Systems Research Journal, 14(1), 3-12.

Resnick, L. B., & Wirt, J. G. (Eds.). (1996). Linking school and work: Roles for standards and assessment. San Francisco, CA: Jossey-Bass Publishers.

Rifkin, J. (1995). The end of work: The decline of the global labor force and the dawn of the post-market era. New York, NY: G. P. Putnam's Sons.

Rogoff, B. (1990). Apprenticeship in thinking: Cognitive development in social context. New York, NY: Oxford University Press.

Rogoff, B., & Lave, J. (1984). Everyday cognition: Its development in social context. Cambridge, MA: Harvard University Press.

Salomon, G., & Perkins, D. N. (1998). Individual and social aspects of learning. In P. D. Pearson & Iran-Jejad, A. (Eds.), Review of research in education, Volume 23 (pp. 1-24). Washington, DC: American Educational Research Association.

Schank, R. (1997). Virtual learning: A revolutionary approach to building a highly skilled workforce. New York, NY: McGraw-Hill.

Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999). Learning by doing. In C. M. Reigeluth (Ed.), Instructional-design theories and models, Volume II (pp. 161-181). Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.

Schank, R. C., & Cleary, C. (1995). Engines for education. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Schiller, D. (1999). Digital Capitalism: Networking the global market system. Cambridge, MA. The MIT Press.

Schmidt, B. J., & Kirby, M. S. (1995). Technology and the development of critical thinking skills. In N. J. Groneman and K. C. Kaser (Eds.), Technology in the classroom. National Business Education Yearbook, No. 33, Reston, VA: National Business Education Association., pp. 32-39.

Scribner, S. (1986). Thinking in action: Some characteristics of practical thought. In R. J. Sternberg & R. K. Wagner (Eds.), Practical intelligence: Nature and origins of competence in the everyday world (pp. 13-50). New York, NY: Cambridge University Press.

Scribner, S. (1997). Thinking in action: Some characteristics of practical thought. In E. Tobach, R. J. Falmagne, M. B. Parlee, L. M. W. Martin, & A. S. Kapelman (Eds.), Mind and social practice: Selected writing of Sylvia Scribner (pp. 319-337). New York, NY: Cambridge University Press.

Secretary's Commission on Achieving Necessary Skills. (1991). What work requires of schools: A SCANS report for America 2000.. Washington, DC: U.S. Department of Labor.

Seels, B., & Glasgow, Z. (1990). Exercises in instructional design. Columbus, OH: Merrill Publishing Company.

Sennett, R. (1998). The corrosion of character: The personal consequences of work in the new capitalism. New York, NY: W. W. Norton & Company.

Sormunen, C., Smith, B., & Lane, J. (1996). A methodology for defining and measuring workplace competencies. NABTE Review, 23, 12-22.

Smith, B. N., Jones, C., & Lane, J. (1997). Employers' perceptions of work skills. Business Education Forum, 51(4), 11-17.

Smith, D. C., Nelson, S. J., & Mayer, K. R. (1996). Desktop publishing in the business environment. NABTE Review, 23, 28-32.

Spiro, R, J., Feltovich, P. J., Jacobson, M. J., & Coulson, R. L. (1996). Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains. [OnLine]. Available at WWW URL: http://www.itl.clumbia.edu/ilt/papers/Spiro.html

Strassmann, P. A. (1997). The squandered computer: Evaluating the business alignment of information technologies. New Canaan, CT: The Information Economics Press.

Stasz, C., & Brewer, D. J. (1999, May). Academic skills at work: Two perspectives. . MDS-1193. Berkeley, CA: RAND and the National Center for Research in Vocational Education. Available at WWW URL: http://ncrve.berkeley.edu/AllInOne/MDS-1193.html

Stasz, C., McArthur, D., Lewis, M., & Ramsey, K. (1990, November). Teaching and learning generic skills for the workplace. Berkeley, CA: RAND and the National Center for Research in Vocational Education.

Stasz, C., Ramsey, K., Eden, R., Melamid, E., & Kaganoff, T. (1996, April). Workplace skills in practice: Case studies of technical work. MDS-773. Berkeley, CA: RAND and the National Center for Research in Vocational Education.

Stasz, C., Ramsey, K., Eden, R., DaVanzo, J., Farris, H., & Lewis, M. (1992, December). Classrooms that work: Teaching generic skills in academic and vocational settings. MDS-263. Berkeley, CA: RAND and the National Center for Research in Vocational Education.

Stern, D. (1992). Institutions and incentives for developing work-related knowledge and skill. In P. S. Adler (Ed.), Technology and the future of work (pp. 149-186). New York, NY: Oxford University Press.

Taylor, H. P., & Jeffers, B. H. (1994). How computers are changing the classroom environment and the teacher's role. NABTE Review. 21, 5-9.

Thomas, R., Anderson, L., Getahun, L., & Cooke, B. (1992, December). Teaching for transfer of learning. MDS-257. Berkeley, CA: National Center for Research in Vocational Education.

Tolman, C. W. (1999). Society versus context in individual development: Does theory make a difference. In Y. Engestrom, R. Miettinen, & R Punamaki (Eds.), Perspectives on activity theory (pp. 70-86). New York, NY: Cambridge University Press.

Van der Meij, H. (1992). A critical assessment of the minimalist approach to documentation. In Conference proceedings of the 10th Annual International Conference on Systems Documentation (SIGDOC92), October, Ottawa, pp. 7-17. New York, NY: Association for Computing Machinery.

Van der Meij, H. (1992). A critical assessment of minimalist approach to documentation. SIGDOC '92, ACM 089791-533-X/92/0010/0007, pp. 7-17.

Van der Meij, H., & Carroll, J. M. (1998). Principles and heuristics for designing minimalist instruction. In J. M. Carroll, (Ed.), Minimalism beyond the Nurnberg funnel. Cambridge, MA: The MIT Press, pp. 19-53.

Vawdrey, C. (1997). Principles of learning. In H. R. Perreault (Ed.), Classroom strategies: The methodology of business education. National Business Education Yearbook, No. 34. Reston, VA: National Business Education Association. pp. 64-73.

Vickers, M. (1996). Working to learn. Cambridge, MA: TERC.

Webler M. (1992). Computer anxiety in microcomputer applications courses. Business Education Forum, 42(30-32.

Wiggs, L. H., & Huter, L. (1995). Identification of factors that contribute to students' learning microcomputer applications. The Delta Pi Epsilon Journal, 37(4), 204-216.

Wiggs, L. H., & Erthal, M. (1998). Microcomputer applications: A unique challenge for educators. Office Systems Research Journal, 16(1), 15-20.

Williams, A. S., Jr. (1996). An analysis of an achievement test for word processing for selected Virginia high school business computer applications students. The Delta Pi Epsilon Journal, 38(2), 100-115.

Winograd, T., & Flores, F. (1987). Understanding computers and cognition: A new foundation for design. Reading, MA: Addison-Wesley Publishing Company, Inc.

Wirth, A. G. (1992). Education and work for the year 2000: Choices we face. San Francisco, CA: Jossey-Bass Publishers.

Wisconsin Technical College System. (1998, June). Office systems technology core curriculum revision project (Project No. 10-004-112-138). Madison, WI: Wisconsin Technical College System.

Zemke, R., & Kramlinger, T. (1982). Figuring things out: A trainer's guide to needs and task analysis. Reading, MA: Addison-Wesley Publishing Company, Inc.

Zhao, J. J. (1996). Recommended computer end-user skills for business students by Fortune 500 human resource executives. The Delta Pi Epsilon Journal, 38(3), 155-166.

Zhao, J. J., Ray, C. M., Dye, L. J., & David, R. (1998). Recommended computer end-user skills for business students by Inc. 500 executives and office systems educators. OSRA Systems Research Journal, 16(1), 1-8.

Appendix

Appendix Item Page

Letter of Approval to Use Human Subjects in Research

Consent to Participate in a Research Study

Developing Employment-Related Technology Skills
National Center for Research in Vocational Education
University of Minnesota-Twin Cities
Judith J. Lambrecht, Ph.D.
Principal Investigator
(612) 626-1256

You are invited to participate in a research study of how employment-related technology skills are taught in secondary and postsecondary education programs. You were selected as a possible participant because the program that you are currently associated with was identified as an "exemplary" program by educators in your state. We ask that you read this form and ask any questions you may have before agreeing to be a participant in the study.

This study is conducted by Judith J. Lambrecht, Ph.D. with the support of the University of Minnesota and the National Center for Research in Vocational Education.

Background Information:

The purpose of this study is to examine the content of education programs that prepare students for employment in business-related jobs that involve the use of computing applications. Specifically, we want to ask you about the types of business technology applications taught in your program, the instructional materials used to teach business applications, how the instructor approaches the class, and your judgment of how effective the instructional practices are in developing employment-related technology skills.

Procedures

If you agree to be in this study, we would like to set up a mutually agreeable time to interview you, no more than two times, about different aspects of your experience with your program. Each interview will last for approximately one hour. The interviewer will note your answers on paper and, with your permission, record the interview so that we may listen to it again at a later time.

Risks and Benefits of Being in the Study

Risks: You may feel uncomfortable talking about the successes and difficulties of your learning experiences, teaching practices, or business operations. You may wonder if anything that you say in the interviews may be reported back to the school staff. However, because no names will appear on the interviewer's notes, and because the interviewer will respect your privacy and the confidentiality of your interview responses, there is little risk in your participation.

Benefits: Your school is part of a larger study of computer-related education practices, and your participation will help researchers understand the educational practices that benefit students both in the classroom and in their transition to the workforce. This understanding will be used to develop guidelines to use for business computing program development.

Costs: There are no costs to you except for your time for the interview.

Confidentiality: The records of this study will be kept private. In any sort of report we publish, we will not include any information that will make it possible to identify a school or an individual participant. Research records will be kept in a locked file; only the researchers will have access to the records. Information from your interview will not be shared with other teachers, students, or employers.

Voluntary Nature of the Study: Your decision whether or not to participate will not affect your current or future relations with your school, the University of Minnesota or with the organization conducting this research, the National Center for Research in Vocational Education. If you decide to participate, you are free to withdraw at any time without affecting those relationships.

Contacts and Questions: The researchers conducting this study are Judith J. Lambrecht, Ph.D. and Kim Ballard, M.Ed. You may ask any questions you have now. If you have questions later, you may contact them at (612) 626-1256.

You will be given a copy of this form to keep for your records.

Statement of Consent

I have read the above information. I have asked questions and received answers. I consent to participate in the study.

Signature____ Date_____________

Signature of parent or guardian_______________________________ Date_____________

Signature of Investigator____________________________________ Date_____________

Study Questions - Related Themes

A Needs/background for successful course participation or internship success
From Students and Teachers

1	Accounting Skills
2	Communication/Written English
3	Keyboarding
4	Math skills
5	Motivation/Reason to Be Here
6	No Business Content
7	No Communication/Written English
8	Not keyboarding
9	Software Skills
10	Spanish Language
11	Special Terminology (Business Law, Medical)
106	Prior Experience (Code number moved; intentionally out of order)

A Needs/background for employment
From Employers

101	Ability to Reason and Make Decisions
102	Advanced Technology
103	Broader View
104	Communication/Written English
105	Current technology

B Student or teacher expectations of program in relation to employment

36	Business Requirements
37	Computer Requirements
38	(No used; code omitted)
39	Friends/Other contacts
40	Mailability Standards
41	Newspaper Ads
42	Oral Communications
43	Team Building
44	Written Communications

Study Questions - Related Themes
(Continued)

C Problems

12	Communication/Written English
13	Computer Phobia
14	Customers on the Job
15	Feedback
16	Independent Decision Making/Working Independently
17	Instructional materials
18	Lack of Challenge
19	Math Phobia
20	Old Software
21	Operating System/File Mgmt
22	Personal Situations
23	Socializing on the Job
24	Special Formatting
25	Special Terminology
26	Specific Software Features
27	Stresses on the Job
28	Working Independently

D Program issues

29	Academic "Turf"
30	Academic Credit
31	Certificate/Program Uniformity
32	Gender Roles with Technology
33	Program Identity
34	Qualified Instructors
35	Software/Hardware Upgrades

E Competencies best learned in school

76	Attitude/Work Ethic
77	Communications/Written English
78	Computer Skills
79	Copying/Faxing
80	Filing/Records Mgmt
81	Keyboarding
82	Mailing Procedures
83	Teamwork
84	Telephone/Voice Mail

Study Questions - Related Themes
(Continued)

F Competencies best learned on the job

85	Accepting Work Assignments
86	Attention to Detail/Task Orientation
87	Business Environment/Motives/Profit
88	Confidentiality
89	Dependability
90	Flexibility
91	Handling Interruptions
92	How to Get Assistance
93	Initiative
94	Interpreting Expectations/Priorities
95	Specific Computer Systems
96	Specific Software
97	Specific Work Procedures
98	Teamwork
99	Telephone/Voice Mail
100	Working with Customers/Clients

G Effective teaching practices

45	Business Focus/"Real-world" Focus
46	Closed-book Exams
47	Computer in Several Courses/Articulation of Courses
48	Concept/Theory exams
49	Course Accommodation (different types of students in same course)
50	Course Diversification (different type of course for different student audience)
51	Employer/Work Involvement
52	Faculty Cohesion
53	Feedback/Constant Evaluation
38	Flexibility - Student Focus
54	Independent Practice/Self-Pacing
55	Instructional Materials
56	Integrated Software (Item omitted; merged with 45, 47, or 65)
57	No Concept/Theory Exams
58	Open-book Exams
59	Open-ended Problems/Exercises
60	Oral Presentations
61	Other Students
62	Performance Exams

63 Portfolio/Resumes/Letters/Work Compilation

Study Questions - Related Themes
(Continued)

G Effective teaching practices (Continued)

64 School-Wide Projects
65 Simulations
66 Software "Help" Screens
67 Specializations/Electives
68 Structure/Schedule for Students
69 Student Focus/Student-Teacher Contact
70 Student Groups/Teams
71 Support - from Administration
72 Support--Technical
73 Support--Tutorial
74 Teacher Demonstration
75 Whole Group Discussion

Table 6
Interview Themes
Total Times Mentioned
Sorted by Students

Themes No.	Theme	Students N=19	Teachers N=16	Employers N=11
54	Independent practice/self-pacing	17	11	1
26	Specific software features	15	11	0
69	Student focus/student-teacher contact	15	15	2
36	Business requirements	13	10	0
37	Computer Requirements	13	5	1
55	Instructional materials	13	12	1
61	Other Students	9	5	0
3	Keyboarding	8	8	0
53	Feedback/constant evaluation	8	8	0
59	Open-ended problems/exercises	8	8	0
51	Employer/work involvement	7	2	3
65	Simulations	7	3	0
68	Structure/schedule for students	7	10	0
70	Student groups/teams	7	5	1
45	Business focus/"real-world" focus	6	11	1
47	Computer in several courses	6	3	1
74	Teacher demonstration	6	8	0
40	Mailability Standards	5	2	0
106	Prior Experience	5	1	0
1	Accounting skills	4	2	0
13	Computer phobia	4	5	3
9	Software skills	3	9	0
41	Newspaper Ads	3	1	0
64	School-wide projects	3	1	0
97	Specific work procedures	3	0	7
2	Communication/Written English	2	4	0
4	Math skills	2	2	0
11	Special terminology (business law, medical)	2	2	0

Table 6
(Continued)
Interview Themes
Total Times Mentioned
Sorted by Students

Themes No.	**Theme**	**Students** N=19	**Teachers** N=16	**Employers** N=11
20	Old software	2	0	0
22	Personal situations	2	3	2
25	Special terminology	2	0	0
29	Academic “Turf”	2	4	0
44	Written Communications	2	1	0
60	Oral presentations	2	4	1
66	Software “Help” screens	2	0	0
67	Specializations/electives	2	2	0
73	Support--tutorial	2	4	1
76	Attitude/work ethic	2	3	8
78	Computer skills	2	0	10
92	How to get assistance	2	0	7
95	Specific computer systems	2	0	4
96	Specific software	2	1	7
98	Teamwork	2	0	3
100	Working with customers/clients	2	0	1
5	Motivation/Reason to be here	1	3	0
6	No Business Content	1	3	0
16	Independent Decision Making/Working Independently	1	2	0
18	Lack of challenge	1	0	0
19	Math phobia	1	1	1
34	Qualified Instructors	1	1	0
38	Flexibility - Student Focus	1	2	0
39	Friends/Other contacts	1	0	0
42	Oral Communications	1	0	0
50	Course diversification (different type of course for different student audience)	1	9	1
58	Open-book exams	1	2	0
63	Portfolio/resumes/letters/work compilation	1	1	0

Table 6 (Continued) Interview Themes Total Times Mentioned Sorted by Students				
Themes No.	**Theme**	**Students** N=19	**Teachers** N=16	**Employers** N=11
72	Support--technical	1	3	0
80	Filing/records mgmt	1	0	1
82	Mailing procedures	1	0	2
84	Telephone/voice mail	1	0	4
86	Attention to detail/task orientation	1	0	4
89	Dependability	1	0	6
90	Flexibility	1	0	1
94	Interpreting expectations/priorities	1	0	7
7	No Communication/Written English	0	1	0
8	Not keyboarding	0	2	0
10	Spanish Language	0	1	0
12	Communication/Written English	0	1	0
14	Customers on the job	0	2	0
15	Feedback	0	1	0
17	Instructional materials	0	0	0
21	Operating System/File Mgmt	0	0	0
23	Socializing on the job	0	1	0
24	Special Formatting	0	0	0
27	Stress on the job	0	1	0
28	Working independently	0	0	0
30	Academic credit	0	5	0
31	Certificate/Program Uniformity	0	1	0
32	Gender Roles with Technology	0	2	1
33	Program Identity	0	4	1
35	Software/hardware upgrades	0	1	0
43	Team Building	0	0	0
46	Closed-book exams	0	1	0
48	Concept/theory exams	0	4	0
49	Course accommodation (different types of students in same course)	0	8	0

Table 6 (Continued) Interview Themes Total Times Mentioned Sorted by Students				
Themes No.	**Theme**	**Students** N=19	**Teachers** N=16	**Employers** N=11
52	Faculty cohesion	0	6	0
56	Integrated software	0	0	0
57	No concept/theory exams	0	3	0
62	Performance exams	0	5	0
71	Support - from Administration	0	1	0
75	Whole group discussion	0	5	1
77	Communications/written English	0	1	6
79	Copying/faxing	0	0	2
81	Keyboarding	0	0	6
83	Teamwork	0	0	1
85	Accepting work assignments	0	0	4
87	Business environment/motives/profit	0	0	5
88	Confidentiality	0	0	2
91	Handling interruptions	0	0	1
93	Initiative	0	0	4
99	Telephone/voice mail	0	0	2
101	Ability to reason and make decisions	0	1	1
102	Advanced technology	0	0	1
103	Broader view	0	0	1
104	Communication/written English	0	1	3
105	Current technology	0	1	1
	Total number of different themes	64		
	Total number of different themes with responses > 2	25		

Table 7 Interview Themes Total Times Mentioned Sorted by Teachers				
Themes No.	**Theme**	**Teachers** N=16	**Employers** N=11	**Students** N=19
69	Student focus/student-teacher contact	15	2	15
55	Instructional materials	12	1	13
26	Specific software features	11	0	15
45	Business focus/"real-world" focus	11	1	6
54	Independent practice/self-pacing	11	1	17
36	Business requirements	10	0	13
68	Structure/schedule for students	10	0	7
9	Software skills	9	0	3
50	Course diversification (different type of course for different student audience)	9	1	1
3	Keyboarding	8	0	8
49	Course accommodation (different types of students in same course)	8	0	0
53	Feedback/constant evaluation	8	0	8
59	Open-ended Problems/exercises	8	0	8
74	Teacher demonstration	8	0	6
52	Faculty cohesion	6	0	0
13	Computer phobia	5	3	4
30	Academic credit	5	0	0
37	Computer requirements	5	1	13
61	Other students	5	0	9
62	Performance exams	5	0	0
70	Student groups/teams	5	1	7
75	Whole group discussion	5	1	0
2	Communication/written English	4	0	2
29	Academic “turf”	4	0	2
33	Program Identity	4	1	0
48	Concept/theory exams	4	0	0
60	Oral presentations	4	1	2
73	Support--tutorial	4	1	2
5	Motivation/reason to be here	3	0	1

Table 7
(Continued)
Interview Themes
Total Times Mentioned
Sorted by Teachers

Themes No.	Theme	Teachers N=16	Employers N=11	Students N=19
6	No business content	3	0	1
22	Personal situations	3	2	2
47	Computer in several courses	3	1	6
57	No concept/theory exams	3	0	0
65	Simulations	3	0	7
72	Support--technical	3	0	1
76	Attitude/work ethic	3	8	2
1	Accounting skills	2	0	4
4	Math skills	2	0	2
8	Not keyboarding	2	0	0
11	Special terminology (business law, medical)	2	0	2
14	Customers on the job	2	0	0
16	Independent decision making/working independently	2	0	1
32	Gender Roles with technology	2	1	0
38	Flexibility - student focus	2	0	1
40	Mailability Standards	2	0	5
51	Employer/work involvement	2	3	7
58	Open-book exams	2	0	1
67	Specializations/electives	2	0	2
7	No Communication/written English	1	0	0
10	Spanish Language	1	0	0
12	Communication/written English	1	0	0
15	Feedback	1	0	0
19	Math phobia	1	1	1
23	Socializing on the job	1	0	0
27	Stress on the job	1	0	0
31	Certificate/program uniformity	1	0	0
34	Qualified instructors	1	0	1

Table 7
(Continued)
Interview Themes
Total Times Mentioned
Sorted by Teachers

Themes No.	**Theme**	**Teachers** N=16	**Employers** N=11	**Students** N=19
35	Software/hardware upgrades	1	0	0
41	Newspaper Ads	1	0	3
44	Written Communications	1	0	2
46	Closed-book exams	1	0	0
63	Portfolio/resumes/letters/work compilation	1	0	1
64	School-wide projects	1	0	3
71	Support - from Administration	1	0	0
77	Communications/written English	1	6	0
96	Specific software	1	7	2
101	Ability to reason and make decisions	1	1	0
104	Communication/written English	1	3	0
105	Current technology	1	1	0
106	Prior Experience	1	0	5
17	Instructional materials	0	0	0
18	Lack of challenge	0	0	1
20	Old software	0	0	2
21	Operating System/file mgmt	0	0	0
24	Special Formatting	0	0	0
25	Special terminology	0	0	2
28	Working independently	0	0	0
39	Friends/other contacts	0	0	1
42	Oral communications	0	0	1
43	Team building	0	0	0
56	Integrated software	0	0	0
66	Software "help" screens	0	0	2
78	Computer skills	0	10	2
79	Copying/faxing	0	2	0
80	Filing/records mgmt	0	1	1
81	Keyboarding	0	6	0

Table 7
(Continued)
Interview Themes
Total Times Mentioned
Sorted by Teachers

Themes No.	**Theme**	**Teachers** N=16	**Employers** N=11	**Students** N=19
82	Mailing procedures	0	2	1
83	Teamwork	0	1	0
84	Telephone/voice mail	0	4	1
85	Accepting work assignments	0	4	0
86	Attention to detail/task orientation	0	4	1
87	Business environment/motives/profit	0	5	0
88	Confidentiality	0	2	0
89	Dependability	0	6	1
90	Flexibility	0	1	1
91	Handling interruptions	0	1	0
92	How to get assistance	0	7	2
93	Initiative	0	4	0
94	Interpreting expectations/priorities	0	7	1
95	Specific computer systems	0	4	2
97	Specific work procedures	0	7	3
98	Teamwork	0	3	2
99	Telephone/voice mail	0	2	0
100	Working with customers/clients	0	1	2
102	Advanced technology	0	1	0
103	Broader view	0	1	0
	Count number of different themes	70		
	Count number of different these with responses > 2	36		

Table 8
Interview Themes
Total Times Mentioned
Sorted by Employers

Themes No.	Theme	Employers N=11	Teachers N=16	Students N=19
78	Computer skills	10	0	2
76	Attitude/work ethic	8	3	2
92	How to get assistance	7	0	2
94	Interpreting expectations/priorities	7	0	1
96	Specific software	7	1	2
97	Specific work procedures	7	0	3
77	Communications/written English	6	1	0
81	Keyboarding	6	0	0
89	Dependability	6	0	1
87	Business environment/motives/profit	5	0	0
84	Telephone/voice mail	4	0	1
85	Accepting work assignments	4	0	0
86	Attention to detail/task orientation	4	0	1
93	Initiative	4	0	0
95	Specific computer systems	4	0	2
13	Computer phobia	3	5	4
51	Employer/work involvement	3	2	7
98	Teamwork	3	0	2
104	Communication/written English	3	1	0
22	Personal situations	2	3	2
69	Student focus/student-teacher contact	2	15	15
79	Copying/faxing	2	0	0
82	Mailing procedures	2	0	1
88	Confidentiality	2	0	0
99	Telephone/voice mail	2	0	0
19	Math phobia	1	1	1
32	Gender roles with technology	1	2	0
33	Program identity	1	4	0
37	Computer requirements	1	5	13
45	Business focus/"real-world" focus	1	11	6
47	Computer in several courses	1	3	6

Table 8 (Continued) Interview Themes Total Times Mentioned Sorted by Employers				
Themes No.	**Theme**	**Employers** N=11	**Teachers** N=16	**Students** N=19
50	Course diversification (different type of course for different student audience)	1	9	1
54	Independent practice/self-pacing	1	11	17
55	Instructional materials	1	12	13
60	Oral presentations	1	4	2
70	Student groups/teams	1	5	7
73	Support--tutorial	1	4	2
75	Whole group discussion	1	5	0
80	Filing/records mgmt	1	0	1
83	Teamwork	1	0	0
90	Flexibility	1	0	1
91	Handling interruptions	1	0	0
100	Working with customers/clients	1	0	2
101	Ability to reason and make decisions	1	1	0
102	Advanced technology	1	0	0
103	Broader view	1	0	0
105	Current technology	1	1	0
1	Accounting skills	0	2	4
2	Communication/written English	0	4	2
3	Keyboarding	0	8	8
4	Math skills	0	2	2
5	Motivation/reason to be here	0	3	1
6	No business content	0	3	1
7	No communication/written English	0	1	0
8	Not keyboarding	0	2	0
9	Software skills	0	9	3
10	Spanish language	0	1	0
11	Special terminology (business law, medical)	0	2	2
12	Communication/written English	0	1	0
14	Customers on the job	0	2	0
15	Feedback	0	1	0

Table 8
(Continued)
Interview Themes
Total Times Mentioned
Sorted by Employers

Themes No.	Theme	Employers N=11	Teachers N=16	Students N=19
16	Independent decision making/working independently	0	2	1
17	Instructional materials	0	0	0
18	Lack of challenge	0	0	1
20	Old software	0	0	2
21	Operating system/file mgmt	0	0	0
23	Socializing on the job	0	1	0
24	Special formatting	0	0	0
25	Special terminology	0	0	2
26	Specific software features	0	11	15
27	Stress on the job	0	1	0
28	Working independently	0	0	0
29	Academic “turf”	0	4	2
30	Academic credit	0	5	0
31	Certificate/program uniformity	0	1	0
34	Qualified instructors	0	1	1
35	Software/hardware upgrades	0	1	0
36	Business requirements	0	10	13
38	Flexibility - student focus	0	2	1
39	Friends/other contacts	0	0	1
40	Mailability standards	0	2	5
41	Newspaper ads	0	1	3
42	Oral communications	0	0	1
43	Team building	0	0	0
44	Written communications	0	1	2
46	Closed-book exams	0	1	0
48	Concept/theory exams	0	4	0
49	Course accommodation (different types of students in same course)	0	8	0
52	Faculty cohesion	0	6	0
53	Feedback/constant evaluation	0	8	8
56	Integrated software	0	0	0

Table 8 **(Continued)** **Interview Themes** **Total Times Mentioned** **Sorted by Employers**				
Themes No.	**Theme**	**Employers** N=11	**Teachers** N=16	**Students** N=19
57	No concept/theory exams	0	3	0
58	Open-book exams	0	2	1
59	Open-ended problems/exercises	0	8	8
61	Other Students	0	5	9
62	Performance exams	0	5	0
63	Portfolio/resumes/letters/work compilation	0	1	1
64	School-wide projects	0	1	3
65	Simulations	0	3	7
66	Software "help" screens	0	0	2
67	Specializations/electives	0	2	2
68	Structure/schedule for students	0	10	7
71	Support - from administration	0	1	0
72	Support--technical	0	3	1
74	Teacher demonstration	0	8	6
106	Prior experience	0	1	5
	Count number of different themes	47		
	Count number of different these with responses > 2	25		