Enter resource planning Systems and their impact on Development and Training: A study of Instructional Methods in North America

Sharon L. Bassette  
St. John Fisher College

Wendy L. Carney  
St. John Fisher College

Lori A. Laney  
St. John Fisher College

Follow this and additional works at: https://fisherpub.sjfc.edu/education_ETD_masters

Recommended Citation

Please note that the Recommended Citation provides general citation information and may not be appropriate for your discipline. To receive help in creating a citation based on your discipline, please visit http://libguides.sjfc.edu/citations.

This document is posted at https://fisherpub.sjfc.edu/education_ETD_masters/109 and is brought to you for free and open access by Fisher Digital Publications at St. John Fisher College. For more information, please contact fisherpub@sjfc.edu.

Abstract
One of the most important issues facing the modern world today is the Year 2000 problem. One of the greatest impacts of this problem is experienced by legacy computer systems. These are the database systems that run our businesses. Legacy systems are operated by segregated software packages that may or may not be able to "communicate" to each other. With the globalization of the economy, business computer systems need to be able to communicate with each other. This type of situation is what enterprise resource planning software is designed to solve. Enterprise Resource Planning (ERP) systems provide a common, consistent system to capture data organization-wide without redundancies. ERP is defined as a software management system that integrates all facets of the business, including planning, manufacturing, sales, and marketing. In addition to integrating the information across functions, also provides a set of tools for planning and monitoring the organizational functions and ensuring progress towards a common organizational goal (Sudhakar, 1998). As ERP methodology has become more popular, software applications have emerged to help business managers implement ERP. The purpose of this pilot study is to identify development times and delivery methods for high-level computer technology training. This has become an issue for Human Resource Development (HRD) because of the Year 2000 situation and its resulting issues, which have compelled businesses around North America to implement these ERP systems. After circulating a survey for almost seven month, the data collection period ended in March 1999. The response rate of 47 returned surveys from 63 requests calculates to 74.6%. A total of 34, or 72%, of respondents developed training programs for high-level computer technology.

Document Type
Thesis

Degree Name
MS in Human Resource Development

Department
Education

First Supervisor
Marilynn N. Butler

Second Supervisor
Jane A. LeClair

Third Supervisor
Taylor Brown

Subject Categories
Education

This thesis is available at Fisher Digital Publications: https://fisherpub.sjfc.edu/education_ETD_masters/109

Sharon L. Bassette
St. John Fisher College

Wendy L. Carney
St. John Fisher College

Lori A. Laney
St. John Fisher College

Follow this and additional works at: http://fisherpub.sjfc.edu/education_ETD_masters

Part of the Education Commons

Recommended Citation

This Thesis is brought to you for free and open access by the Ralph C. Wilson, Jr. School of Education at Fisher Digital Publications. It has been accepted for inclusion in Education Masters by an authorized administrator of Fisher Digital Publications.
Enterprise Resource Planning Systems and Their Impact on Development
and Training: A Study of Instructional Methods in North America

Abstract
One of the most important issues facing the modern world today is the Year 2000 problem. One of the
greatest impacts of this problem is experienced by legacy computer systems. These are the database systems
that run our businesses. Legacy systems are operated by segregated software packages that may or may not be
able to "communicate" to each other. With the globalization of the economy, business computer systems need
to be able to communicate with each other. This type of situation is what enterprise resource planning
software is designed to solve. Enterprise Resource Planning (ERP) systems provide a common, consistent
system to capture data organization-wide without redundancies. ERP is defined as a software management
system that integrates all facets of the business, including planning, manufacturing, sales, and marketing. In
addition to integrating the information across functions, also provides a set of tools for planning and
monitoring the organizational functions and ensuring progress towards a common organizational goal
(Sudhakar, 1998). As ERP methodology has become more popular, software applications have emerged to
help business managers implement ERP. The purpose of this pilot study is to identify development times and
delivery methods for high-level computer technology training. This has become an issue for Human Resource
Development (HRD) because of the Year 2000 situation and its resulting issues, which have compelled
businesses around North America to implement these ERP systems. After circulating a survey for almost
seven month, the data collection period ended in March 1999. The response rate of 47 returned surveys from
63 requests calculates to 74.6%. A total of 34, or 72%, of respondents developed training programs for high-
level computer technology.

Degree Type
Thesis

Degree Name
MS in Human Resource Development

Department
Education

First Supervisor
Marilynn N. Butler

Second Supervisor
Jane A. LeClair

Third Supervisor
Taylor Brown

This thesis is available at Fisher Digital Publications: http://fisherpub.sjfc.edu/education_ETD_masters/109
Enterprise Resource Planning Systems and
Their Impact on Development and Training: A Study of
Instructional Methods in North America

By
Sharon L. Bassette
Wendy L. Carney
Lori A. Laney, MBA
© 1999

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May, 1999
We approve the paper of Sharon L. Bassette, Wendy L. Carney, and Lori A. Laney.
Abstract


Sharon Bassette

Wendy Carney

Lori A. Laney, MBA

Master of Science; May 1999

St. John Fisher College

Dr. Marilynn N. Butler, Advisor

One of the most important issues facing the modern world today is the Year 2000 problem. One of the greatest impacts of this problem is experienced by legacy computer systems. These are the database systems that run our businesses. Legacy systems are operated by segregated software packages that may or may not be able to "communicate" to each other. With the globalization of the economy, business computer systems need to be able to communicate with each other. This type of situation is what enterprise resource planning software is designed to solve. Enterprise Resource Planning (ERP) systems provide a common, consistent system to capture data organization-wide without redundancies. ERP is defined as a software management system that integrates all facets of the business, including planning, manufacturing, sales, and marketing. In addition to integrating the information across functions, also provides a set of tools for planning and monitoring the organizational functions and ensuring progress towards a common organizational goal (Sudhakar, 1998). As ERP methodology has become more popular, software applications have emerged to help business managers implement ERP.

The purpose of this pilot study is to identify development times and delivery methods for high-level computer technology training. This has become an issue for Human Resource Development (HRD) because of the Year 2000 situation and its resulting issues, which have compelled businesses around North America to implement these ERP systems.

After circulating a survey for almost seven months, the data collection period ended in March 1999. The response rate of 47 returned surveys from 63 requests calculates to 74.6%. A total of 34, or 72%, of respondents developed training programs for high-level computer technology.
The results of the pilot study indicate that average development ratios exhibit a large amount of variance regardless of training delivery method. The results also argue that the majority of training development time is spent on the actual development of the training materials and "Hands-on" is the most widely used training delivery method without respect to geographic location or industry classification. It was also found that most companies do not benchmark their training practices yet 78% do measure for success of training.
We grant St. John Fisher College the non-exclusive right to use this work for the College’s own purpose and to make single copies of the work available to the public on the not-for-profit basis if copies are not otherwise available.

Sharon L. Bassette, MS

Wendy L. Carney, MS

Lori A. Laney, MBA, MS
Acknowledgments

First and foremost, I want to dedicate this to my parents. Without their love, support, and dedication to education, I would not be where I am today. Thanks to Mark who nagged me, loved me, and helped me through this whole process since January 1998. Thanks to Sarah and Julie. Sarah for living with the messes I created and Julie who let me be her "weekend roommate". Finally, Wendy and Lori deserve a world of thanks for putting up with me as their sickly group member!

- Sharon L. Bassette

First I would like to thank my group members, Lori and Sharon, for all they contributed and for sticking through this project with me. I would also like to thank Deb and Julie for getting this project off the ground with us in 501. I would especially like to thank my husband, Eric, for his love, support, and guidance and for nagging, assisting, and leaving me alone as needed.

- Wendy L. Carney

I wish to thank my parents for their love, support, and how they always make me strive for the best. I also wish to thank Mike for his love, support, and ability to keep me grounded. Thanks to my group members, past and present, for their hard work and dedication to this project!

- Lori A. Laney

We would all like to thank our advisor, Dr. Marilynn Butler, for her continued guidance, patience, and ability to get us motivated and keep us motivated on this project. We would also like to thank Eric Carney for his technological expertise.

- "The Graduate Human Resource Development Program Project Group"
<table>
<thead>
<tr>
<th>Chapter One</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>13</td>
</tr>
<tr>
<td>Purpose of the pilot study</td>
<td>14</td>
</tr>
<tr>
<td>Preview of Paper</td>
<td>14</td>
</tr>
<tr>
<td>Glossary of Terms</td>
<td>15</td>
</tr>
<tr>
<td>Background of the pilot study</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter Two</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Review of Related Literature</td>
<td>19</td>
</tr>
<tr>
<td>Year 2000: Y2K</td>
<td>19</td>
</tr>
<tr>
<td>Date Mathematics</td>
<td>20</td>
</tr>
<tr>
<td>Valid Date Determination Systems</td>
<td>22</td>
</tr>
<tr>
<td>Computers That Stop Functioning</td>
<td>22</td>
</tr>
<tr>
<td>Historical Background: Where It All Began</td>
<td>23</td>
</tr>
<tr>
<td>COBOL Language: Past, Present, and Future</td>
<td>25</td>
</tr>
<tr>
<td>History</td>
<td>26</td>
</tr>
<tr>
<td>Breakdown of COBOL Program Components</td>
<td>26</td>
</tr>
<tr>
<td>Versions</td>
<td>27</td>
</tr>
<tr>
<td>COBOL’s Present State</td>
<td>27</td>
</tr>
<tr>
<td>The Future of COBOL</td>
<td>28</td>
</tr>
<tr>
<td>Embedded Systems: Another Y2K Issue</td>
<td>30</td>
</tr>
<tr>
<td>What is ERP?</td>
<td>32</td>
</tr>
</tbody>
</table>

<p>| Definition                                                                | 32|</p>
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Based Training</td>
<td>52</td>
</tr>
<tr>
<td>Asynchronous Computer Training</td>
<td>53</td>
</tr>
<tr>
<td>Electronic Performance Support Systems</td>
<td>54</td>
</tr>
<tr>
<td>Instructor-Led Training</td>
<td>56</td>
</tr>
<tr>
<td>Coaching</td>
<td>56</td>
</tr>
<tr>
<td>Combining Different Types of Training</td>
<td>57</td>
</tr>
<tr>
<td>Costs of Training</td>
<td>59</td>
</tr>
<tr>
<td>Development Time of Technical Training</td>
<td>60</td>
</tr>
<tr>
<td>What Does the Ratio Depend On?</td>
<td>60</td>
</tr>
<tr>
<td>Varying Ratios</td>
<td>61</td>
</tr>
<tr>
<td>Instructional Design Project Management Model</td>
<td>61</td>
</tr>
<tr>
<td>The Amount of Non-Writing Time</td>
<td>62</td>
</tr>
<tr>
<td>Deliverable Variances</td>
<td>62</td>
</tr>
<tr>
<td>Documentation of the History and Development Process</td>
<td>63</td>
</tr>
<tr>
<td>Summary</td>
<td>65</td>
</tr>
<tr>
<td>Chapter Three</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>---</td>
</tr>
<tr>
<td>Methodology</td>
<td>66</td>
</tr>
<tr>
<td>Survey Development</td>
<td>66</td>
</tr>
<tr>
<td>Sample</td>
<td>68</td>
</tr>
<tr>
<td>Data Collection</td>
<td>68</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>70</td>
</tr>
<tr>
<td>Conclusion</td>
<td>71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter Four</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>72</td>
</tr>
<tr>
<td>Results</td>
<td>72</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>73</td>
</tr>
<tr>
<td>Summary of Measurement of Successful Training</td>
<td>80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter Five</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Conclusions</td>
<td>87</td>
</tr>
<tr>
<td>Implications</td>
<td>88</td>
</tr>
<tr>
<td>Questions for Further Research</td>
<td>89</td>
</tr>
</tbody>
</table>

<p>| References   | 91 |
| Appendices   | 97 |</p>
<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix A</td>
<td>ERP Market Share By Vendor</td>
<td>97</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Evolution of ERP</td>
<td>98</td>
</tr>
<tr>
<td>Appendix C</td>
<td>Copy of Survey</td>
<td>99</td>
</tr>
<tr>
<td>Appendix D</td>
<td>Copy of Invitation Letter</td>
<td>100</td>
</tr>
<tr>
<td>Appendix E</td>
<td>Follow-up Letter</td>
<td>101</td>
</tr>
<tr>
<td>Appendix F</td>
<td>Geographical Location Summary</td>
<td>102</td>
</tr>
</tbody>
</table>
Table of Tables

Table 1: Average Development Ratios by Training Delivery Method 74

Table 2: Percentage of Total Development Time 76
by Development Component

Table 3: Percentage of Total Training Time by Training Delivery Method 77

Table 4: Most Common Training Delivery Methods 79
by Geographic Location

Table 5: Most Common Training Delivery Methods 80
by Industry Classification
Chapter One

Introduction

The Year 2000 (Y2K) problem has the potential to affect all computer systems sooner or later. For example, if you are riding in an elevator on December 31, 1999 and the clock strikes midnight will the elevator fall to the ground, probably not. Then what is all this talk about the Year 2000, and how is it supposed to disable the entire world? Since the elevator probably will not fall to the ground on December 31, 1999, then what will happen to the elevator because of the Y2K problem? Experts say that everyone will experience inconveniences from the Y2K problem (Stanglin & Amhad, 1998). If the Y2K problem has not been resolved by the Year 2000, there is a large possibility that this elevator would not have had the proper maintenance done on it when it was required. Elevator maintenance is determined by a computer program that is supposed to keep track of its operations and automatically dispatch maintenance personnel to do the proper work at the proper time. Therefore, it is possible that this elevator would come crashing down later on in the year, perhaps in July 2000, depending on the computer program. The Y2K problem will not happen at the stroke of midnight on December 31, 1999, but its cumulative effects will begin, and may cause serious damage to all aspects of society (Simpson, 1998).

In dealing with the Y2K problem, companies are faced with choosing one of two options: fix the current systems or replace the current systems. Most are choosing to replace the current systems with Enterprise Resource Planning (ERP) Systems. Replacements, or implementations as they are referred to in this paper,
create new methods of conducting business and operating day to day events of an organization. It requires the acceptance, understanding, and active participation of every member of the organization. All employees must be trained on the system so that when the company's old system is shut down there is a successful conversion to the new the ERP system. To accomplish the task of training all employees, new programs and methods of instruction delivery need to be created. Due to the high importance of successful training, companies are struggling with finding the right combinations of methods to ensure that success takes place.

Purpose of the Pilot Study

The purpose of this pilot study is to identify development times and delivery methods for high-level computer technology training programs with respect to Enterprise Resource Planning Systems in North America. This paper provides a five-fold presentation that brings this topic into focus.

Preview of Paper

To introduce this pilot study, Chapter One opens with a glossary of terminology. In addition, a background discussion of the problem is presented. Chapter Two offers a literature review that describes the research that has been conducted in this area. In this review 11 main categories are examined. This examination provides the scope of the problem. Chapter Three discusses the methodology followed to conduct the pilot study. Chapter Four presents the results of the pilot study. Chapter Five examines what the results of the pilot study mean, and will explore possible recommendations for further research.
Glossary of Terms

This section identifies and defines a list of terms used throughout this paper.

**ERP**

ERP is short for *enterprise resource planning*, a business management system that integrates all facets of the business, including planning, manufacturing, sales, and marketing. In the midst of the massive globalization of business and industry, information technology systems must be able to communicate with one another. As the ERP methodology has become more popular, software applications have also emerged to help business managers implement ERP (http://webopedia.internet.com, 1998).

**Y2K**

Y2K is an abbreviation for the *Year 2000*. The issue at hand is how computers and applications will manage data calculations when dates begin to include years in the 2000's. There are three issues that are central to the Y2K problem. These are date mathematics, systems that check the date for some purpose to determine if a valid date is being used, and computers that will stop functioning altogether (Randall, 1997; White, 1998).

**Embedded Systems**

Beyond the threat of the date being miscalculated in the Year 2000, there is an even more serious concern involving "embedded chips" or embedded systems. Embedded chips are individual microchips that have computer codes
“burned” right into them which are used to control the operation of equipment, machinery, or even plants (Vector International, 1998).

**Legacy Systems**

Legacy systems are the computer systems that many companies currently have or had in the past before they implemented a company-wide ERP system. Legacy systems are computer systems run by segregated software packages that may or may not be able to “talk” to each other.

**High-level Technical Training**

High-level technical training refers to training for any of the ERP systems in the business industry such as SAP, Oracle, PeopleSoft, or Windows NT systems. High-level computer technology training can be defined as training for both end-users and Information Technology professionals on computer applications and systems (Anonymous, 1998). The system that the most data has been collected on for this pilot study is the SAP system.

With the increasing demand for the installation and utilization of ERP systems in business and industry combined with the Y2K issue, the need to conduct research for training and development around high-level computer technology training delivery is imperative.

**Technical Training Development Time and Training Delivery Methods**

Training development time refers to the amount of time and work that goes into preparing one hour of classroom training. Training delivery specifically refers to examining instructional models used to deliver the necessary training. A variety of training delivery methods have been developed and implemented to
train employees in high-level computer programs. Some of these models include computer-based training (CBT), Internet based training (IBT), asynchronous computer conferencing (ACC), instructor-led training (ILT), and coaching.

Background of the Pilot Study

The development of this pilot study began when a multinational Rochester, New York organization contacted the research team requesting that the pilot study be conducted. Realizing that all aspects of their businesses would no longer be able to function independently, their concern was with the fact that they would have to implement an ERP system very soon and would need to develop and deliver high-level technical training to all employees in the organization. Their daily operations would need to act and react as one company-wide system. This company felt that there would be a major challenge in educating and training their business entities and their employees to use their new system effectively, and therefore the research team was contacted to examine this issue in further detail.

From this initial contact, the survey was developed. The methods that were used to develop this survey include examining literature to define development time and delivery methods of training and a brainstorming session to create questions to support the literature findings. A panel of instructional designers were contacted and interviewed as well. This draft of the survey was constructed from this information. This draft was reviewed by a panel of instructional designers and trainers who worked specifically with high-level computer technology training programs. From this review, the panel made
suggestions and changes to the survey expanding it and making the questions more specific and relevant.

The survey was first developed with a global focus, but interest levels and response rates were very poor. It is speculated that the reason for the low response rate is that the countries that the organization suggested we look at had not done much to develop their own training. Canada, which had shown great interest in the survey, wasn’t even on the organization’s list of countries to contact. It was partially for this reason and, because the group felt a smaller scale focus was needed, that the survey was halted and reborn with a new target population. The new target population included organizations in North America (Mexico, Canada, and the United States).
Chapter 2
Review of Related Literature

The purpose of the research reported in this pilot study is to explore development times and delivery methods for high-level technology training programs with respect to ERP systems. This pilot study examines ERP systems and their impact on development and training within instructional methods in North America.

The chapter offers a review of related literature. To open, a discussion of the Year 2000 computer issue and its history is presented. The following component offers a discussion of ERP systems and how their implementation process affects those businesses. The final component of this chapter focuses on Year 2000 and technical training issues in relation to the different types of training and their respective development times.

The Year 2000: Y2K

One of the important issues facing today’s businesses is the Year 2000 (Y2K) problem. Y2K is an abbreviation for the Year 2000. The new millenium presents the issue of how computers and computer applications will manage data calculations when dates begin to include years in the 2000’s. This date year will affect legacy computer systems, which run today’s businesses. Legacy systems are computer systems run by segregated software packages that may or may not be able to “talk” to each other. In order for businesses to continue to operate and compete globally, their computer systems need to be able to communicate with each other.
The Y2K problem will be one of the most expensive problems in human history. In the United States, it is estimated that more than four months of effort may be needed on the part of every software professional and the cost to fix it may exceed $900 for every citizen (Jones, 1997). As the millennium approaches, vendors, consultants, and information systems professionals are encouraging companies to begin addressing computerized date problems and are offering their own year 2000 services. Solving these problems promises to be a lucrative business (DeVoe, 1995).

The driver behind these problems is how computers and computer applications will manage data calculations when dates begin to include years in the 2000's (DeVoe, 1995). There are three functions that are central to the Y2K problem: (1) date mathematics; (2) valid date determination systems (White, 1998); (3) computers that stop functioning altogether (Randall, 1997). A closer examination of each function provides greater understanding of the Y2K problems.

Date Mathematics

For years businesses have used "date mathematics" to compute many functions such as aging schedules, due dates, and past due accounts, among others. Many high technical computer systems support the use of date mathematics such as Lotus 123, Microsoft Excel, and Microsoft Access. These computer applications work by using the base year of January 1, 1900, as a starting point. The system then tracks the date and time numerically from January 1, 1900, by figuring out how much time has elapsed since that point (White,
Engineers and computer programmers chose to represent calendar dates in the six-digit format (i.e., mm/dd/yy). Example 1 offers an illustration of how this format creates a problem.

**Example 1.** To calculate the age of someone born in 1940, which was represented as “40” in the system, the computer would take the current day’s date and subtract the birth date. If this calculation was done in 1998, the computer would take “40” and subtract it from “98.” The resulting number would be “58.” This system works well until the computer reaches midnight on December 31, 1999. This is the point when the computer will see the date with “00” and interpret it as the year 1900. If the same age calculation was done when the computer showed the date “00,” the resulting age would be “-60.” Since it is impossible to have a negative age, it would mean that the individual was not born yet, which in turn would result in many problems (Simpson, 1998). The date calculations that are wrong could cause problems such as miscalculating age, retirement date, the amount of time spent on a long distance phone call, the calculation on the interest of a mortgage, and many more problems too numerous to list.

Many other computer systems use the default date of January 1, 1980. In a computer that uses this date as its default, a system shut down would cause the computer to think the date is January 1, 1980. When the century changes, the computers may not respond to the correct time and date and will constantly revert back to their January 1, 1980, default. This lack of response could severely damage automatic archiving programs, spreadsheet and database calculations.
along with file management capabilities. If the resulting chaos happened to a computer that was running a business, all accounting and invoicing functions would be inoperative (Randall, 1997).

Valid Date Determination Systems

Many computer systems are set up to determine if a valid date is being used. Example 2 illustrates what may happen with a credit card expiration date.

Example 2. The program that checks the credit card expiration date when the card is swiped asks the question “Is today’s date greater than the expiration date of the card?” If this system is not Y2K compliant, January 1, 1999 would read greater than January 1, 2000. This misreading of the date would result in the decline of the credit card.

Example 3. This example describes what may happen with a building security system using pass cards. If a security system checks to see if the current date on the card is a valid date before allowing access to a building, it will determine the date “00” is out of range. Hence, the system would shut down and all the doors would lock, allowing for no entry or exit from a building with a system like this (White, 1998).

Computers That Stop Functioning

The third possible effect of the Y2K problem is that computers may just stop functioning on January 1, 2000. This result is due to how systems that work closely with the system date have no way of interpreting that date string (Randall, 1997). A date-string is a data type, just like a number or a letter, but whose value is a date. Many of the older operating systems were programmed to recognize the
date 9/9/99 as the end of the file. In these systems, the programs will shut down even before the turn of the century (Nixon, Hargrave, Devans & Doyle LLP, 1998). This particular problem will mostly affect the mainframe community of computers (Simpson, 1998).

Mainframes that run on legacy operations which perform large and critical tasks are still in operation today in big corporations and government agencies (Randall, 1997). A legacy system is one that will not receive any further development or enhancement. A mainframe may have millions of lines of codes.

Historical Background: Where it all Began

The Y2K computer problem dates back to the 1950's and the development of COBOL, a widely used programming language that used only two digits to represent a given year. Warnings about the problem began in 1971, but were largely ignored until awareness reached critical mass in the middle 1990's (Simpson, 1998).

There are many reasons why COBOL programmers chose to represent years by using just two digits. For instance 55, for the year 1955, or 10/23/76 and 23/10/76 for October 23, 1976. Decades ago digital space was limited, computer memory was expensive, and typical punch cards were only 80 columns wide. To save precious space on these 80-column punch cards, COBOL programmers used just six numbers to represent the day's date. The date was represented by two numbers for the day, two numbers for the month and two numbers for the year.

The Y2K problem clearly stems from the idea of saving space and money. Memory for a computer used to come at a very high cost. In 1970, one megabyte
of memory cost $3.5 million. In 1980, the cost went down to $1 million. Today, it is impossible to even purchase one megabyte of memory; the minimum is usually 16 megabytes. The cost for 16 megabytes is $500 (Nixon, et al., 1998)!

Today, a Pentium computer probably has more memory and storage capacity than all the machines in a major city in the 1960’s (Simpson, 1998).

In the early days of computing, machines could only handle a very limited amount of data. Representing the dates with ten characters (mm/dd/yyyy) would have taken up more space than the six characters that everyone has grown familiar with. Every time the date would surface in a computer program, programmers gave themselves four extra characters to play with (Randall, 1997).

Programmers felt that this idea of saving space made sense at the time, and the idea of using six instead of ten characters in the date field was passed along. Furthermore, programmers figured that no one would be using the same programs in their computers today as they did back then (Nixon, et al., 1998). Many programmers truly, yet incorrectly, believed that the software they were writing would long be retired before the new millennium. In the 1980’s, the average life span of a computer was only ten years (Simpson, 1998). To make matters worse, programmers who came after the original ones wrote more computer codes on top of the original codes (http://www.cbn.org/y2k/basics.asp). In addition to coding mistakes, software developers such as IBM did their part to add to the Y2K problem. They made their new software programs to be compliant with their older ones—the ones with the Y2K problem (Nixon, et al., 1998). Today, the western world runs on computers. Consequently, older machines still running on
versions of COBOL may well crash when we hit the moment of truth on January 1, 2000 (DeJagger, 1999).

**COBOL Language: Past, Present, and Future**

The Y2K bug exists because of a computer programming language known as COBOL. COBOL stands for *Common Business-Oriented Language*. It was invented in the late 1950's when computers where the size of office cubicles and the most advanced data-storage system came on strips of punched cardboard (http://www.software.IBM.com/ad/cobol/wpappal.htm) called punch cards (“The Backyard Project”, 1999). A committee was formed called the “Conference on Data Systems and Languages” that was led by Grace Murray Hopper, a naval officer, to create a portable language for business. Finding no suitable one available, Grace Murray Hopper and Robert Bemer created one. They invented the first computer compiler as well as helping to write COBOL language.

The problem that emerged was space considerations. To solve the problem quickly, COBOL programmers used just a six-digit code for the date (i.e., mm/yy/dd) instead of an eight-digit code (i.e., mm/dd/yyyy) (http://www.software.IBM.com/ad/cobol/wpappal.htm). The situation this created however would come back to haunt its inventors 40 years later. How big is the problem? No one really knows. COBOL, due to its ease of use, has had the most success of any language except maybe the computer language known as C. COBOL has had two revisions—one in 1974 and the other in 1993. Eighty percent of computer code is written in COBOL (http://www.jdsa.com/~cobolgo/solution.htm).
The COBOL programming language was designed specifically as a development tool for business-related application software (http://www.mydreamjob.com/tex/tools/cobol.htm). It has been the backbone of the business data processing industry. Moreover, it became the language of choice of computer professionals in automating accounting, financial, and other corporate functions (http://www.mydreamjob.com/tex/tools/cobol.htm). Thus, the Y2K bug has spurred an enormous demand for COBOL programmers.

**Breakdown of COBOL program components.** The language itself is a general purpose programming language that provides opportunities for all levels of programmers from entry-level to advanced (http://www.mydreamjob.com/tex/tools/cobol.htm). It has four main components called divisions. In a COBOL program, divisions appear in the following order:

- **IDENTIFICATION DIVISION**
- **ENVIRONMENT DIVISION**
- **DATA DIVISION**
- **PROCEDURE DIVISION**


The IDENTIFICATION DIVISION, as the name suggests, documents the program's name, author, installation, date-written, date-compiled, and security factors. The ENVIRONMENT DIVISION defines the physical structure of the program files. The DATA DIVISION contains the statements that describe the data used by the program. Finally, the PROCEDURE DIVISION contains the
COBOL statements that run the program itself
(http://www.mydreamjob.com/text/tools/cobol.htm). Each division can be divided
into smaller units called sections. COBOL sections are further broken down into
paragraphs and sentences.

Versions of COBOL. There have been five versions of COBOL
(http://www.mydreamjob.com/text/tools/cobol.htm). The first version was
published on May 5, 1961, under the name COBOL-61. Programmers began
using this version in early 1962 when the first compiler for COBOL became
available. A compiler, for those who aren’t familiar with the word, interprets a
program that is written in a manner that we can understand and translates it into a
language that the computer can understand. The second version, COBOL-68 t,
came out in August 1968, and is known as the first official standard version. The
third revision came in 1974, and the fourth in 1985. The last revision, known as
COBOL 9x, can be implemented both on mainframe computers as well as PC’s.
Unfortunately, our group has been unable to locate a release date for that version.

COBOL’s present state. COBOL has been slow to adapt to the Y2K
problem, but is beginning to move forward. Many tools and utilities are now
available or in development to produce more modern applications (McFarland,
1998). McFarland (1998) notes seven facts that summarize the present state and
future implementations of COBOL:

1. It is estimated that up to 200 billion lines of COBOL code is in
existence; that is 30 million man-years.
2. There are over 3 million COBOL programmers in the world; 2 million of which are in the US alone this is still not enough to completely solve the Y2K problem by the year 2000.

3. Over 50% of all new mission critical computer applications are still being done in COBOL.

4. COBOL 85 language level compilers contain interactive debuggers.

5. Most compilers allow for bridging COBOL’s data by the powerful tools available today.

6. Most all interface (can talk) with database systems that use direct access database data.

7. Most COBOL implementations today support the client server model upon which ERP systems are dependent.

The future of COBOL. So what is being done to combat the dreaded Millennium bug? IBM has created a software application called the millennium date compression tool, which designates dates up to 2059 as “A0”. Further research on this particular application has not unearthed any better explanation although it is a topic that will be looked at for clarification (IBM, 1999). This application allows a user to identify where within an application a Y2K problem exists. It also allows for identification of code issues and overflows that are not Y2K related, and helps to verify that an application is Y2K compliant after changes are completed (IBM, 1999). Another application that IBM has created is called VisualAge COBOL Enterprise. VisualAge gives users the capability to
analyze their applications, convert their code to the latest COBOL standard version, and to identify and fix potential date-related fields (IBM, 3/30/99). A review of "The Great COBOL Debate" (December 6, 1995), that discusses customer testimonials about VisualAge and other object oriented technology is available on-line at http://www.software.ibm.com/ad.htm.

COBOL has been and continues to be widely utilized in the business industry due to its ease of use and portability. Although it was the "perfect solution" at the time, the invention of the language in the mid-1950's has repercussions 40 years later as we head into a new century. The necessity to conserve space caused a scramble to fix the two-digit date dilemma. This scramble, in turn, creates a huge demand for COBOL programmers.

Unfortunately, there is not enough time to fill that demand. Even though great strides have been made in the way COBOL programs can be altered to accommodate "00" dates, much still needs to be done. Client-server environments are new to COBOL. Hopefully, the ERP systems of today are adaptable enough to assist in the transition of legacy programs to wholly interactive programs.

COBOL was once the most widely used language in computer programming. As companies downsized, many of the original programmers who wrote the programs in COBOL found themselves without jobs. Now that the Y2K issue has become a hot topic, many companies have to scramble to find people who know COBOL to help them through their own company's Y2K issues (Simpson, 1998). These programmers are now few and far between which has
created a challenge for information system managers to lure, and keep, COBOL talent (Callaway, 1997).

Embedded Systems: Another Y2K Issue

It is estimated that only about 10% of the 4 billion microchips manufactured in the year 1996, actually went into a device that we call a computer. This fact leaves us with the question, “Where did the rest of them go?” (Kappelman, 1998). Beyond the threat of the date being miscalculated in the Year 2000, there is the more serious concern: “embedded chips” or embedded systems.

Embedded chips are individual microchips with computer codes “burned” right into them. Embedded chips are devices that are used to control the operation of equipment, machinery, or even manufacturing plants (Vector International, 1998). Embedded systems are built-in, pre-programmed chips that govern everything from microwave ovens to heart-lung machines to nuclear reactors (Kirsner, 1998). About 2% of the 25 million embedded chips in use worldwide may malfunction after the century date change. This means that we can expect Y2K problems in approximately 500 million systems, devices, or components (Kappelman, 1998).

Embedded systems have many attributes different from what we normally refer to as computers. Embedded systems:

• are packaged in an enclosure that does not look like a computer.
• usually do not have a keyboard or monitor display. Special equipment must be used to test these systems.

• usually involve "real-time" operations. Real-time is defined as a program that runs in actual time, for instance something that takes 2 minutes to do will actually take 2 minutes on the computer. This places constraints on the operating systems used in addition to application development, programmer, and other required skills.

• often use obscure operating systems, for which compilers may not be readily available.

The questions that should be asked are, "Which embedded systems will fail?" and "Where are these systems in use?" (Kappelman, 1998). If answers to these questions are found, the next task is to contact the vendor of the product (if it is still in business) and determine if the device is Y2K compliant. It is generally a matter of retiring or replacing the device altogether (Gibbons-Paul, 1998) because the source codes cannot be fixed (http://www.cbn.org/y2k/basics.asp). Patches or "work arounds" are not effective (Kirsner, 1998).

There are four categories of embedded systems that have a large possibility of being affected by the Y2K problem: (1) individual microprocessors; (2) small assemblies of microprocessors with no timing function subassemblies
with a timing function; and (3) computer systems used in manufacturing or process control (Vector International, 1998).

Finally, embedded systems are problematic because they often have programs that are loaded into “Read Only” memory and “PROMs.” Here, after the software is remediated, the repair process will then require a repair action to replace that/those-integrated circuit(s). (“The difficult process of remediating embedded systems”, 1998).

With a greater awareness of the Y2K problem and its historical emergence, the role of ERP systems is important to understand. The following section discusses ERP.

What is ERP?

Definition

ERP is short for enterprise resource planning, a business management system that integrates all facets of the business, including planning, manufacturing, sales, and marketing. As the ERP methodology has become more popular, software applications have emerged to help business managers implement ERP (http://webopedia.internet.com, 1998).

Evolution of ERP Systems

The focus of manufacturing systems in the 1960’s was on inventory control. Most of the software packages then, which were usually customized, were designed to handle traditional inventory concepts. In the 1970’s the focus shifted to Material Requirement Planning (MRP) systems (Shankamarayanan, 1998). The MRP systems instructed manufacturers when to order materials and
how much to order, whether from their own plant or from outside vendors. The requirements of an MRP system included a master production schedule, bill of material, inventory availability, listing of purchase orders, and production orders outstanding, in addition to lead times for each component or product (http://cbpa.louisville.edu, 1998). This MRP system translated the master production schedule. This master production schedule displayed the items already produced and forecasted items to be produced (http://cbpa.louisville.edu, 1998).

During the 1980's the concept of Manufacturing Resource Planning-II (MRP-II) evolved as an extension of MRP to shop floor and distribution management activities. In the early 1990's, MRP-II was further extended to cover areas like engineering, finance, human resources, and project management. In short, it covered all activities within any business enterprise. Hence, the term ERP (Enterprise Resource Planning) was coined (Shankarnarayanan, 1998).

In addition to system requirements, ERP addresses technology aspects like client/server distributed architecture and object oriented programming. A seamless integration is essential to provide visibility and consistency across the enterprise (Shankarnarayanan, 1998). A chart on the evolution of ERP systems follows in Appendix B.

Why is it important?

Manufacturing organizations are complex systems where the interactions between the various functional areas have to be managed in order to promote a common purpose of delivering quality at optimal prices. In reality, each function/department works towards their own goals and objectives, rather than an
overall organizational goal. This type of problem is what enterprise resource planning software is designed to solve. The solution is to provide a common, consistent system to capture data organization-wide without redundancies. In addition to integrating the information across functions, ERP also provides a set of tools for planning and monitoring the organizational functions and ensuring progress towards a common organizational goal (Sudhakar, 1998). This is the only way for companies to remain competitive through such an effective and efficient integration of operational functions. Because of Y2K, companies are scrambling to get their computer systems Y2K compliant. Here, the problem is that there are many different types of software currently running on their computer systems. A way needs to be found to enable the different software to communicate with one another.

Implementation of an ERP System

The implementation of an ERP system creates new methods of conducting business and operating day to day events of an organization. It requires the acceptance, understanding, and active participation of every member of the organization. All employees must be trained on the system so that when the company's old system is shut down there is a successful conversion to the new ERP system. The most effective combination of delivery media, methods, and timeliness must be utilized and accounted for when converting a company to an ERP system. According to a leading ERP system developer, there has been a shift from formal classroom setting to hands on. As much as "75% of ERP systems training took place in the class . . . but now only about 25% is instructor-led. The
other 75% of the time, users at classroom workstations practice on the R/3 system” (King, 1996, p 12). There has been a shift from formal classroom setting to hands on. In order to successfully train employees, customized multimedia courses in training and implementation can be obtained from developers such as SAP, Andersen Consulting, IBM, and Hewlett Packard.

Overview of the Installation Process

After the software is bought, a core project team is assembled. The core project team members are then sent to training, usually at the ERP vendor’s training center. When they return they start configuring the software and reshaping the business processes so that everything fits together; including examining current work processes and then planning how they will operate once [the ERP] is implemented (Filipcza, 1997). As the team works, it learns more about [the ERP], and changes both the software and planned business processes accordingly. As it does this, the software programmers on the team also figure out how to transfer all the data from the legacy systems to the new ERP databases. The next step is to set an implementation date or the ‘go live’ date as it is referred to in the industry.

How To- and Not To-Implement ERP

The above is, of course, a simplified overview of how an ERP is implemented. A lot of planning goes into an implementation. Before installing business planning systems such as SAP’s R/3 system, companies typically hire outside help to diagnose what is wrong with operations and create a treatment plan (Radosevich, 1998). They then revamp their processes and install the ERP
system (Radosevich, 1998). Cindy Bernstein, vice president of sales and marketing at IMG in Switzerland points out, "Most customers are spending millions and millions of dollars implementing R/3," (Radosevich, 1998). IMG and the University of St. Gallen, conducted a study examining process benchmarking at five-multinationals which found that companies were not systematically implementing process management after implementing the R/3 system (Radosevich, 1998). The study also found, as with similar studies, that companies could significantly improve their processes by regular benchmarking of processes. One example given was a chemical company whose benchmarking had caused a change in its order entry such that they saw a 7% improvement in data quality performance.

Another aspect of ERP implementation that is often overlooked is how much of the budget should be allocated to end-user training. SAP AG says about 12% of any SAP implementation budget should be allocated to end-user training (Torode, 1998). This training should not only be on how to use the system itself, but also on how the end-user's job will change once the system is implemented. "The success [of this implementation] hinges on understanding the processes," said Bill Magruder, director of change management at Clarke American Inc., located at Alamo, Texas (Torode, 1998). Due to the importance of understanding the processes, Clarke American has arranged with Contra Software, Inc. and DA Consulting to create an online help system that documents all of Clarke's processes (Torode, 1998).
So when will, and will not, implementations succeed? According to Robert S. Putrus, manager of Business Systems Advisory Service at Coopers & Lybrand located in Detroit; although firms need to take a careful, methodological approach to implementation, “Rapid implementation is key,” to go slow is to court failure (Vasilash, 1997). Putrus points out that since ERP crosses functions within a company, the program needs a champion with the authority to bring functional managers together (Vasilash, 1997). If these people do not see it in their best interest to devote time and resources to the project, they will undoubtedly find something other than working on ERP to do (Vasilash, 1997).

Putrus therefore recommends a four-step approach to implementation:

1. Define the requirements. This is project planning. Develop a list of the key features, and functions. Develop business scenarios, from order to shipment and collection.

2. Perform a system evaluation and selection. Do reference checks. Select a vendor. (Putrus suggests that consultants are extremely helpful at this stage because they have had experience working with various ERP suppliers.)

3. Procure and pilot the system, and

4. Go live/cross-over to the ERP system (Vasilash, 1997).

How an ERP implementation affects the business

An enterprise system, by its very nature, imposes its own logic on a company’s strategy, organization, and culture. It pushes a company toward full integration even when a certain degree of business unit segregation may be in its
best interests. It also pushes a company toward generic processes even when customized processes may be a source of competitive advantage (Davenport, 1998).

In order to understand the attraction of enterprise systems as well as their potential dangers, you first need to understand the problem they're designed to solve: the fragmentation of information in large business organizations. Every big company collects, generates, and stores vast quantities of data. In most companies, though, the data is spread across dozens or even hundreds of separate computer systems each housed in an individual function, business unit, region, factory, or office. Each of these so-called legacy systems may provide invaluable support for a particular business activity (Davenport, 1998).

Enter the enterprise system (ES). The database collects data from and feeds data into modular applications supporting virtually all of a company's business activities across functions, across business units, and across the world. When new information is entered in one place, all related information is automatically updated (Davenport, 1998). An ES streamlines a company's data flow and provides management with direct access to a wealth of real-time operating information. For many companies, these benefits have translated into dramatic gains in productivity and speed, but the very quality of the systems that makes those benefits possible, their almost universal applicability, also presents a danger (Davenport, 1998). An enterprise system is, after all, a generic solution. However, some degree of ES customization is possible. Because the systems are modular, for instance, companies can install only those modules that are most
appropriate to their business. However the system’s complexity makes major modifications impracticable. As a result, most companies installing enterprise systems will need to adapt or even completely rework their processes to fit the requirements of the system. An executive of one company that has adopted SAP’s system sums it up by saying, “SAP isn’t a software package; it’s a way of doing business” (Davenport, 1998). It will often be in a company’s interest to go ahead and rework its processes to fit the system requirements. The alternative, customizing the system to fit the processes or writing proprietary application modules, will simply be too expensive to justify (Davenport, 1998).

Tracking implementation practices and training methods is at the core of this pilot study. An examination of the development of ERP software provides insight for this quest. Here, the level of technology training required for successful ERP system implementation can be understood more clearly.

ERP Software Developers: The Market Share

Competition in enterprise software is fierce and complex. Over 500 software makers worldwide are competing for sales. Ten main companies dominate the core of this business. These companies include SAP, Oracle-Applications, J. D. Edwards, PeopleSoft, Baan, SSA, JBA, Marcam, Intentia, and QAD. These main suppliers of ERP software make up 73% of the total market share (See Appendix A). The remaining 27% is represented by other up-and-coming companies including Siebel Systems, Vantive, Clarify, and Remedy (Kirkpatrick, 1998). J.D Edwards, Baan, Oracle, PeopleSoft and SAP primarily sell products to automate finance, manufacturing, and human resources. SAP and
the remaining companies, especially Oracle and PeopleSoft, assert they can provide software for virtually every corporate function (Kirkpatrick, 1998). The following discussion briefly summarizes background information about the top ten ERP system developers.

**SAP**

SAP stands for *System Applications and Products* (Hernandez, 1997, p 2). In 1992, four former IBM employees founded the company that developed SAP and SAP AG. Company headquarters are based in Waldorf, a German town close to Heidelberg, where the university is a continuous source of employees at SAP (Hernandez, 1997, p 1-2). Since its foundation, SAP has made significant development and marketing efforts on standard applications software, being a global market player with its R/2 system for mainframe applications and its R/3 system for open client/server technologies (Hernandez, 1997, p 2). R/2 and R/3 are also known as Enterprise Resource Planning (ERP) Systems software.

As an ERP system, "SAP is a completely integrated, enterprise wide information system that replaces legacy systems with a series of software modules that communicate with each other seamlessly, replacing current business processes with best practices" (Filipczak, 1997, p 41). In other words, it relates all company information from manufacturing to accounting into one total information system using relational databases. Data that is input at one point in the system can be accessed in real-time and utilized at any other point in the system. This enables the distribution of pertinent data to decision-makers at all levels at a company or enterprise-wide scope.
SAP AG

Since the introduction of R/3 in 1992, SAP AG has become the world's leading vendor of standard applications software. One of the reasons for SAP's success is that since it is a standard package, it can be configured in multiple areas and adapted to the specific needs of a company (Hernandez, 1997, p 2). SAP has greatly based the functionality of its R/2 and R/3 software systems in the business process concept. SAP defines a business process as the complete functional chain involved in business practices, whatever software module has to deal with it (Hernandez, 1997, p 4).

Another important part of the strategy to meet today's complex business needs is international applicability. For SAP, this means not only having the software available in different languages but also covering the differentiating aspects of each country, currency, taxes, legal practices concerning human resources, along with import/export regulations. For example, users from a multinational company in different countries can work simultaneously in the same system using their own language, currency, and taxes (Hernandez, 1997, p 5). The company's main focus is to provide comprehensive solutions for businesses of all sizes and industries using this integrative computer software.

SAP is the world's largest enterprise software company with 29% of the ERP market share (See Appendix A). They provide companies of all sizes with business solutions that deliver a better return on information. SAP products and services integrate an organization from financials and human resources to manufacturing, sales and distribution (http://www.sap.com, 1999).
Oracle

Oracle holds the second largest ERP market share at 10% (See Appendix A). Headquartered in Redwood Shores, California, Oracle is the first software company to implement the Internet computing model for developing and deploying enterprise software across its entire product line: databases and relational servers, application development and decision support tools. Consulting, education, and support services are also offered (http://www.oracle.com/corporate/pressroom/html, 1999).

J. D. Edwards

J. D. Edwards has 7% of the ERP market share (See Appendix A). They develop enterprise/supply chain computing solutions that enable companies to translate ideas into realities quickly and efficiently. Their comprehensive, integrated solutions for manufacturing, distribution, finance, and human resources combine with a flexible technology enable companies to put their ideas into action with unprecedented control. J. D. Edwards positions organizations for the year 2000 with technology that accommodates forward century dating. Their current releases of One World and World Software incorporate the Julian dating format, which is the key to correctly interpreting dates in the year 2000 and beyond (http://www.jdedwards.com, 1999).

PeopleSoft

PeopleSoft holds 6% of the ERP Market Share (See Appendix A). Since 1987 they have built global enterprise application solutions that meet the changing business demands and organizations worldwide. PeopleSoft applications are built...
according to specific Year 2000 standards. There is consistent storage of all date fields in the YYYY-MM-DD format, which enables their application to distinguish between centuries and eliminates the confusion and complexities that arise from inconsistent date storage and processing. There are centralized routines for the consistent storage, input, retrieval, and manipulation of the dates including date math, leap year calculations, and other date-related variables (http://www.peoplesoft.com, 1999).

Baan

Baan holds 5% of the ERP Market Share (See Appendix A). The Baan Company is one of the world’s leading providers of scaleable business solutions, with more than 3,000 customer systems implemented across 5,000 cities worldwide. Part of the Baan Company, Baan Corporate Office Solutions was formed following Baan’s acquisition of The Coda Group in May 1998 (http://www.baan.com, 1999).

SSA

SSA holds 5% of the ERP Market Share. (See Appendix A). SSA’s product line, BPCS Client/Server V6 is currently live or being implemented in more than 1,000 major industrial sector firms in over 4,000 sites worldwide. The BPCS Client/Server solution delivers unparalleled agility and re-configurability to meet changing market demands through a leap forward in ERP technologies that delivers significant business benefits, including century dating (http://ssax.com/about [3/12/99]. The BPCS Client/Server product suite integrates all mission critical
operations, from global financials to multi-mode manufacturing to management of the total supply chain (http://www.ssax.com, 1999).

**JBA**

JBA holds 4% of the ERP Market Share (See Appendix A). JBA is the leading worldwide supplier of enterprise management software to the highly competitive mid-market sector. They are a global supplier of integrated business systems for companies involved in the manufacture, supply, and service of industrial and domestic goods (http://www.jbana.com, 1999).

**Marcam**

Marcam holds 3% of the ERP Market Share (See Appendix A). Marcam Solutions, Inc. specializes in providing manufacturing software for process industries and for broader asset management markets. Having pioneered the creation of process-focused Manufacturing Resource Planning (ERP) software, Marcam continues to distinguish itself with process industry expertise, proven mission-critical solutions operations and key components that easily integrate with core applications from the leading ERP providers, SAP and PeopleSoft (http://www.marcam.com, 1999).

**Intentia**

Intentia holds 2% of the ERP market share (See Appendix A). Intentia is the third-largest supplier of enterprise management systems in Europe and one of the top ten in the world (www.intentia.com, 1999).
QAD

QAD holds 2% of the ERP market share (See Appendix A). QAD manufactures and distributes ERP and Extended Supply Chain software. This software enables companies throughout the world to acquire raw materials, transform them into top-quality products, and deliver them to customers within the shortest possible time. QAD specializes in MFG/PRO software that includes an extensive set of solution components for manufacturing, distribution, financial, supply chain, and service/support management. Configurable and interoperable, it is open to many components, uses either Oracle or Progress databases, and runs in UNIX, Windows, and Windows NT environments (http://www.qad.com, 1999).

With over 500 software makers worldwide competing for sales finding the right one to customize a company’s enterprise software can be a difficult task. Not only will the business be affected, but also the employees.

ERP Influence on Job Redesign and Employee Training

The very nature of this integrative software results in a more comprehensive, rather than functional, view of the company by employees. Terri Younger Miller, CEO of DDS, a Minneapolis-based SAP training and consulting firm says, “The integrated nature of SAP forces employees to understand how other functions in the company operate” (Filipczak, 1997, p 41). After an ERP system is implemented, no longer will employees be able to do their work or make decisions in functional silos. Every piece of information that is input into the system will be immediately available and will be used by other functional areas to make decisions. Therefore, it is essential that the information be accurate
and timely. In the area of job redesign, ERP implementations require people to form working relationships with other areas of the company that were not necessary previously. Also, it requires functional areas to share information and make decisions that, in the past, have been closely guarded (Appleton, 1997). It is these aspects of ERP systems that affect employee training. Before training employees on the technical skills they will need to do their jobs, it is necessary to address change management and employee education of the new integrated environment in which they will be working (Filipczak, 1997).

Year 2000 Training Issues

Training issues dealing with the Y2K problem will eventually affect all employees of a firm whom have any contact with a computer in their day to day duties. Before these employees are affected, another class of employees needs to be dealt with. These employees include computer programmers, and the rest of the Information Technology (IT) staff who will be responsible to implement Y2K systems. While it is true that almost every employee that uses a computer will have to be re-trained in the new Y2K compliant computer systems, the computer programmers and the rest of the IS staff that is responsible for implementing Y2K systems will have to be trained first. Much of the training that needs to be done has overwhelmed many IS departments, and many companies are finding the need to outsource their training. Luckily, there are a growing number of companies that are finding their business very lucrative with the onset of the Y2K issue by producing such training.
Before the employees of a firm can be trained in their new systems, a Y2K project needs to be initiated. This means that a project manager will either have to be hired for the company, or someone from inside will need to take the duties over. Anyone who holds experience in project management training is hard to find today because of the onset of Y2K. According to Stamps (1998), a recent survey reports that the average salary for Y2K project directors is $121,000.

Many companies are also realizing that they can not put Y2K training programs into the hands of IS departments and hope that everything will eventually fall into place. Because of these issues, many training vendors are cashing in on different company’s Y2K projects. ComputerPREP Inc. sells training courses on technology has been selling about 1,000 copies per month of a course called Year 2000 Executive Awareness Program (Stamps, 1998).

According to Stamps (1998) IT training will be one of the first things to get sacrificed due to the huge cost of Y2K. Because some IT projects had to be put on the back burner because of Y2K, IT training has been reduced on average by 40%. Training has had to be postponed in many instances. The Y2K deadline is one that companies cannot afford to miss.

Shortage of Trained IT Professionals

Because of the highly specialized IT skills that the Y2K issues are bringing to the table, many people who know some of the programming languages that were used before (COBOL, Oracle, etc.) are finding themselves in high demand, and even higher compensated. This has brought about a shortage of
knowledgeable IT staff. Thus, so many training programs have popped up to handle this shortage.

The University of Maryland and Caliber Learning Network Inc. have jointly developed a training program to teach employees COBOL skills that can be applied to Y2K projects. Employers of the students have paid the $4,000 cost to train their employees. This program is aimed at producing programmers that are capable of handling basic Y2K fixes and testing in COBOL (Caldwell, 1998). According to King (1997), there is a company offering a three-week, study at home crash course to learn COBOL. The course was designed to help ease the Y2K skills crunch that many programmers are facing. There has been a great deal of debate surrounding this course. Many experts are skeptical as to whether or not the student will learn enough COBOL to fix real-life Y2K programming problems.

Other companies are finding it worthwhile to train people who are new to the IT field by also offering training to professionals from other fields in mainframe programming languages. Once these new recruits have been trained, they can be hired. Complete Business Solutions Inc. has hired over 50 of them for their firm. The rest of the students who passed through their training classes were hired by other companies for a placement fee averaging $1500 (Torode, 1998).

Technical Training

Information technology training can be defined as training for both end-users and IT professionals on computer applications and systems (Anonymous, 1998). In a study that examines the technical training practices of the United
States Post Office, Selden (1996) reports that in order for employees to perform most types of technical work, they need to gain a theoretical understanding of their job. Selden (1996) also states that employees need to develop the physical and mental skills to accomplish the tasks associated with their jobs as well. Katz (1955) states that technical training is geared to helping employees increase an understanding of, and proficiency in, a specific kind of activity, particularly one involving methods, processes, procedures, or technique.

The United States corporate training market was positioned at more than $60 billion in 1997, and $20 billion of this figure was spent on technical training (Bernstein, 1998). It is important that companies understand the processes of technical training for their employees and customers. Training must be conducted as an integral part of the ERP project, and the earlier the knowledge transfer occurs in the implementation process, the better the chances that the company implementing the system will have a successful adoption of ERP. The United States' ERP education and training services market was $770 million in 1997 and is estimated to exceed 30% growth level over the next 5 years (IDC PR Newswire, 1998).

High-level technology training can be delivered via many different mediums. In 1996, respondents to a survey sponsored by Training magazine estimated that only 17% of all computer-skills-training in their organizations was delivered via technological means. In 1997, that figure jumped to 24%. It was also found that workers in some industries, such as business services, manufacturing, transportation, communications, and utilities were more likely to
get their computer skills training via technological means rather than being instructor led (See Table 5) (Anonymous, 1998).

Bernstein (1998) states that although many companies are beginning to realize just how important technical training of their employees is to their company’s survival, there remains a dilemma that cannot be overlooked. How can these companies effectively train their employees on technology without spending millions of dollars and time sending people around the world for crash courses? It is even more difficult to keep employees up with the rapid pace of information technology changes year-round with traditional instructor-led training. McCartney (1996) found that many information service executives and professionals believe that development cycles of computer software and hardware are too rapid and they can not afford or absorb all the new tools coming their way.

According to Koonce (1998), employee trainers should be able to anticipate the need for their companies to implement change, redesign processes, accelerate employee learning, and introduce new technologies. They can do this by becoming anticipatory learners themselves, especially in training that deals with technology. It is imperative that they improve their understanding of distance learning, how to conduct Internet and Intranet based learning, performance measurement, and how to coach. A critique of the types of high-level technology training delivery systems provides focus for discussion. A discussion of each kind provides more detail as well as the pros and cons of each kind.
Types of Technical Training

Many companies are at a loss for which type of high-level technical training they should be implementing for their new Y2K systems. Ginsburg (1997) suggests that deciding on the correct method of training depends on a number of factors including the expense, the number of people that are to be trained, and their skill level. Employees who are already computer-savvy might only require self-paced training like computer-based-training. It might be better to start employees who are at an intermediate level with instructor-led course and allow them to move gradually into a computer-based training system as they feel more comfortable with the new technology.

Technical training can be delivered by many methods. Some of these include computer-based training (CBT), Internet based training, (IBT), asynchronous computer conferencing (ACC), Electronic Performance Support Systems (EPSS), Instructor-Led Training (ILT), & Coaching.

CBT. Computer Based Training (CBT) is an interactive training experience between a trainee and a computer, in which the computer provides most of the stimulus. The delivery of the training consists of information, quizzes, and tests. The program ultimately acknowledges if the student has learned the material (Munger, 1996). Horowitz (1997) states that CBT shares some attributes with IBT. Both involve self-directed, self-paced instruction and non-linear learning. Students have the freedom to review the material presented to them in any order they want. According to Horowitz (1997), when
McDonald's Corporation switched from classroom training to CBT, they cut their training costs by 18%.

CBT tutorials have been found to offer companies many advantages such as availability, cost, convenience, being self-paced, ease of distribution, ability to change, on-line help, reference guides, job aids, electronic brochures, and work simulation (Ganger, 1994). A pitfall of CBT is that many students may not be as motivated to learn without an instructor standing over them at all times, and it is hard to justify the costs of using CBT for small groups (Horowitz, 1997). Comaford (1997) also suggests that students might find CBT boring to use.

Oakes (1997) states that few people have tracked the number of hours of development per one hour of a CBT course. Most CBT developers can tell you how many months it took to develop the course, but not the number of "people hours". In order to track the development time of CBT accurately, the following factors must be considered. These are the complexity of the instruction, the degree of interactivity, the amount of media used in the course (i.e. video, audio, three-dimensional animation), and the amount of tracking the course will feature.

IBT. As mentioned above, IBT is similar in many ways to CBT. Horowitz (1997) states that many IS managers are reluctant to employ IBT because it is new and unproven, but a growing number plan to adopt the technology soon. Above and beyond CBT, IBT allows students to ask questions via electronic mail, get immediate feedback, and interact via online chat sessions. IBT offers more advantages such as the opportunity for a large number of people to participate and collaborate. There is also an opportunity for continuous
learning and sharing among peers (Koonce, 1998). Some companies offer a variety of courses to their employees worldwide via the Internet. Employees can access most programs at any time and are helped by an online automated tutor. They can even join in real-time discussions. It is also possible for trainees with digital cameras attached to their computers to see each other during these discussions (Munger, 1997).

A glitch that might be found in IBT is that some Web sites might change addresses or disappear because of the unique structure of the Internet. Most companies will also need to put firewalls on their networks when implementing IBT so that they can keep outsiders and viruses from getting in. Finally, the skills and tools needed for developing IBT are still new and in development (Munger 1997).

Munger (1997) states that more sophisticated IBT and web pages require more time than a simple home page which can be developed more quickly, sometimes in a matter of hours. A “reasonable” ratio for the development time of an IBT course is six hours for every instructional hour.

ACC. ACC uses computers as the medium for delivery as well. It provides trainer-to-trainee and vice-versa interaction in an electronic setting. It is possible for trainees to work together on the same topic and receive messages from the trainer simultaneously. The instructor will usually provide information and illustrations online while the text and supplementary materials are sent to the trainee separately. On their own time, the trainees will participate in electronic-forum discussions, presentations, and examinations (Munger, 1997).
Much like CBT and IBT, students can learn when it is convenient for them. High level video and graphics are often used in ACC. It is an excellent means of training for members of a team that work at different locations. On the other hand, instructors of ACC will also need training to use this approach to training effectively. Students that enjoy more personable training (such as instructor-led or one-on-one) may not be as motivated to learn via this medium (Munger, 1997).

Due to the many similarities between ACC and CBT, the time and expense to develop ACC is around the same as that of CBT (Munger, 1997).

**EPSS.** Electronic Performance Support Systems (EPSS) are computer based training systems that are able to give the user on-demand access to coaching, task specific training, task-specific information access, and expert advice needed to solve job-performance problems (Benson, 1997; Stevens & Stevens, 1996). In addition to providing technical information, EPSS can also serve as both on-line technical manuals and complete training modules.

There is a similarity between EPSS and multi-media training because both are collections of software that operate on a computer system, both have been developed to allow employees to perform better, and both use multi-media such as texts, graphics, and animation. The main difference between the two is that multi-media training is typically done when an employee is away from his or her job and an EPSS is utilized when an employee is actually in the process of performing his or her job (Dahmer, 1994).
While EPSS allows employees to do their jobs better and faster, there is a myth about the cost effectiveness concerning this kind of training tool. Stevens & Stevens (1996) state that the time and costs of developing and implementing EPSS can be recovered from the invaluable information and skills it provides to employees. Employees can look up a question or how to do a particular task on their EPSS before they call the help desk, which can involve extra costs to the company. Benson (1997) found that because the initial cost of purchasing and initiating in-house development of an EPSS can be so high, senior management is more likely to make the final decision on this matter, not the training department. Although training departments are most likely to take responsibility for budgeting for EPSS.

Benson (1997) reports that larger companies of more than 1,000 employee tend to use EPSS more than smaller companies. Benson also reported that it is more likely to find that companies that do utilize EPSS tend to use simpler versions of this system, such as an on-line help system.

Stevens & Stevens (1996) explain that in order for EPSS to be effective in any company, there needs to be a partnership between both the training department and the IS department. Each department can complement the other with their unique sets of skills and understandings about training and software implementation. For example, trainers tend to have knowledge of the training needs that exist within the organization's line workers and the best way to organize and prevent training. The IS staff should be able to bring their
knowledge of computer system design, development technology, and programming know-how.

ILT. Instructor-led training (ILT) involves teaching a group of students in a typical classroom setting. ILT has been the traditional way of training employees for many years. This was before new technology that offered trainers new approaches to training. While it seems that ILT is becoming more and more obsolete with the new computer training tools that are out there, Broadbent (1998) reports that 75% of all training in the United States is instructor-led at a total cost of more than $40 billion. This is probably because many trainers still see ILT as a good way for employees to develop and build a deep knowledge base (Comaford, 1997). Gibbons-Paul (1997) states that ILT can be especially useful if a company is going to be involved in a major technological move. The employees who go through an ILT class can start to get accustomed to the new environment that they will be thrust into.

Coaching. While coaching is typically used for training employees in soft skills, or management practices, it is another alternative available to deliver high-level technological training to employees. When training employees in high-level technology computer skills, someone who is a coach is often seen internally by employees as a mentor. This is much different than an in-house help desk (Ginsburg, 1997). A coach can be useful in helping employees deal with changes in technology, and to get them excited about the new application at hand. Koonce (1998) suggests that when a trainer becomes a better coach, they will enhance
their professional opportunities now, as well as the market demands for their services in the future.

Coaching can also be looked at as an apprentice relationship. For example, if one employee is leaving the company and a new one has come on board to learn the job duties, a manager may want to allow the employee that is on their way out the door to coach the new employee on certain skills as a one-to-one basis. The benefits of this approach include being one of the cheapest ways to deliver training in addition to the minimal time needed to prepare and develop the training. There are also some drawbacks to this kind of apprentice/coaching approach. The employee that is leaving the company may not be very motivated and simply may not care about what kind of information is given to the new employee. The end-result of this type of training is also hard to evaluate. Finally, the information that is delivered to the new employee may not be all-encompassing of duties that may have to be performed at different times throughout the year (Houser, 1992).

Combining Different Types of Training

It is not at all uncommon to find companies combining several different training methods to help their employees become adept at their new technology. Schatz (1996) discusses a training method for computer applications called Show/Do/Cue which facilitates quick understanding and works well when there are different members of a training group who are at different skill levels. First, Show/Do/Cue utilizes a brief lecture (or ILT) which gives an overview of the computer program that is going to be taught. The next step involves hands-on-
learning with teacher coaching (or modified CBT), followed by independent practice with hints or cues guiding the student along the way.

There are many reasons for companies to combine training deliveries. At many companies, cost is a major issue when selecting a training delivery type. Other companies may look at how much transfer, if any, occurred when the employees had to go back to their jobs and work after spending hours in a classroom. It may not make sense to spend thousands of dollars on putting each employee through ILT sessions just to have them forget it once they return to work. At this point, it would be a good idea to incorporate EPSS into training after ILT has been completed.

Timing of the training is another issue that should be taken into account when choosing appropriate training delivery techniques. Gibbons-Paul (1997) states that if employees are trained too early in information technology skills (before they can actually use it on the job), an expensive refresher course will most likely have to be offered down the road once the employees have forgotten everything from the initial training. Training employees too late in information technology skills may send the message that the company expects them to rely on the help desk or each other.

Perhaps the best option for companies is to implement a user-friendly EPSS when changing to a new high technology system. ILT should be offered to all employees to introduce the change that is coming and offer some basics before any actual CBT, IBT, or other form of training takes place. This way, transfer of
training can almost be guaranteed because employees will actually have help with their work every step of the way.

**Costs of Technical Training**

Estimating the range of possible costs for a training program can be difficult. Hassett (1992) states that the cost of training equals the number of developer hours per hour of training, multiplied by the development cost per hour, multiplied by the number of hours training.

According to a survey of Fortune 500 and other private companies in 1995, 29% of these companies spend less than .5% of their payrolls on training, 29% spend .5 to 15%, and 21% spend 1.6% to 3.5%. (White, 1996).

Ginsburg (1997) states that the most expensive training options are ILT which are taught by outside instructors at the company’s facilities. These sessions can range anywhere from $125 to $350 per user per day. ILT taught at a trainer’s facility are a little less expensive, costing anywhere from $175 to $250 per user per session. Ginsburg explains that the prices’ ranges reflect different factors such as the size of the group, or what kind of customization the group needs for training. Houser (1992) suggests that ILT can be cost-effective if there is a significant amount of trainees, the subject matter that is being taught is highly specialized, or the curriculum meets needs that are only unique to your organization.

If ILT does not look like it would be cost effective, CBT could possibly be a cheaper option. The cost per module for self study diskettes is somewhere around $575 to $800 (Ginsburg, 1997). On the other hand, Schatz (1996) states
that CBT is very time consuming to develop and as a result of this, is very expensive to update and keep current with all the product upgrades that are constantly offered.

Why is technical training so expensive? An examination of technical training development time sheds light on this aspect of the problem.

Development Time of Technical Training

How long does it take to develop one hour of training? In search of one single answer to this question of training, experts disagree. Most of the time, the first response you will hear is, “it depends”.

What Does the Ratio Depend On?

A study by the U.S. Office of Personnel Management came up with some rules of thumb for creating one hour of classroom-based, instructor-led training. Again, the amount of hours of development time that they came up with varied a great deal, but it does start a trainer out with six basic guidelines (Zemke, 1997):

1. Five to fifteen hours for technical formal courses
2. 50-100 for self-contained training ready for hand-off to other instructors
3. 20-30 hours of conventional management development
4. 50-100 for self-contained training ready for hand-off to other instructors
5. 20-30 hours of conventional management development
6. One to three hours of technical on-site content
Initial instructor preparation was also examined. For a course that is supposed to last five days or less, a designer should plan on three hours of development time for each hour spent teaching. For courses that are to last between five and ten days, the developer should budget in two and a half hours of preparatory time per hour spent teaching. There should be an allotted two hours per teaching hour ratio for courses that will be lasting for over ten days in length.

Other issues to consider when figuring out how long it takes to design training include varying ratios, the dependence on the instructional design project management model that is used, the amount of non-writing time, the variance of “deliverables”, and the documentation of the history of the development process (Greer, 1996).

**Varying ratios.** First, ratios vary tremendously. Different training developers will give ratios anywhere from as low as 10:1 to as high as 200:1 (Greer, 1996). To narrow down these ratios to more specific times, the training developer should know exactly what they want. They need to be specific and have definitive specs (Monitor, 1998).

**Instructional design project management model dependence.** Second, whatever kind of training is being developed, what matters most when estimating development time is the instructional design project-management model that is used. This means that a developer needs to account for all the activities that they will be performing that are not directly related to writing and revising instructional materials. It can be referred to as “non-writing time” (Greer, 1996). Instructional design programs help trainers to create such specifications as needs,
goals, objectives, content, and test questions. These are some pre-authoring tools that help trainers to complete a thorough needs analysis while simultaneously accelerating the design process (Glade, 1998).

The amount of non-writing time. Third, the non-writing time can often consume up to 80% of a developer's project time. This time would include front-end analysis, brainstorming, preliminary design, review and feedback from sponsors and subject-matter experts (SME), administrating/debriefing student tests, and other “hidden” administrative duties (Greer, 1996).

Deliverable Variances

The way that the instructor delivers the training can vary immensely. Some examples of different delivery techniques include lectures, lectures with video, CBT, and case studies. Once the training developer knows exactly how the material will be delivered, they should be able to make a reasonable estimate of the writing and revising time that will be required, and add it on to their non-writing time (Greer, 1996).

Documentation of the history and development process. If a trainer developer has created their organization’s unique instructional design project-management model, they need to document their history of the time that was spent executing the various steps of the model. They will then be able to make a detailed estimate of their time and cost estimates (Greer, 1996). A training professional’s credibility and financial success greatly depends on how accurate they are when they give a client, or their employer, a measurement of time to design and develop materials for a training program (Broadbent, 1998).
According to White (1996), preparing for an ILT can take 40-100 hours of development time for each one hour of training. One hour of CBT delivery can take anywhere from 100 to 400 hours to develop. The range depends on the designer's skill, experience, and complexity of the material.

Broadbent (1998) offers some factors and variables to examine when trying to estimate the development time while designing high level technological training courses. First, ask, “Who is involved in the design process?” This step helps you examine the designer, the target group, and the client. Secondly, Broadbent (1998) notes two things that should be taken into account:

1. What will be the final elements of the training program?
2. How much work will be involved in designing the content of the course and the packaging of the materials?

Broadbent (1998) also advises on finding the answers to the following three questions:

1. How are the materials going to be designed?
2. How is the designer going to deal with the client?
3. How interactive will the program be?

Implementing high-level technology training takes a great deal of planning before the system can be operational. Dust (1996) offers seven steps for successful software implementation of this magnitude:

1. Determine the business need. Ask these questions:
   - Why is the company installing this new system?
• How much time is there to implement the system?
• How much budget, staff, equipment, and management commitment is there?
• What part does quality play?

2. Select the software and vendor. (Make sure the vendor can provide support in the future.)

3. Establish the environment—make sure all existing operating systems, networks, databases, and communication systems are accounted for. Make sure there is a backup system in place.

4. After the environment has been established, integrate all data. Replacing an older system with a newer system sometimes requires running both of the systems at the same time while fine tuning the newer system.

5. Change processes in the company. A new system will often bring about a need for a new way of doing things.

6. Install product and equipment. (The training department should typically work with the MIS department here.)

7. Training. Train the users intensely, train the management on benefits (get buy-in) and train the support staff not only on their responsibilities, but also on the importance of the system so that they can understand how important their roles are.
Summary

This chapter offered a review of literature as it relates to ERP systems. Topics included Year 2000 and technical training issues in relation to the different types of training delivery methods and development times of those methods. In order for businesses to continue to operate and compete globally their computer systems and applications must be Y2K compliant. As organizations move toward compliance, developing technical training programs becomes a critical component of implementation. This pilot study will promote a greater awareness of common development times and delivery methods may assist organizations plan the implementation process more efficiently. Chapter Three discusses the survey that was conducted regarding how high-level technology training is developed and delivered and how that survey was developed.
Chapter Three

Methodology

This pilot study examines and identifies development times and delivery methods for high-level computer technology training programs with respect to Enterprise Resource Planning Systems. This chapter presents the methodology that was used to conduct this research. It examines the development of the survey, the sample involved in the pilot study, data collection techniques that were employed, and how data were analyzed.

Survey Development

As discussed in Chapter One, the development of this pilot study began when a multinational organization, located in Rochester, New York, contacted the research team requesting that the pilot study be conducted. From this initial contact, the survey was developed. The methods that were used to develop this survey include examining literature to define development time and delivery methods of training and a brainstorming session to create questions to support the literature findings. A panel of instructional designers were contacted and interviewed as well. This draft of the survey was constructed from this information. A panel of instructional designers and trainers who worked specifically with high-level computer technology training programs reviewed this draft. From this review, the panel made suggestions and changes to the survey expanding it and making the questions more specific and relevant.

The original revised survey was developed with a global focus and an invitation to participate was sent to companies in the countries designated by the
company that the group was working with. The companies were decided upon from names that were showing up in the literature that the group reviewed. Response rates, however, were very poor. It was speculated that the reasons for the low rate of response include too scattered a focus, the clarity of the questions, and a lack of willingness to participate. The results that were received were analyzed, presented in a report to the company, and the group was dissolved.

Although the initial results were disappointing, the topic remained a point of discussion among those that had been in the group. A series of “what ifs”, “we should have done...”, “it would be interesting to see what the US is doing”, and “Canada should have been included” were uttered in these discussions. It was then that the realization of how “cutting edge” the topic was sunk in and three of the original five members decided to revamp the survey and utilize it for a master’s project.

In the fall semester the three people decided the focus had been too far reaching and, due to the large interest from Canada, that the focus of the new survey should be on North America. The next decision that was made was to conduct a convenience sample for the survey. A convenience sample is defined by Parker and Rea (1997) as “…one in which interviewees are selected based on their presumed resemblance to the working population and their ready availability” (p 142). The original survey was then redesigned in a virtual brainstorming session. The reason the brainstorming session is being referred to as “virtual” is because one group member typed out what she felt the survey should look like and then emailed it to the remaining group members for
suggestions and additions or deletions. The original invitation letter was then redone to reflect the changes in the group and the focus of the study. A new literature review was then started. In reviewing the related literature, company names and industries were identified that developed and/or utilized training for ERP systems.

Sample

Organizations positioned in North America were chosen partially through this literature review and partially through Internet searches using the keywords “ERP and technical training”. It was in this manner that a convenience sample was created of 188 companies to receive an invitation to participate. Survey participants self-classified their industry category when responding to the survey. Based upon the demographic responses to this question, six categories were arbitrarily created to organize this raw data. The six categories that were created are as follows: (1) Financial Services; (2) Software; (3) Consulting/Training; (4) Government; (5) Healthcare; and (6) Manufacturing.

Data Collection

Data were collected employing survey research (See Appendix C). The following five steps document the data collection process: (1) an invitation was solicited; (2) a survey request was received; (3) the survey was sent; (4) completed surveys were obtained; and (5) follow-up on completed surveys was conducted.

Step 1: Invitation was sent. An invitation letter was sent out to the 188 organizations based on our convenience sample (see Appendix D). This invitation
originated from a centralized e-mail account through http://www.hotmail.com. It was also posted daily to the Training and Development Listserv (TRDEV-L). TRDEV-L is a moderated listserv discussion group that serves a group of professionals and students who work in the training and development field or the Human Resource Development field. TRDEV-L provides a forum for the exchange of information and ideas about improving performance of people in organizations (http://kell167.ed.psu.edu/trdev-l/, 1999). The invitation letter introduced the research team, the research issue, and a brief background of the pilot study. It then solicited participation in the pilot study and provided contact information for the research team including e-mail, fax, telephone, and postal mail address.

Step 2: Survey request received. Requests for the survey were received by e-mail. A log of survey requests was established as an e-mail folder for tracking purposes.

Step 3: Survey sent. The content of the survey included questions regarding methods of training development and allocation of time for the components of analysis, design, and development of the training itself, as well as time allocated to testing the instruction instrument. The content also covered discussion of delivery of instruction, benchmarking for methods, geographic location, industry classification, and measurement of success (see Appendix C). Most survey requests were honored within 24 hours with a reply by e-mail. Others were faxed or mailed. Each survey was sent with a personalized introduction and appreciation greeting.
Step 4: Completed survey received. Completed surveys were received by e-mail, fax, and mail. Responses were recorded in an established data file.

Step 5: Follow-up. Completed surveys were reviewed for missing information and information that may have indicated a misunderstanding or misinterpretation of the question proposed. Respondents were asked to revisit portions of the survey if either condition was present. In addition, follow-up was conducted for requests in which a completed survey was not returned. (See Appendix E).

Data Analysis

Data were analyzed using Microsoft Excel 98. Means, frequencies, and percentages were calculated. In addition, cross-tabulation analysis was employed to examine two research questions that dealt with relationships between geographic location and industry classification with respect to most common training delivery method. These two research questions were derived from the survey questions regarding the geographic location, industry classification, and methods used to deliver technical training programs (see Appendix C). The research questions are as follows: (1) Is there a relationship between most commonly used delivery method and geographic location; and (2) Is there a relationship between most commonly used delivery method and industry classification? A cross-tabulation (Pivot Table) creates an interactive table that quickly summarizes large amounts of data. Finally, content analysis techniques were also employed to analyze how the respondents measured the success of their technical training programs. Content analysis included summarization of
responses to the questions regarding measurement of success and benchmarking that was done as well as arbitrary categorization of geographic location and industry classification. The summary of responses to measurement of success was grouped according to methods identified. The five categories of training measurement were as follows: (1) evaluation; (2) testing; (3) participant practice; (4) monitoring performance; and (5) coaching. Categorization of geographic location resulted in eleven categories (see Appendix F). Content analysis was conducted in this manner because as Parker and Rea (1997) state “...based upon a verbatim listing of all responses to an open-ended question, the researcher...uses his or her judgment to develop categories into which the responses can be placed...” (p 78).

Conclusion

This chapter presented the five-step process followed to conduct the pilot study. Results of the analyses are presented in Chapter Four. At this point, it is important to note the limitations of this research paper. First, this can only be considered a pilot study and that since it is a convenience sample, any generalizations may only pertain to the sample that participated. Second, the survey, in its present form, is flawed and in need of fine-tuning. Third, it is noted that the layout of the survey itself is at some points awkward and that in some cases a range would have been a better measurement than a percentage. However, as a pilot study, the information generated by the research is relevant, timely, worthwhile, and of interest to the Human Resources Development discipline. These issues are further discussed in Chapter Five of this paper.
Chapter Four

Purpose

The purpose of this pilot study is to explore development times and delivery training delivery methods for high-level computer technology training programs with respect to Enterprise Resource Planning (ERP) Systems. The intent of this chapter is to discuss the data analysis results of the survey that was conducted. For this purpose, the chapter opens with a recap of how the data were analyzed. Chapter Four next discusses a description of survey respondents. In addition, the rate of response is detailed and finally the results of the data analysis are presented. Data were analyzed using Microsoft Excel 98. Means, frequencies, and percentages were calculated and cross-tabulation tables were conducted. A cross-tabulation (PivotTable) is an interactive table that quickly summarizes large amounts of data. Content analysis techniques were also employed.

Results

Responses were received from various locations in the United States, several Canadian provinces and five non-North American based areas. For this reason, geographic locations were grouped into ten regions. The United States encompasses seven of these regions: (1) East North Central; (2) Middle Atlantic; (3) Mountain; (4) New England; (5) Pacific; (6) South Atlantic and (7) West North Central. The remaining three regions include Canada, International, and “Other.” Appendix F shows which of the locations fit into each of the regions. The respondents were grouped into six industry classifications: (1) Financial
Services; (2) Software; (3) Consulting/Training; (4) Government; (5) Healthcare; and (6) Manufacturing.

Data Analysis

Rate of Response

There are two response rates to discuss in this chapter: rate of survey invitation response and rate of survey request response.

Rate of survey invitation response. An invitation to participate in the survey was sent to 188 companies via e-mail and was also posted on a discussion group listserv over a period of several weeks. A total of 63 (33.5%) survey requests were generated from these invitations. It is speculated that the reasons for the low rate of survey invitation response are: (1) companies did not have the time to respond; (2) do not participate in student surveys; (3) or that ERP systems training is such a new area that not many companies are developing high-level technology training at this time.

Rate of survey request response. Of the 63 requests received, 47 surveys were completed via email. This puts the survey response rate at 74.6%. This response rate offers findings representative enough to draw worthwhile generalizations from.

Survey Response by Question

The following discussion presents the responses to each question asked in the survey and two research questions that were derived from the survey responses (See Tables 1-5).
Question 1. "Do you develop training programs for high-level computer technology?" For this question a 34 (72%) respondents answered "yes" and the remaining 13 (28%) responded "no".

Question 2. "What training delivery methods do you use?" Respondents were asked to choose from a list of eight methods and give the ratio of development hours per hour of delivered training for each method used. Table 1 summarizes the results.

Table 1 Average Development Ratios by Training Delivery Method

<table>
<thead>
<tr>
<th>Training Delivery Method</th>
<th>Development Ratios (development hrs/hr of delivered training)</th>
<th>% of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean x</td>
<td>Median</td>
</tr>
<tr>
<td>Instructor Led</td>
<td>28.50</td>
<td>22.10</td>
</tr>
<tr>
<td>Web Based</td>
<td>64.90</td>
<td>32.50</td>
</tr>
<tr>
<td>CBT</td>
<td>50.00</td>
<td>25.00</td>
</tr>
<tr>
<td>Printed Media</td>
<td>28.30</td>
<td>16.25</td>
</tr>
<tr>
<td>Coaching</td>
<td>12.65</td>
<td>15.00</td>
</tr>
<tr>
<td>OJT</td>
<td>12.38</td>
<td>9.75</td>
</tr>
<tr>
<td>EPSS</td>
<td>61.67</td>
<td>55.00</td>
</tr>
<tr>
<td>Other</td>
<td>39.20</td>
<td>20.00</td>
</tr>
</tbody>
</table>

1 Total respondents (N) = 33
CBT = computer based training
OJT = on-the-job training
EPSS = electronic performance support systems
S. D. = standard deviation
Freq. = frequency of response or "N"
Table 1 shows that the three most frequent responses to this question were Coaching (freq. =13), OJT (freq. =12), and Instructor-Led (freq. =31). Coaching indicates the least amount of variance ($\bar{x} = 12.65$ and $S.D. = 9.98$), followed by OJT, ($\bar{x} = 12.38$, $S.D. = 10.86$); and Instructor-Led, ($\bar{x} = 28.5$ and $S.D. = 20.60$). These results indicate that there is a high amount of variance among development ratio times even in the most commonly used methods. The fact that most implementations are, as indicated by the majority of respondents, for custom systems explains why there is such a large amount of variance among times. The most variant training delivery method is Web Based (WBT) with a response rate of 30%, mean of 64.90 and median of 32.50. The second most variant training delivery method is Computer Based (CBT) with a response rate of 21%, mean of 50.00 and median of 25.00. Since the mean is so much higher than the median, it signifies little to no standardization in development times for these methods. The possible reasons for this large amount of variance are that WBT and CBT are relatively new training delivery methods, there is a large amount of time required to develop these training delivery methods (Oakes, 1997) and there may be some confusion regarding their differences.

Question 3. “How is development time allocated?” Respondents were asked to provide the percentage of time allocated for the areas of analysis of training needs, design of training method, and actual development of training materials out of total training time (See Table 2).
Table 2 Percentage of Total Development Time by Development Component

<table>
<thead>
<tr>
<th>Development Component</th>
<th>Percentage of Total Development Time</th>
<th>% of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean %</td>
<td>Median %</td>
</tr>
<tr>
<td>Analysis</td>
<td>23.93</td>
<td>20</td>
</tr>
<tr>
<td>Design /Planning</td>
<td>23.28</td>
<td>20</td>
</tr>
<tr>
<td>Develop</td>
<td>49.73</td>
<td>50</td>
</tr>
<tr>
<td>Other</td>
<td>15.60</td>
<td>14</td>
</tr>
</tbody>
</table>

When calculating the distribution of total development time by component, results indicate that the majority of time is spent on actual development of the training materials since the mean was the highest at 49.72. What is interesting is that the time for analysis and design/planning of the training seems to be about evenly allocated. The means for these components are 23.93 and 23.28 respectively.

**Question 4.** “What instructional methods do you use?” The respondents were asked to answer this question by listing the combination of training delivery methods used and designate, if possible, what percent of total training time was spent on each training delivery method (See Table 3).

---

2 Total respondents = 33
S. D. = standard deviation
Freq. = frequency of response or “N”
Develop. = development of materials
Table 3 Percentage of Total Training Time and Responses by Training Delivery Method*

<table>
<thead>
<tr>
<th>Training Delivery Method</th>
<th>Percentage (%) of Total Training Time</th>
<th>% of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean (x)</td>
<td>Median</td>
</tr>
<tr>
<td>Lecture</td>
<td>29.25</td>
<td>30</td>
</tr>
<tr>
<td>Demo</td>
<td>21.29</td>
<td>20</td>
</tr>
<tr>
<td>Hands-on</td>
<td>37.88</td>
<td>40</td>
</tr>
<tr>
<td>Mastery Test</td>
<td>8.25</td>
<td>7.5</td>
</tr>
<tr>
<td>Review Q's</td>
<td>9.41</td>
<td>10</td>
</tr>
<tr>
<td>Other</td>
<td>34.58</td>
<td>22.5</td>
</tr>
</tbody>
</table>

In terms of total time, the data shows that “Hands-on” is the most widely used (x = 37.88 and RR = 57%). Lecture is the second most used (x = 29.25 and RR = 57%).

*Total respondents = 33
S. D. = standard deviation
Freq. = frequency of response or “N”
Demo = demonstration
Review Q’s = review questions
RR = response rate
Question 5. "Did your company benchmark other companies for these training delivery methods and practices?" Out of the 34 respondents who do develop high level technical training, 26 or 81% reported that they do not benchmark. The possible reasons for this include that: it is a relatively new field of training and most training is for custom systems. Of those that do, the majority either benchmark with the company that created its ERP system (e.g. SAP), or are government departments that benchmark with other government departments.

Research Questions

Two research questions were derived from questions six and seven of the original survey (See Appendix C). Cross-tabulation was employed to analyze these data. The purpose for this analysis was to examine if a relationship existed between most common delivery method and the qualifiers of geographic location and industry classification respectively (See Tables 4 and 5).

Research question 1. "Is there a relationship between most commonly used delivery method and geographic location?" shows the most commonly used training delivery method for each geographic location (See Table 4).
Table 4 Most Common Training Delivery Method by Geographic Location

<table>
<thead>
<tr>
<th>Geographic Location</th>
<th>Most Common Training Delivery Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canada</td>
<td>Hands-on</td>
</tr>
<tr>
<td>East North Central</td>
<td>Hands-on</td>
</tr>
<tr>
<td>International</td>
<td>Lecture</td>
</tr>
<tr>
<td>Middle Atlantic</td>
<td>Demo</td>
</tr>
<tr>
<td>Mountain</td>
<td>Hands-on</td>
</tr>
<tr>
<td>New England</td>
<td>Other</td>
</tr>
<tr>
<td>Other</td>
<td>Hands-on</td>
</tr>
<tr>
<td>Pacific</td>
<td>Lecture</td>
</tr>
<tr>
<td>South Atlantic</td>
<td>Lecture</td>
</tr>
<tr>
<td>West North Central</td>
<td>Hands-on</td>
</tr>
</tbody>
</table>

Table 4 demonstrates that “Hands-on” is favored in 50% of the geographic locations. Lecture is the second most common in terms of total training time with a 30% measure by location. No evidence of a relationship between geographic location and most common training delivery method seems to exist.

Research question 2. “Is there a relationship between most commonly used delivery method and industry classification?” presents the most commonly used training delivery method for each industry classification (See Table 5).

---

4Total respondents = 33
Demo = demonstration
Table 5 Most Common Training Delivery Method by Industry Classification

<table>
<thead>
<tr>
<th>Industry Classification</th>
<th>Most Common Training Delivery Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consulting/Training</td>
<td>Hands-on</td>
</tr>
<tr>
<td>Government</td>
<td>Hands-on</td>
</tr>
<tr>
<td>Healthcare</td>
<td>Hands-on</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>Demo</td>
</tr>
<tr>
<td>Pharmaceutical/Chemical</td>
<td>Hands-on</td>
</tr>
<tr>
<td>Software</td>
<td>Lecture</td>
</tr>
<tr>
<td>Financial Services</td>
<td>Lecture</td>
</tr>
</tbody>
</table>

These results support that “Hands-on” is the most popular method with four out of six or 67% response. Once again, however there is no conclusive evidence that there is any relationship.

The remaining survey question: “How do you determine if the participants have learned the subject matter?” was an open-ended question. Here content analysis was applied. The results indicate how organizations measure whether or not the training was successful. A wide variety of responses were received. Given this, the best way to discuss the findings is in summary format.

**Summary of Measurement of Successful Training**

Of the 33 survey respondents that conducted high-level computer technology training, 25 reported that they did determine if their delivery

---

5 Total respondents = 33
Demo = demonstration
method(s) and instructional practice(s) were successful. The results and responses that were gathered as to how this was accomplished varied greatly, ranging from conducting follow-up surveys to having managers measure performance results a few weeks after the training was implemented. It was found that there were five categories that most responses fell into when the respondents reported how they measured the success of their training programs. In other words, did the participants walk away from the training being able to perform what they were supposed to in the end?

The five categories of training measurement were as follows:

(1) evaluation; (2) testing; (3) participant practice; (4) monitoring performance; and (5) coaching. The following sections will discuss each of these categories and report the responses that were relevant to the pilot study at hand.

**Evaluation.** Training evaluation is always a popular way to measure to see if the participants walked away with the skills, knowledge and attitudes that the designer intended for them. Some of the respondents used evaluation techniques at the end of the course in the form of a course evaluation survey. An ERP software vendor responded by explaining at the end of a class, they have class participants fill out a course evaluation survey, in which they can anonymously indicate how well they think they have learned the new concepts and skills. Evaluation surveys were also conducted well after the training was completed (anywhere from 2 weeks after to 6 months after). The University of Missouri’s Education and Consulting department uses class evaluations and comment sheets. The Education and Consulting department also has some instructors go out to the
work sites and observe the students 3-4 weeks after the course had ended using an "objective" evaluation form. A research and development firm was found that conducted post surveys 3-6 months following the training session. The post surveys asked what the training participants had done with the techniques, skills, and knowledge they were supposed to obtain, if anything. The post surveys also asked questions to find out what the trainees needed to know to make optimal use of the information and training they were given. The information was then used for reporting success stories for planning the next phases of training.

The data analysis revealed that some companies used participant feedback to measure results, as well as feedback by the participant's managers after the employee has used the new technology for awhile.

Generis Corporation, a company based in France that specializes in systems integration, consulting, training and documentation for the pharmaceuticals industry, has a cumulative set of evaluation exercises that are measured against a "100% result." The exercises that are utilized have been developed from the beginning of training, so that as soon as training objectives are defined, exercises are built into the training to test the achievement of them. Generis feels that this has worked for them to highlight areas of training that need to be improved or amended and also to identify students who might require more help. Generis also performs an evaluation of effectiveness of the job after several months of use, including how many times the participants require help (use of the help desk, manuals, or other users, etc.)
On the other side of evaluating training, responses ranging from evaluation not being included in training budgets (when training budgets are in place) to measurements of success as “Unknown at this time”.

Testing. An obvious way to measure training participants’ performance is to test them in the high-level technology computer skills they should have acquired during the training process. One IT company scored their training participants’ results at the end of their CBT training with an on-line test. Other companies used mastery tests, exercises and review questions. The University of Missouri’s Education and Consulting department has a “no pass, no password” policy in effect. It does not give passwords for employees to use the live system until they have passed a mastery test. An accounting and financial services firm responded that if training is for a specific programming or design skill, they administer a test for that specific skill for a mastery level. One company also used testing as a way to gain feedback about the course as well as the success of the training.

Participant Practice. The third category in evaluating the effectiveness of a training program is to allow the participants to practice. It was detected that the respondents conducted practice exercises for their participants in many ways. The list includes: frequent short review sessions, process/role simulations, workshops at the end of the course, allowing the trainees to demonstrate their competence in the classroom by teaching it, and offering end users exercises and review questions.
One international consulting firm uses a process/role simulation for critical roles to test ability to perform their job in a simulated work environment. While another, Information West Ltd., a training consulting company in Vancouver, makes sure that their class participants have demonstrated their competence while they are still in the classroom. The Strong Group, a consulting and IT services firm in the NY metropolitan area, utilizes mini-projects to offer their trainees practice, and they have found that it is a good way for them to evaluate whether any learning has actually occurred. Another consulting firm structures student activities and other practice exercises to mirror real-life situations. They used real source documents, and determined any gaps in the transfer of training through close classroom monitoring. A software and services firm holds workshops at the end of a course. It is stated that this allows the instructor to evaluate whether the student has learned the necessary skills to perform his or her job. Another respondent, a manufacturing firm, asks their trainees to complete practice exercises during class to make sure they can complete the process that is being taught and also have training rooms set up at all of their locations where end-users can practice using the system. Since the “go-live” date for this firm has been delayed, frequent short review sessions are being offered as well to ensure that transfer of training is not lost. Perhaps a consulting firm located in the Eastern United States does the most accurate practice measurement; this firm really grasps the importance of understanding not only how to operate the ERP system, but also how the system itself works. By the end of the training their participants are able to describe a business process and its
related tasks, as well as a description of their data sources, how the data they create is used, and how their business functions fit into the whole ERP system.

**Monitoring performance.** Monitoring the participants' performance both during the training and after the training was a technique utilized by some of the respondents. Managers of the participants were also encouraged to give feedback on their employees' performance after they had returned to work at one company. One firm simply stated that if participants can use the system accurately, then the firm knows that training was successful. Another firm monitors the number of transactions that are being performed, and when someone was noticeably slower, a call is put into management.

As stated in a previous section, a consulting firm, in combination with classroom exercises and student activities, has used monitoring the trainees during the class. An IT firm uses both the managers' feedback on their employees' performance three months later, as well as improvements on the employee's performance ratings. Vodden Consulting in Ontario, Canada stays on-site starting with 5 days a week, then 4, 3, 2, 1, as the need is indicated. Vodden Consulting can gain a solid knowledge of the performance effectiveness of users in this manner.

Finally, an ERP software vendor has its training instructors check on how well their participants are doing during the classroom exercises so find out problems can be identified.

**Coaching.** The last method of measurement is coaching. A few of the respondents answered this question by stating that they continuously coach the employees on the use of the system. In one case, the trainees were offered
coaching throughout the training sessions, while at other companies, follow-up coaching was offered. One company that responded has its participants coach others to perform tasks on-line and ask questions about the nuances of using the system in specific business circumstances.

**Benchmarking Other Companies for Training Delivery Methods and Practices**

Based on the responses received from the survey, little to no benchmarking was found with respect to training delivery methods and practices for high-level computer technology training. A manufacturing firm that responded went as far as to say that they had trouble finding other companies who had completed SAP training. They were able to find several copies of training materials and used them to benchmark their documentation template. The data denotes that government agencies benchmark with other government agencies in their area.

Chapter 4 documents the results of the survey. Here development times and delivery methods for organizations in six industry classifications were reported. In addition, how organizations measure whether or not the training was successful is discussed. Finally the amount of benchmarking that companies are doing with respect to their training delivery methods and practices was presented. Chapter 5 offers conclusions, recommendations, and discusses the implications of these results.
Chapter 5

Conclusions

This pilot study aimed to identify development times and training delivery methods for high-level computer technology training programs with respect to Enterprise Resource Planning (ERP) systems in North America. Chapter 5 provides a brief review of the pilot study, a discussion of the implications, and presents recommendations for further research.

High-level computer technology training can be defined as training for both end-users and Information Technology professionals on computer applications and systems (Anonymous, 1998). The survey questions were created with this definition in mind and an invitation to participate in the survey was sent to North American companies that would be likely to conduct this type of training. In defining ERP systems for clarification as to the types of questions that needed to be asked, two common themes emerged: (1) the Year 2000 problem (DeVoe, 1995; IBM, 1999; Jones, 1997; Kappelman, 1998; Kirsner, 1998; Stamps, 1998; Stanglin & Ahmad, 1998; White, D., 1998; & White, K. 1996) and (2) the computer language of COBOL (Callaway, 1997; Garnet, 1999; IBM, 1999; & McFarland, 1998). In addition to defining ERP systems, research was conducted to discern what different training delivery methods for high-level computer technology existed (Anonymous, 1998; Benson, 1997; Comaford, 1997; Dahmer, 1994; Ganger, 1994; Glade, 1998; Houser, 1992; Monitor, 1998; Munger, 1996; & Stevens & Stevens, 1996). The list of training delivery methods
that were listed on the survey was partially drawn from what was revealed in the literature researched (Schatz, 1996; Selden, 1996; Torode, 1998; & Zemke, 1997).

Question three of the survey indicated that "Hands-on" was the most widely used training delivery method followed by Lecture. It was found that question four and five of the survey also indicated this same finding. However, hands-on as most popular is contrary to what was revealed in the literature researched (Anonymous, 1998). In reviewing the data analysis for a relationship between the variables of industry classification and "Hands-on" to explain this phenomenon, none was found. The search for explanation was taken a step further and it was noted that 42% of the respondents were from the Consulting/Training industry.

Implications

As the literature review indicates, it is not uncommon for companies to combine training methods when delivering high-level technology training (Broadbent, 1998). An example of combined methods would be to hold an instructor led course that is followed by a hands-on or computer-based practice session (Schatz, 1996). The results of the survey indicate that instructor-led, coaching, and on-the-job training are the most standardized training delivery methods and that hands-on and lecture are the most commonly used instructional methods. Standardization, in this context, is defined as the least variant among the various ratios of development time to given hour of training. What does this finding mean for training professionals? The most widely accepted method of employee training is still classroom instruction. However, because of the way ERP systems alter the roles of the Human Resource and Information Systems
departments, trainees will need to have actual experience with the ways that the
data they are going to utilize and create affect the ERP system. This change in
these roles implies a trend toward more use of computer-based and role/process
simulation training. As a result, an increased need for computer-savvy Human
Resource Development professionals will emerge. Furthermore, the survey
results also indicate that there is a large amount of variance in the development
times for even the most commonly used training delivery methods. This finding
indicates that Human Resource Development professionals will have to draw
more strongly upon their ability to be flexible and creative in their approach to
training.

Questions for Further Research

It would be interesting to revisit the topic of Y2K within two years to see what
its impact will truly be on daily operations. It is likely that due to an increased
awareness, much more information will be made available on the topic of training
development and delivery for these systems. This pilot study adds to the body of
knowledge by having a role in increasing that awareness. It is recommended that
a follow-up study be conducted. Here, those respondents who currently are
unable to measure the success of their training delivery methods and practices
should be contacted. The questions to ask would be what are their success
measures and how much benchmarking is being done.

The Year 2000 looms like an ominous cloud of uncertainty with respect to
how society will function on a daily basis when Jan 1, 2000 arrives. Equally
uncertain is the way future training will be affected by the increased demand for
ERP systems and people who can accurately and efficiently utilize these systems.

In this world of "high-tech", "want it now", "information overload" that is emerging, who knows what the fallout will be. Jeremy Rifkin (1995) presents an intriguing picture of how the future of work will look. Rifkin (1995) states in his book *The End of Work*:

> Increasingly, computers are providing needed information and helping to structure the coordination and flow of activity in the economic process, eliminating the need for salespersons, account executives, truck drivers, warehouse handlers, shipping personnel, and billing department people.

(p. 105)

With that thought in mind; is it any wonder that businesses are rushing to implement ERP systems that can manage all of that information? And; Is it any wonder that they are in dire need of personnel to run the systems accurately? HR professionals should dust off those computer books, take a new class, and gain proficiency in ERP systems. It may become necessary for the survival of their discipline.
References


Baan web site. URL http://www2.baan.com [3/12/99].


Kirkpatrick, D. (1998). The E-Corporation: the e-ware war SAP is out to rule the market for business software. A host of smaller companies are in its way. *Fortune 102 +* [WWW document]. URL http://www.elibrary.com/s/edumark/...303637@library_I&dtype=0-0&dinst=0


Appendix A: ERP Market Share by Vendor

This chart describes ERP market share.
Appendix B: Evolution of ERP

Industry Trends:
• ERP
• Re-engineering
• Client-server
• Windows NT
• Integration

ERP Trends
• Functionality
• Migrate to Enterprise Solutions
• Client/Server Windows Environments
• Modular Approach
• Reduce Implementation Investment
• Rapid Systems Development
• Business Re-engineering

This appendix shows the evolution of ERP systems. Source http://www.erpdo.com/whyerp.htm
Appendix C: Survey

1. Does your company develop training programs for high-level computer technology (i.e., SAP, Oracle, and PeopleSoft)?

2. Does your company develop training programs for high-level computer technology (i.e., SAP, Oracle, and PeopleSoft)?

- Instructor-Led Classroom Training
- Web Based Training
- (CBT) Computer-Based Training
- Paper Based Self-Paced Training
- Coaching
- (OJT) On-The-Job Training
- Electronic Performance Support Systems
- Other - Please Specify

3. How is the time for development allocated? Please indicate the percentage of time allocated for each area.

<table>
<thead>
<tr>
<th>Development Component</th>
<th>Percentage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td></td>
</tr>
<tr>
<td>For Development</td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td></td>
</tr>
<tr>
<td>Other - Please Specify</td>
<td></td>
</tr>
</tbody>
</table>

4. What instructional methods do you use? Please respond to all that apply with percentages if available.

- Lecture
- Demonstrations
- Hands-on Practice Exercises
- Mastery Tests
- Review Questions
- Other Please Specify ________________

5. Did your company benchmark other companies for these training methods and practices? If "yes", which ones? Please list.

6. What is your geographic location?

7. What is your industry classification?

8. How do you know if your delivery method(s) and instructional practice(s) are successful?

9. May we reference the name of your company in the results report of this pilot study? If "yes", please provide written permission* via this e-mail address: ghrd590@hotmail.com or by fax at 716/385-8344 attention Graduate Human Resources Development Project Team. All other responses will be kept confidential.

10. May we contact you for future follow-up regarding this survey?

11. Please provide the following contact information:

   Company Name:
   Contact Name:
   Postal Address:
   E-mail Address: Phone Number:
   Fax Number:
Appendix D: Invitation Letter

We are a group of three graduate students from St. John Fisher College in Rochester, New York conducting a research project regarding training issues associated with Enterprise Resource Planning Systems within North America. We are looking for assistance in finding answers to our questions. We hope you will agree to participate in this important investigation. The following paragraph offers an introduction to the pilot study:

With the increasing demand for the installment and utilization of Enterprise Resource Planning Systems in business and industry (i.e., SAP, Oracle, PeopleSoft), the need to conduct research for training and development issues around high level computer technology training is imperative. Specifically, problems with regard to instructional methods and delivery practices are on the rise. This pilot study focuses on instructional models that are used for the development of training programs for high-level computer technology (i.e., training delivery methods, instructional methods, materials development time allocation, and development ratio).

We would like to hear from you! Please e-mail us at ghrd590@hotmail.com and request our survey. The survey should only take a few minutes of your time and we will be sharing the results with everyone who participates. Thank You!

Please advise of your interest in order for us to forward our survey to you. Thank You!

The Graduate Human Resources Development Program Project Group
e-mail Address: ghrd590@hotmail.com
Organization: St. John Fisher College
Location: Rochester, New York
Phone: 716/385-8157 or 716/416-2945
Fax: 716/385-8344 attention: Graduate Human Resources Development Program Project Group
Postal Mail Address:
Graduate Human Resources Development Program Project Group
C/O Dr. Marilynn Butler
Program Director/Department Chair Graduate Human Resources Development
3690 East Avenue
Rochester, NY 14618
Appendix E: Follow-up Letter

You recently requested a copy of a survey from the Graduate Human Resource Program Project Group concerning training issues associated with Enterprise Resource Planning Systems. Our records indicate you have not submitted a completed survey. If you would still like to complete a survey please do so below. Please respond as soon as possible. Thank You!

Thank you for agreeing to participate in our survey about training programs for high-level computer technology. Your response is extremely important to us. The value of this pilot study is directly related to the number of individuals who return completed surveys. Therefore, your participation is of great importance. Please take the time to complete and return the survey. If you know of someone else who may also be interested in participating in this pilot study please pass the survey on to him or her. On behalf of the Graduate Human Resources Program Project Group we sincerely thank you for your part in this important effort. Please remit your completed survey to the following e-mail address:
ghrd590@hotmail.com

The objective of this survey is to investigate and seek answers regarding training issues associated with Enterprise Resource Planning Systems.

(See Appendix C for actual survey)
### Appendix F: Geographical Location Summary

<table>
<thead>
<tr>
<th>Geographic Region</th>
<th>Abbreviated State Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>New England</td>
<td>(CT, ME, NH, RI, VT)</td>
</tr>
<tr>
<td>Middle Atlantic</td>
<td>(NJ, NY, PA)</td>
</tr>
<tr>
<td>South Atlantic</td>
<td>(DE, FL, GA, MD, NC, SC, VA, WV)</td>
</tr>
<tr>
<td>East North Central</td>
<td>(IL, IN, MI, OH, WI)</td>
</tr>
<tr>
<td>West North Central</td>
<td>(IA, KS, MN, MO, NE, ND, SD)</td>
</tr>
<tr>
<td>West South Central</td>
<td>(AR, LA, OK, TX)</td>
</tr>
<tr>
<td>Mountain</td>
<td>(AZ, CA, ID, MT, NV, UT, WY)</td>
</tr>
<tr>
<td>Pacific</td>
<td>(AK, CA, HI, OR, WA)</td>
</tr>
<tr>
<td>Canada</td>
<td></td>
</tr>
<tr>
<td>International</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>Non-North American based respondents</td>
</tr>
</tbody>
</table>