Considering the Customer: Determinants and Impact of Using Technology on Industry Evolution

by

Steven J. Kahl

B.A. Philosophy/Senior Fellow
Dartmouth College, 1991

M.A. Philosophy
University of Minnesota, 1995

SUBMITTED TO THE SLOAN SCHOOL OF MANAGEMENT IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2007

© 2007 Steven J. Kahl. All Rights Reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of the thesis document in whole or in part in any medium now known or hereafter created.

Signature of the Author: ____________________________

MIT Sloan School of Management
May 16, 2007

Certified by: ____________________________

Michael Cusumano
Sloan Management Review Distinguished Professor of Management
Thesis Supervisor

Accepted by: ____________________________

Birger Wernerfelt
Professor of Management Science
Chair, PhD Program, Sloan School of Management
Considering the Customer: Determinants and Impact of Using Technology on Industry Evolution

by

Steven J. Kahl

Submitted to the Sloan School of Management on May 16, 2007 in Partial Fulfillment of the Requirements for the Degree of Doctorate of Philosophy in Management

ABSTRACT

This dissertation raises two questions: How do customers come to understand and use a technology? What is the influence of customers using a technology on industry evolution and competition? I use two historical cases to answer these questions. The first considers how insurance firms came to initially understand and use the computer in the 1940s and 50s (in collaboration with JoAnne Yates). We argue that the understanding of a new product category is related to which comparisons are actually made and why. Insurance firms made multiple comparisons that collectively helped distinguish the computer from any given pre-existing category. The second case analyzes the history of how U.S. manufacturers used manufacturing planning software from its inception in 1954 through its first significant technological change in the 1990s. Manufacturers significantly changed how they used the software primarily through learning by using the software. Market intermediaries, such as consultants, played an important role in diffusing these uses and in developing a collective understanding of the technology – what I call a dominant use. Using additional entry/exit data from the software industry, I argue that how customers use a technology helps explain the industry’s population dynamics. I argue that the emergence of a new dominant use, even without a corresponding technological change, can lead to significant new market entry. In addition, established firms are less likely to fail when customers use a new technology in similar ways to how they used the old technology. More generally, firms – established or not – are more likely to survive the introduction of a new technology if they are able identify and adapt their products to how customers develop uses for that product. The dissertation concludes by considering the scope conditions of this argument and further theoretical and managerial implications of more explicitly treating the processes through which customers use a technology.

Thesis Supervisor: Michael Cusumano
Title: Sloan Management Review Distinguished Professor of Management
Table of Contents

Acknowledgements ........................................................................................................ 9

Chapter One: Considering the customer, an introduction to the research ................. 10
  1.1 Motivation of research question ........................................................................... 11
  1.2 Main argument .................................................................................................... 15
  1.3 Scope conditions .................................................................................................. 18

Chapter Two: Conceptual underpinnings of customer use ........................................ 21
  2.1 Customer use: Understanding, justification, deployment ...................................... 24
  2.2 Learning by using ................................................................................................ 32
  2.3 Dominant use ....................................................................................................... 37

Chapter Three: Creating a category: The introduction of the computer into the insurance industry, 1947 – 1960 (with JoAnne Yates) .................................................... 40
  3.1 Introduction .......................................................................................................... 40
  3.2 How insurance firms created a new product category for the computer, 1940’s – 1960s ................................................................................................................. 46
    3.2.1 What makes the computer a new product category ........................................ 49
    3.2.2 The insurance firms’ use of tabulating equipment ........................................... 52
    3.2.3 Pre-adoption categorization – The computer as tabulator and brain ............ 53
    3.2.4 Early insurance adoption and use of the computer ......................................... 59
  3.3 Conclusion ............................................................................................................ 66

Chapter Four: Dominant use: The social evolution of U.S. manufacturing’s understanding and use of planning software, 1954 – 1996 .................................................. 68
  4.1 Introduction .......................................................................................................... 68
  4.2 Data Sources ........................................................................................................ 70
  4.3 The setting: U.S. manufacturing and planning software ........................................ 73
  4.4 1950s: A “Practical Approach” to using planning software .................................. 75
  4.5 1960s: Alternatives to the “Practical Approach” ................................................... 88
  4.6 1970s: The MRP Crusade – establishing a dominant use .................................... 105
    4.6.1 The Minicomputer and a new market segment .............................................. 109
4.6.2 Pre-packaged software ........................................ 113

4.7 1980s: Variations of a dominant use ................................................. 124

4.8 1990s: Client/Server and a new dominant use, ERP .................................. 137
   4.8.1 The technological discontinuity: From host-based to client/server architectures ........................................ 140
   4.8.2 Changes in Intermediators: Industry Analysts, Consultants, and BPR .................................................. 144
   4.8.3 Different customer responses to client/server .................................. 149

4.9 Conclusion ........................................................................ 154

Appendix A to Chapter Four – A brief overview of U.S. Manufacturing ............. 162
   4A.1 Manufacturing operating processes and use of technology pre-computer ........................................ 165
   4A.2 Macro factors influencing the manufacturing market ........................................ 169

Appendix B to Chapter Four – A brief overview of software technology ............ 171

Appendix C to Chapter Four – Key historical events in evolution of use .......... 176

Chapter Five: Radical incrementalism: The impact of customer use on industry evolution ................................................................................. 181
   5.1 Introduction ........................................................................ 181
   5.2 Entry/Exit patterns of planning software industry ........................................ 182
   5.3 Radical incrementalism ........................................................................ 200

Appendix A to Chapter Five – Coding of entry/exit patterns ....................... 207

Chapter Six: Conclusion ........................................................................ 211
   6.1 Implications for theory and research ............................................. 211
      6.1.1 Industry-level selection and firm-level adaptation ................. 211
      6.1.2 Diffusion ........................................................................ 214
      6.1.3 Technological Standardization ............................................ 215
   6.2 Managerial Implications ........................................................................ 217
   6.3 Conclusion ........................................................................ 219

References ........................................................................ 222
Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.1</td>
<td>Functional specificity and performance uncertainty of select technologies</td>
<td>20</td>
</tr>
<tr>
<td>2.1</td>
<td>The different components of customer use</td>
<td>32</td>
</tr>
<tr>
<td>4.4.1</td>
<td>GE's understanding, justification, and deployment of computer applications, 1954</td>
<td>82</td>
</tr>
<tr>
<td>4.6.1</td>
<td>APICS membership, select years 1962 – 1980</td>
<td>105</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Establishing MRP as the new dominant use</td>
<td>108</td>
</tr>
<tr>
<td>4.8.1</td>
<td>Large and small manufacturers’ changes in use of software in response to client/server</td>
<td>140</td>
</tr>
<tr>
<td>4.8.1.1</td>
<td>Differences between host-based and client/server architectures</td>
<td>141</td>
</tr>
<tr>
<td>4.8.2.1</td>
<td>Development of BPR</td>
<td>149</td>
</tr>
<tr>
<td>4.9.1</td>
<td>Large manufacturer use of manufacturing planning software</td>
<td>155</td>
</tr>
<tr>
<td>4.9.2</td>
<td>Comparison between small and large manufacturers’ usage patterns</td>
<td>158</td>
</tr>
<tr>
<td>4.9.3</td>
<td>Comparing usage pattern with technological change pattern</td>
<td>159</td>
</tr>
<tr>
<td>4B.1</td>
<td>Software technology stack</td>
<td>173</td>
</tr>
<tr>
<td>4B.2</td>
<td>Some significant technological changes in the software technology stack</td>
<td>174</td>
</tr>
<tr>
<td>4B.3</td>
<td>SAP’s user interface</td>
<td>175</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Entry/exit pattern for U.S. manufacturing planning software market, 1954 – 2003</td>
<td>183</td>
</tr>
<tr>
<td>5.2.2</td>
<td>U.S. manufacturing software market, 1983 – 2003</td>
<td>187</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Various categorizations of technological change</td>
<td>192</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Entry/exit patterns by customer segment served, 1954 – 1996</td>
<td>196</td>
</tr>
</tbody>
</table>
Figure 5.3.1  Use and technical aspects of technological change ............ 201

Figure 5.3.2:  Different changes and outcomes in the U.S. manufacturing planning market ......................................................... 205
Table of Tables

Table 3.1: Comparative features of IBM’s electronic data-processing equipment .......................................................... 61

Table 4.1: Changes in large manufacturers’ use of planning software, 1954 – 1996 ....................................................... 69

Table 4.4.1: Some computer applications developed in U.S. manufacturing, 1954 – 1957 ........................................... 77

Table 4.5.1: Comparison of “Practical Approach” with new MRP use ................................................................. 94

Table 4.6.1: Early adopters of MRP systems ................................................................. 106

Table 4.6.1.1: U.S. Manufacturing Establishments by number of employees, select years, 1947-2002 ........................................ 110

Table 4.6.1.2: Comparison between large and small manufacturer use ................................................................. 113

Table 4.7.1: Article count of key APIC topics, select years 1962 – 1989 ................................................................. 126

Table 4A.1: Distribution of U.S. Discrete/Process Manufacturing, select years, 1950-2000 .................................................. 164

Table 4A.2.1: Various U.S. Business Cycles, 1953 – 1996 .................................................................................. 170

Table 4C.1: Summary of key historical events ......................................................................................... 176

Table 5.2.1: Shipments of different kinds of computers, 1975 – 1990 ............................................................. 185

Table 5.2.2: U.S. business spending in software, 1970 – 2002 ................................................................. 186

Table 5.2.4: Comparison of SAP with Dun & Bradstreet ................................................................. 199
Acknowledgements

Being trained as a social scientist, I have learned about the value of social interaction and the significance of the context in which one works. This dissertation and my own intellectual development have benefited greatly from studying at MIT in Cambridge, Massachusetts. I am grateful to my dissertation committee – Michael Cusumano, Jesper Sorensen, Michael Tushman, and Ezra Zuckerman. Their constructive criticism and high bar for academic achievement challenged and pushed my thinking. I am also indebted to JoAnne Yates, with whom I collaborated to write Chapter 3. Working with her was a true pleasure and learning experience. I look forward to future collaborations. I am also grateful for other faculty who took interest in my work and offered helpful critiques: Erik Brynjolfsson, Rebecca Henderson, Eric Von Hippel, Wanda Orlikowski, Ed Roberts, Merritt Roe Smith, Freddy Suarez, James Utterback, and Chris Wheat.

The graduate students at MIT help make it a great place to study. John Paul Ferguson challenged my assumptions and pushed me to clarify my arguments. Sinan Aral (now at NYU), Sung Joo Bae, Rodrigo Canales, Karim Lakhani (now at HBS), Rafael Lucea, Melissa Mazmanian, Ethan Mollick, Ramana Nanda, Tim Quinn, Brian Rubineau, Lourdes Sosa (now at LBS), and Yanbo Wang all offered invaluable advice and support.

This dissertation could not have been done without access to archives, interviews with various participants in the history of manufacturing software, and generous funding. IDC and AMR graciously provided access to their archives of industry reports.

Finally, I would like to thank my family, in particular Suzie, Henry, and Hattie and their grandparents, for keeping me sane and grounded throughout this process. Switching careers is a big change, and I appreciate their sacrifice and support. I look forward to our next adventure.

My sincere appreciation.
Chapter One

Considering the customer, an introduction to the research

Broadly stated, my research examines the relationship between technological change and the population dynamics of the industries that supply these technologies. I explore how technological change impacts the number and types of firms we observe within an industry. This phenomenon is well researched – dating back to at least Schumpeter (1934; 1942) and stretching across a wide variety of disciplines, including industrial economics (Sutton, 1998), evolutionary economics (Nelson & Winter, 1982), organizational sociology (Hannan & Freeman, 1977), technology management (Christensen, Suarez, & J., 1999; Utterback, 1994), sociology and history of technology (Bijker, 1995; Landes, 1983), and marketing (Chandy & Tellis, 2000).

I hope to contribute to this research stream by more systematically considering how customers come to understand and use a technology as it matures in a market. Recognizing that customers must interpret and actually use a technology certainly is not a novel insight. Since at least Gilfillan (1935), scholars have recognized that the local adoption of a technology is important to technological development. However, much work in industry evolution has not explicitly considered how customers use a technology as an independent variable that helps explain industry dynamics. But, why should we care about
how customers use a technology? Does it really help us explain the relationship between technological change and industry population dynamics?

1.1 Motivation of research questions

A substantial body of research has shown that some significant technological changes can induce wholesale "creative destruction" (Schumpeter, 1942). New firms enter the industry by developing the new technology, while incumbent firms often struggle with the technological transition and exit. Scholars have observed creative destruction in a wide variety of industries and technologies: automobiles, typewriters, televisions, transistors, and the computer (Suarez & Utterback, 1995; Utterback, 1994), airlines, cement, and photocopying (Tushman & Anderson, 1986), photolithography (Henderson & Clark, 1990), and disk drives (Christensen & Bower, 1996). However, there are other examples in which even though the industry’s technology significantly changes, wholesale "creative destruction" does not occur (Abernathy, Clark, & Kantrow, 1983). In these cases, incumbent firms successfully transition through significant technological changes. NCR did in cash registers (Rosenbloom, 2000), Intel did in microprocessors (Burgelman, 1994), and Mergenthaler did in typesetting (Tripsas, 1997).

These examples suggest that we have not completely specified all the relevant factors and conditions that explain the competitive significance of technological change. Historically, explanations of "creative destruction" focused on factors relating to the production of the new technology. Incumbents either lack the incentives to invest properly
to make the necessary technological changes (Reinganum, 1983) or they are incompetent in their ability to develop the new capabilities or routines necessary to create the new technology (Henderson & Clark, 1990; Nelson & Winter, 1982; Tushman & Anderson, 1986). According to these explanations, survival issues lie with the producers of the technology and their inability to produce and market the new technologies.

However, recent research suggests that customer related factors may also play an important role in determining the competitive significance of technological change. Rather than just focus on how firms produce new technologies, this approach considers how customers’ different preferences or experiences with the technology shape the competitive landscape. Intuitively, expanding the scope of analysis to include the customer perspective makes sense. After all, the purchase and use of a technology make innovations and industries viable. Customer evaluations of technology affect a firm’s incentive to innovate and the industry’s profitability (Nelson & Winter, 1982; Schumpeter, 1934).

Generally speaking, there have been two distinct approaches that relate customer factors to technological change and industry dynamics. One approach focuses on the structural aspects of the market, in particular, the differences between the preferences of different customer segments (Adner, 2002, 2004; Christensen, 1997; Christensen & Rosenbloom, 1995). These explanations assume that preferences within a segment remain relatively static, but the overall composition of the market segments may change. The entry and exit of different customer segments with different preferences and uses of technology helps explain the mortality rates of incumbent firms. Thus, this work pays less
attention to the individual customer’s experience with the technology and more to the
structure of the overall market for a technology.

The second approach focuses more upon the changes within a customer segment (Clark, 1985; Rosenberg, 1983; Tripsas, 2006; Von Hippel & Tyre, 1995). This perspective concentrates on how customers interact with the technology. Through these interactions, customers develop an understanding of the technology (Orlikowski, 2000), make innovative changes to the technology (Von Hippel, 1988), or even learn new unforeseen uses of the technology (Rosenberg, 1983). Thus, unlike the market structure approach, this perspective is very much interested in how particular customers change their use of technology over time. Typically, research in this tradition address questions about the rate and direction of technological change. For example, Yates (1993) observed that changes in how insurance firms used the tabulating machine in the early 1900s led to significant technological changes in the machine itself. One of the goals of this dissertation is to apply some of these insights about how customers develop and change their understanding and use of a technology to questions about the population dynamics of the industry that supplies the technology.

The primary question of this dissertation asks how does customer use impact industry evolution and competition? More specifically, how do the processes of customers coming to understand what a technology is, using the technology in a particular way, and changing their use over time influence the number and types of firms we see in an industry?
To answer this question requires more systematic analysis of how customers come to understand and develop uses of a technology. Because previous research into customer use has been more interested in how it relates to specific periods of technological change, it has been less concerned about how the process of customer use unfolds (for a notable exception see (Tripsas, 2006). Why, when, and how customers may change their use and understanding of a technology remain less understood. For instance, it is entirely possible that customers change their use of a technology only when there is a significant technological change. Or, one could explain such changes simply in terms of a temporal lag. It takes time for customers to learn new uses inherent in the technology itself. If this is the case, factoring in the process of how customers use a technology over time may not really increase our explanatory power of industry population dynamics.

Although certainly true in some cases, it is important to recognize that the impetus for a new use does not necessarily come from a discovery of a latent characteristic of the technological artifact. Rather, it is through the process of putting the technology into use within a particular context that customer learn about its performance characteristics and develop new uses (Orlikowski, 2000; Rosenberg, 1983). For example, the discovery to use Botox to temporarily remove wrinkles occurred only after physicians used the drug during the clinical trial process for treatment of eye disorders. There is also evidence that a change in customer use can lead to technological change in the industry. In the typesetter industry, some newspapers extended their use of typesetters beyond the traditional printing of text to include images, precipitating a shift from hot metal typesetting to analog phototypesetting (Tripsas, 2006).
The concept of customer use remains unclear. We do not have a deep understanding of the processes through which customer use evolves and how this process influences the competitive dynamics of an industry.

1.2 Main argument

I address the research questions in two ways. First, I try to clarify the concept of customer use by identifying what it entails and how it may change over time. To do this, I integrate insights and concepts from existing literatures that study how users purchase and deploy a technology. I argue that customer use consists of how customers understand what the technology is, justify its purchase, and actually deploy the technology. The relationship between these different aspects of customer use is dynamic. Initial understanding and purchase justification may shape initial uses of the technology, but on-going use can generate new knowledge that changes future understandings and justifications. I also argue that technological changes alone do not determine changes in customer use. Other sources of change include organizational factors, social actors such as market intermediaries, and learning from experiences with the technology. Based on these additional factors, the process of changes in customer use should be decoupled from the process of technological change. How customers use a technology is an independent, but related process to technological change.

Empirically, I present two historical cases that analyze how customers understand, justify, and deploy a technology as the technology matures. Historical case
studies provide some advantages over cross sectional or even quantitatively focused longitudinal research. Historical analysis is particularly relevant since I am primarily interested in explaining the process of how this new understanding and use developed. It is important to get the record of customers’ experiences and uses of a technology straight just as it is important to get the record of how the technology is produced straight. My historical analysis focuses on technologies used by organizations, not individual consumers. This was intentional because organizations document their justification of the purchase and how they intend to use the technology. In addition, the organizational complexity adds an intriguing dimension to how the use gets determined. A drawback of this focus is that these insights may not apply to individual consumers because they purchase and use technologies in different ways than organizations. In the dissertation, I address these broader scope concerns after my historical analysis.

The first historical case considers how insurance firms initially understood and used the computer from the late 1940s through the early 1960s. JoAnne Yates did the primary historical analysis on this case and we collaborated on its application. The second case considers how U.S. manufacturers used manufacturing planning software from its inception in 1954 through a significant technological change in the 1990s. The insurance case focuses on the commercial introduction of a new technology and how customers develop an initial understanding and use. To create a product category for the computer, insurance firms made multiple comparisons: with tabulating machines, an existing technology heavily used at the time, and with the human brain. These comparisons, taken collectively, helped distinguish the computer from any given pre-existing category. The
interplay between producers and customers, mediated by third parties, as well as historical usage patterns of existing technologies, influence which comparisons are made and emphasized. Consequently, understanding of the new category is related to \textit{which} comparisons are actually made and \textit{why}; this understanding, in turn, plays an important role in the viability of a new category.

The manufacturing case focuses on the dynamic aspects of use – how manufacturers changed their use over time. The initial understanding and use of a technology often changes. After extensive use of the software, manufacturers learned that how they currently deployed the software did not yield the desired return. In some cases, this learning ironically extended the life of the software and in others it helped identify alternative uses. Market intermediaries played an important role in diffusing this learned knowledge and in developing a collective understanding of the technology – what I call a dominant use.

I then apply these insights about the patterns of customer use to argue that how customers use a technology is an important variable that helps explain the competitive outcomes of technological changes and industry evolution. In particular, I argue that the emergence of a new dominant use, even without a corresponding technological change, can lead to significant new market entry. In addition, established firms are less likely to fail when customers use a new technology in similar ways to how they used the old technology. More generally, firms – established or not – are more likely to survive the introduction of a new technology if they are able to identify and adapt their products to
how customers develop uses for that product. These insights help explain the initial anomaly – why established firms may persist through significant technological changes.

The structure of the dissertation roughly follows this argument, concluding with thoughts about the boundaries of this analysis and additional theoretical and managerial implications.

1.3 Scope conditions

One of the drawbacks of using historical analysis is that it is difficult to generalize from the particular history. Even though I considered two different industries, insurance and manufacturing, the technology and time period overlapped significantly. There may have been something peculiar to the computer and computer applications or to the time period of consideration. Nevertheless as a means to identify what the scope conditions are it is instructive to think about what is particular to this technology that fostered learning by using and variety in use.

Two characteristics about computers and computer applications in particular stick out: the uncertainty about the technology performance and general purpose nature of the technology. If customers are certain about the operating characteristics of the technology before deployment, then we may not expect significant variation in use. Such uncertainty seems to be greatest when there is a high level of technical complexity and integration requirements with other technologies. High levels of organizational change to support the technology would also increase uncertainty about its performance. In addition, if the technology was functionally specific, we would expect less variation in use. However, this
is not as clear cut as uncertainty about performance, for even functionally specific
technologies can have a variety of uses. Recall the airplane engine being used as a fire
extinguisher. Nevertheless, based on these conditions, we would expect functionally
specific technologies with high degrees of performance certainty to exhibit less variation in
use and learning by using.

Figure 1.3.1 speculates where several different kinds of technologies may fit along
these dimensions. I am almost certain that I have misclassified several of these
technologies, but the exercise is helpful in identifying other technologies to try to test the
scope conditions of this case. For instance, even though pharmaceutical drugs may have
many applications, the government approval process limits legal use of the drug without
high degrees of certainty about its performance related to a specific function. Therefore,
pharmaceuticals may be higher in certainty of performance and lower in functional
specificity. Consequently, I would predict variations in use that change over time.
Additional work needs to be done to test the significance of these boundary conditions.
Figure 1.3.1: Functional specificity and performance uncertainty of select technologies

- High Functional Specificity
  - Pharma
  - Drugs
  - Airplanes
  - Disk Drives

- Low Functional Specificity
  - Software
  - Computers
  - Electricity

Certainty of performance

High

Low
Chapter Two

Conceptual underpinnings of customer use

Scholars distinguish technology from science on the grounds that technologies fulfill some set of practical purposes (Allen, 1997; Arthur, 2007; Heidegger, 1977). Technologies, like dams or aircraft engines, have concrete applications, like generating electricity or propelling aircraft; whereas, scientific discoveries define more basic principles and relationships. A central characteristic of technology is that it has a set of uses.

Consequently, various literatures, including the sociology of technology, technological strategy, and history of technology, have considered how customers use a technology and how this process impacts the rate and direction of technological change. During development, different customer expectations about how a new technology should be used can influence the initial technical design (Pinch & Bijker, 1987). During commercialization, customers must adapt technologies to local needs. This process can improve the technology in subtle ways (Gilfillan, 1935; Rosenberg, 1983; Von Hippel & Tyre, 1995) or significantly change the technology’s design (Clark, 1985; Danneels, 2002). In some cases, the locus of innovation shifts from the original producer of the technology to the user (Von Hippel, 1988, 2005). During the introduction of a new technology, different customer segments may develop different uses (Utterback, 1994). Producing
firms may focus their attention and resources on the wrong customer segment, significantly impeding the firms’ ability to respond to the technological discontinuity (Christensen & Rosenbloom, 1995). Over time, customers may learn new uses or requirements from using the technology, leading to significant technological changes (Tripsas, 2006; Yates, 1993).

Despite recognizing the importance of how customers use a technology, the existing literature has yet to rigorously define “customer use”. Appealing to customer use’s import without conceptual clarity is theoretically dangerous. In fact, by focusing on different aspects of customer use, the literature supports inconsistent views about how it functions. Some scholars, such as Yates and Rosenberg, consider the act of using the technology; whereas, others, such as Christensen, concentrate on how different “systems of use” represent different technological requirements. Christensen’s view maintains that use within a customer segment remains relatively static over time. Change in use primarily occurs when a new customer segment enters with a different system of use. In contrast, Yates’ and Rosenberg’s perspective supports the view that customers learn new uses. Change in use is not restricted to the entry and exit of new customer segments but also occurs within a customer segment through continued use of the technology.

As a result, important basic questions about customer use and how it functions in relation to technological change remain unexplored. What explains the differences in use between customer segments? Christensen simply tells us that different customer segments have different “systems of use”, but does not identify what aspects of these systems of use explain the differences. How does use within a customer segment change over time? Are there patterns to these changes like there are patterns of technological change? Do
dominate uses emerge just like dominant designs? If so, what does this pattern look like and what are the mechanisms that determine this pattern? How does using a technology relate to customer need and demand?

This chapter addresses these questions with the hope of bringing some conceptual clarity. The intent of the chapter is to begin to more concretely identify the core components of customer use, how these components relate to each other, and how they can change over time. I warn that this is just a beginning and is not meant to be conclusive. Some concepts probably will need refinement and other components may be identified. But, by taking a more systematic approach to defining customer use, this chapter integrates different strands in the literature, hopefully in a way that resolves some of the inconsistencies and exposes some erroneous assumptions. This exercise also establishes more concrete ways to measure what customer use is and whether it changes. The empirical part of the dissertation then leverages these measures.

Specifically, the chapter identifies three core concepts that define customer use: understanding what the technology is, justification for its use, and the actual deployment of the technology. Differences in understanding, justification, and deployment can explain differences between customer segments. These different aspects of customer use interact and mutually influence each other. How customers interpret a technology and justify its purchase shape how they initially deploy a technology. Following Rosenberg, customers can learn new uses and generate new requirements through their experiences of using the technology. Finally, the chapter argues that certain patterns of convergence and divergence in how customers use a technology can occur. In particular, just as there are
dominant designs in a technology’s architecture, customers can converge to a “dominant use” – a particular way of using a technology. However, such a convergence is not inevitable; other mechanisms must be present for a dominant use to emerge.

2.1 Customer use: Understanding, justification, and deployment

A logical place to begin to define customer use is to consider the act of using a technology, henceforth called deployment for clarity sake. A substantial literature has examined how customers deploy a technology and how this deployment can change over time. This literature has identified two important aspects of deployment: what functions the technology performs and how it is deployed within the use context.

Customers use a technology to perform a certain set of functions. For example, customers utilize televisions to gather information and to be entertained; passengers use airplanes to travel to distant locations more efficiently; firms use business application software to automate different business processes. Some technologies such as electricity or computers may be capable of performing a diverse set of functions, but other technologies such as disk drives perform a much more limited set of functions. Economists call the former types of technologies general purpose and the latter specific.

An interesting question is where does the functionality of a technology come from? At first glance, it seems logical that the technical design primarily dictates the technology’s use (Hargadon & Douglas, 2001). Even though an automobile emits significant heat, its
design restricts customers from using it to cook food. However, customers may use technologies in ways unintended by the technical design. This may occur even with functionally specific technologies. Consider the case of airplane jet engines. This technology appears to be functionally specific – jet engines are designed to propel aircraft. But, after the Kuwait war, a group of firefighters used jet engines as fire extinguishers to help put out oil fires (Oudshoorn & Pinch, 2003). In this case, the particular problem at hand, as opposed to the technical design, determined the technology’s functionality. As the sociology of technology literature is quick to point out, technical features alone do not fully determine how a technology is actually used (Orlikowski, 2000). Different problems may generate different deployments of the same technology, even if it was designed for a specific purpose.

Related to what function the technology actually performs is how the technology is deployed. This aspect of deployment includes who the actual users of the technology are, how often is the technology used, or how the technology operates within the organizational structure and processes. Barley’s (1986) famous CT scanner study has shown that how the technology is deployed is an important and different dimension from the functionality the technology performs. Different users may utilize the technology to perform the same function but deploy it in significantly different ways. In Barley’s case, two hospitals used CT scanners to perform the same function, but one hospital centralized organizational

---

1 This example implies that we may want to rethink the general/specific distinction as something inherent to the technology itself. Specific technologies may have more general applications depending upon the creativity of potential users and the problems they face. And, general technologies may be used in more specific ways than originally anticipated.
processes around its use of the scanners; whereas, the other decentralized its organizational processes.

Thus, the organizational context of where the technology is deployed also influences how the technology is actually used.² Within organizations, there may be multiple kinds of users with different requirements. For example, consider the use of a computer within a firm. Information technology practitioners use the computer quite differently than business users. IT specialists program applications, perform maintenance, and manage data; whereas, business practitioners use computers to complete work tasks and analyze information. In addition, who actually procures the technology may differ from who actually uses the computer. This was certainly the case with early mainframe purchases. The cost of the equipment typically required upper management approval, but this level in the organization had limited interaction with the computer (Yates, 2005).

Lastly, the new technology often must be integrated with pre-existing organizational processes, culture, and technologies. Bresnahan, Greenstein, and Saloner (1996; 1997) observed that the adoption of client/server technology in the early 1990s was heavily influenced by how much organizational processes were automated on the older mainframe technology. Since firms have different organizational structures, processes, and cultures, the organizational context into which the technology is used can lead to different deployments of the same technology.

These differences in technological deployment are important because they help shape the user requirements (Christensen & Rosenbloom, 1995; Dosi, 1982). The

---

² This is a significant difference between users who are individuals and users who are organizations. However, this difference is more in degree than kind. Individuals, like organizations, still need to integrate the technology into pre-existing environment, but the level of complexity tends to be less significant.
firefighters who use the jet engine as a fire extinguisher have a different set of needs for
the jet engine than aviators. Similarly, the two different hospitals in Barley’s study may
very well have different needs based on the differences in decentralized and centralized use
of the scanners. These requirements, in turn, influence how a customer evaluates a
technology. And, customer evaluations of technology affect the producing firm’s incentive
to innovate and the direction of the technological change (Dosi, 1982; Nelson & Winter,
1982; Schumpeter, 1934).

However, other factors prior to the actual deployment of the technology may also
influence how the technology is deployed. The technology implementation literature only
considers the actual deployment of the technology, and ignores activities done during the
investment decision process. From the marketing and economic literature, investment
requires some level of understanding what the technology is and justification of its
purchase. How customers understand and justify their investment in a technology
influences how they initially deploy it.

The marketing literature asserts that making an investment in a technology requires
some level of understanding what the technology is. Specifically, this understanding
entails categorizing the technology in relation to pre-existing product categories and then
comparing alternatives (Urban, Hulland, & Weinberg, 1993; Zuckerman, 1999). For
example, customers compared the automobile to the horse carriage, calling it a “horseless
carriage” (Clark, 1985). In fact, vestiges of this comparison remain today as an engine is
measured in terms of “horsepower”. But, what determines which product comparisons are
made? Often technologies potentially have multiple product comparisons. In addition to
horse carriages, an automobile could be compared to other forms of transportation such as trains. In fact, later adopters of the automobile began comparing automobiles to living rooms, calling them a “living room on wheels” (Clark, 1985). Why the automobile and living room were identified as the salient comparisons is an interesting and important question because different product categorizations create variety in how customers interpret and use the technology.

Producing firms try to influence how customers categorize technologies through making explicit comparisons in advertisements and incorporating specific design features (Hargadon & Douglas, 2001). However, other actors influence what product comparisons customers make. Market intermediaries, such as industry associations, consultants, industry analysts, and distributors, mediate market transactions between customers and producers. Ruth Schwartz Cowan calls this interaction space the “consumption junction” (Cowan, 1987), arguing that it heavily influences how customers adopt a technology. Often these intermediaries play the role of “product critics” (Zuckerman, 1999) and shape how customers categorize the technologies. In addition to these actors, customers may categorize technologies differently based upon local experiences and understanding.

To see these different factors at work, consider the early commercialization of the telephone. Telephone manufacturers initially compared the telephone with the telegraph and insisted that the phone should be used to conduct business as opposed to socialize (see Fischer, 1992; Kline, 2000 for more complete history). Urban customers made this comparison and adopted this use, but farmers, who generally did not use telegraphs, interpreted the technology as a means for social interaction. They compared the telephone
to activities like a porch side chat. Industry organizations, such as the USDA, also promoted such uses and interpretations of the telephone as well as helped establish organizations to support this more social use (primarily because the telephone manufacturers refused to do so). Against the intent of the telephone manufacturers, farmers primarily used the telephone for social purposes.

Therefore, how customers interpret a technology is embedded within a social process not completely under the control of the producing firms. Intermediaries with different agendas than producers and customers can help shape how customers categorize and interpret what the technology is. Customers may come to understand and use a technology in ways not intended by the producing firms. Consequently, the commercialization of a technology may take unexpected twists and turns.

Aside from understanding what the technology is, customers must also justify its purchase. In economic language, a customer must be willing to pay for a good in order to consume it. In general, justification entails calculating the potential benefit of using the good against the cost of its purchase and implementation. Where understanding entails making judgments about what the technology is, justification focuses more on what the technology does. For instance, one could measure the benefit/cost of purchasing an automobile. They may consider how they intended to use the automobile. For the sake of argument, imagine the intended use is to commute to work. The cost/benefit analysis probably would include a comparison with using public transportation, where the primarily benefit of the car could be flexibility but at a higher cost of ownership. Thus, customers have specific functions in mind to calculate the potential benefits and costs.
Technology-intensive products are typically designed to perform many functions, only some of which a given customer might evaluate. Just as different customers may make different product comparisons, they may also consider different functions when justifying a technology’s purchase. To continue with the automobile example, consider a different function such as touring. In this case, one presumably would use the car for vacations covering long distances instead of short trips for commuting purposes. The cost/benefit analysis would be different than the analysis of the commuting car, focusing more on comfort, fuel efficiency, and storage capacity. This different analysis is significant because it generates different requirements along with these different uses. A person using an automobile for touring purposes would be interested in features such as cruise control, trunk size, and lumbar support; whereas, a person using the automobile for commuting would not necessarily be interested in these features.

It is important to recognize that complete understanding or justification of a technology is not necessary for customers to purchase a technology. Customers may consume for social reasons as opposed to utility (although one could make the argument that status is a form of justification) (Veblen, 1899). For instance, some insurance firms felt compelled to purchase a computer when it was first introduced even though they did not fully grasp what it was. Instead, they believed that buying a computer would impress their customers because it represented technological advancement (Yates, 2005). At other times, customers may simply appeal to the generally accepted understanding of the technology without careful consideration of their own specific situation. Once they deploy the technology, however, they may realize that their particular circumstances do not
conform to the generally accepted view. Such mismatches between blind initial understanding and reality in use can destabilize a technology leading to abandonment (Zbaracki, 1998). Therefore, although understanding and justification may not be required to make a purchase, the actual deployment of the technology may generate new understandings and ways to justify the technology (Orlikowski, 2000).

As way of summary, Figure 2.1 identifies the three main elements of customer use and how they can be used to explain variation between customer segments. Understanding and justification are essential parts of the investment decision process. Understanding a technology entails comparing it to other products categories. To the extent that customers make different product comparisons, they may understand and use the technology differently. Where understanding focuses on what the technology is justification identifies what it could do. Justification involves calculating the cost/benefit of using the technology to perform a certain set of functions. To the extent that customers use different cost/benefit logics, they may use the technology differently. After making a purchase, customers implement the technology. Technological deployment involves applying the technology to perform a certain set of functions in a certain way. Differences in the customer’s organizational context can lead to different customer deployments. Therefore, customer segments can differ along any one of these dimensions – in how they understand and justify the technology as well as in how they deploy the technology.

Thus far, I have presented a fairly static view of customer use, focusing on the salient factors that help determine how customers use a technology and why their use

---

3 Implementation, of course, may not occur. During the 1990s, purchasing but never using business application software was well documented (Lanzolla & Suarez, 2006). Such software was called “shelfware”. The implications of non-use are interesting, but currently out of scope of this analysis.
may differ. However, another important feature of use is that it is a dynamic act that is ongoing. Using a technology has important feedback properties that may affect future customers’ understanding and justification of the technology. Considering the act of using a technology is the focus of the next section.

2.2 Learning by using

Standard economic explanations of consumption generally assume that customers have relatively static preferences. This assumption does not mean that customer preferences cannot change, but economists tend to treat such changes as exogenous to their
models (see (Teubal, 1979) for an exception). Many scholars in the technology management literature share this assumption. We have already seen how Christensen’s notion of a disruptive technology implies that customers within a segment do not substantially change their use and preferences. In his model, customer change is equivalent to compositional changes within the overall market through the entry/exit of new customer segments (Christensen, 1997; Christensen & Rosenbloom, 1995). Building off of Christensen’s model, Adner and Levinthal (2001) describe evolution in customer demand as a function of the technology satisfying the customers’ functional thresholds. This approach assumes that these thresholds remain static and are different across customer segments. Once functionally satisfied, customers switch from valuing different product features to price. Sutton’s (1998) explanation of the relationship between technology and market structure factors in the potential substitutability of related technologies and the customers’ demand for a technology. But, it also assumes a relatively static demand curve over time.

However, there is good reason to believe that customers can and often do endogenously change their preferences and needs, especially for technologies. Rosenberg (1983) identified learning by using as a principle mechanism for this change. He asserts that using a technology is a form of experiential learning that is also a feedback mechanism: what customers learn could change future understandings and evaluations of the technology. Certain characteristics about technologies in general make using a technology particularly conducive to learning something new. Many technologies are complex – both in their inherent technical design and with the requirements to get it to
work. Getting a technology to work typically requires integration with other technologies and processes that interact with the technology. Given this inherent complexity, it is difficult to predict the technology’s performance with certainty prior to deployment (Von Hippel & Tyre, 1995). When customers initially put the technology into use, they may discover unpredicted technical issues. For example, early users of the automobile learned that it lacked power and waterproofing to perform well traveling through hilly and wet terrain (Franz, 2005). Rosenberg (1983) calls such knowledge embodied as it leads to subsequent improvements in the technology.

In addition, many technologies have prolonged use. Continued experience with the technology can lead to new practices and uses of the technology. For example, Yates observed that initially insurance firms used tabulating equipment for computational purposes, specifically mortality studies (Yates, 1993). Over time, they realized that these same machines could be used to process information requests such as managing customer bills or records. Rosenberg (1983) calls such knowledge disembodied as it leads to changes in use. He argues that such learning is important because it can extend the life a technology as well as lower operating costs. In addition to these benefits for the technology, the new uses can lead to changes in customer requirements and understandings. When the insurance firms expanded their use of tabulating machines to include record management, they created new needs like alphanumeric keys and printing capabilities. As a result, in using a technology, customers may learn new things that generate new interpretations and needs of the technology. These changes may affect future purchase decisions.
This view of learning by using as a generative mechanism is consistent with sociologically-based theories of technology. Orlikowski (2000) argues that technologies do not come with embedded structures and meanings which users must discover. Rather, the process of using the technology, or in her words putting the “technology into practice”, generates the technologies structure and meaning. According to this view, the act of using the technology is an important part of identifying requirements and needs for a technology. Over time, as these uses change so do the structures and meaning.

However, not every change in use leads to a change in customer understanding and needs. And, learning certainly is not the only mechanism that generates changes in use. Understanding the conditions under which changes in use occur, therefore, is important. It is possible that changes in use only happen when the technology changes. In many cases, changes in use occur after prolonged experience with the technology, possibly representing a delayed reaction to technological changes. In these cases, changes in how customers use a technology may simply be an extended consequence of technological change. There is some precedent for this view. Utterback (1994) observed that significant new uses of the DC-3 airplane emerged only after its technical design standardized. The setting of the dominant design may be an occasion to expand use. Similarly, as Christensen notes, new uses tend to emerge during periods of discontinuous change in the technology.

However, there are cases in which a change in customer use precipitates technological change. In her study of the typesetter industry, Tripsas (2006) observed that the early primary customers, newspapers, first used this technology to print text, but over time they began to print pictures as well. This change in use led to significant change from
hot metal typesetting to analog phototypesetting. Moreover, Von Hippel’s (1986) notion of a lead user suggests that certain users have incentive to make modifications to a technology themselves because it does not adequately solve the problem at hand.\(^4\) In these cases, technological change did not induce changes in how customers used the technology.

As discussed in the previous section, factors local to the organizational context into which the technology is to be deployed also shape how customers use a technology. These organizational factors can also lead to changes in use even when there is no corresponding technological change. Insurance firms want to extend the use of tabulating machines into other parts of the organization given its usefulness in actuarial studies. Newspapers expand the use of typesetters because of changing competitive conditions in their business (see (Tripsas, 2006) for additional sources of changes beyond the technological).

As a result, technological change is neither a necessary nor a sufficient cause for changes in customer use. Customers can change their use of technology without a corresponding technological change as in the case of typesetters and tabulating machines (the technical changes occurred after the change in use). Conversely, customers need not change their use of a technology even though the technology changes significantly. We will see examples of this in Chapters 3 and 4. For instance, insurance firms initially used a radically new technology, the computer, just as they did tabulating machines. They simply converted well automated tabulating applications to the computer. Therefore, technological changes alone do not determine changes in customer use. Other sources of change include organizational factors, social actors such as market intermediaries, and

\(^4\) Technically, von Hippel considers lead users during the production of a new technology and not during commercialization, but there is no reason to suppose that lead users do not exist subsequent to the introduction of a technology into a market.
learning from experiences with the technology. Based on these additional factors, the process of how customers use a technology should be decoupled from the process of technological change.

2.3 Dominant use

Technology management scholars have observed an evolutionary pattern of technological change in various industries. Technical variety within an industry gives way to a dominant design (Utterback & Abernathy, 1975), which eventually could be replaced by the introduction of new variety based upon a new technology (Tushman & Anderson, 1986). For example, several variations of the automobile engine existed before the combustible engine emerged as the dominant design (Abernathy, 1978). It is believed that the setting of the dominant design corresponds with a shift in an industry’s competitive dynamics from product to process innovation (Klepper, 1996; Tushman & Anderson, 1986; Utterback & Abernathy, 1975). Consequently, we should ask similar questions about the patterns of customer use. Does how customers use a technology follow a pattern? Is there a convergence to a particular use, a dominant use, just as there may be a convergence in technical design?

Anecdotally, there appears to be several different patterns of customer use. Sometimes technologies enter the market with a specific use, but over-time these uses multiply as customers learn more about the technology. For example, Botox was

---

5 Recognize that the causal mechanisms in these models differ. Utterback’s model asserts that setting the dominant design causes the shift in competition; whereas, Klepper argues the reverse – competition switches to focus on process innovation which reduces product variation.
originally approved to be used for a specific use, treating eye disorders, but during clinical trials it was found to have uses in dermatology. Aspirin has followed a similar pattern. Its uses increase as the government approves the drug to be used for new applications such as preventing heart disease or strokes. Similarly, baking soda originally was used for baking and cleaning, but today it has expanded to deodorizing and cleaning teeth. Other technologies, however, appear to follow more of an evolutionary path, where initial variety gives way to a more specified dominant use. Clark (1985) and Teubal (1979) assert that this is a common pattern because the initial uncertainty during market introduction generates variety in use, but over time customers learn what the technology does well and a more specified use emerges.

But, what causes this convergence? It is important to recognize that learning associated with using a technology implies a non-convergence in use. Learning by using is a form of experiential learning. Since each customer has different experiences due to adapting the technology to local conditions, customers will not learn the same kinds of things. As a result, convergence to a dominant use is not automatic. Yet, it does occur.

There are several ways a dominant use may occur. One is structural: even though customers may be using the technology slightly differently, these differences are not that significant such that it essentially is the same use. In Adner’s (2002) terminology, the structure of the market may be that the difference uses “overlap” and are “symmetrical”. For instance, the farmer’s interpretation of the telephone for social purposes may overlap enough with the cosmopolitan interpretation of the phone for business purposes that the different groups essentially use the phone in the same way. A dominant use emerges. A
dominant use may also form through social processes. Even if there are multiple product comparisons or uses, customers may come to a collective understanding that focuses on one. As we will see in the next chapter, insurance companies initially categorized the computer in several ways, comparing it to the human brain and to tabulating machines. But, through the influence of industry associations and early uses of the computers, insurance firms focused on tabulating machine comparisons. Social constructivists describe this process as the “closure of the interpretive flexibility” of a technology (Pinch & Bijker, 1987). This process is inherently social in which different participants negotiate a collective understanding.

There may be other mechanisms for convergence. That others exist beyond structural and social ones is not important here. What is important is that one must exist; dominant use is not automatic. Nor is a dominant use automatically retained. Because learning from on-going use continues there is the possibility that new uses can be learned which replace the dominant use or customers learn that the actual deployment does not match the perceived benefit of the dominant use (Zbaracki, 1998).

To summarize this chapter, how customers use a technology is interesting because it sits at the intersection of technological development and market demand. Producing firms design technologies to be used in certain ways, but in order to get a technology to actually work, customers must adapt it to local conditions. At the same time, use is on-going. This dynamic characteristic of use influences both technological change and market perceptions, often in unpredictable ways.
Chapter Three

Creating a category: The introduction of the computer into the insurance industry, 1947 - 1960 (with JoAnne Yates)

Having clarified the concept of customer use, we can now turn to more concrete empirical issues and puzzles associated with how customers understand and deploy a technology. In many ways the most significant issue is how customers come to understand and initially use a never before seen or experienced technology. Just as there is uncertainty in the production process, customers often face tremendous uncertainty about what the technology is and what it can and cannot do (Clark, 1985). Given this uncertainty, how these interpretations and uses get formed is not immediately obvious. In this chapter, we explore these issues by analyzing the historical development of how one set of customers, insurance firms, came to understand and use a new to the world technology, the computer. This chapter was done in collaboration with JoAnne Yates, whose historical account of the insurance industry’s use of computers forms the basis of the empirical analysis.

3.1 Introduction

A central belief in organizational and market theories is that actors interpret social objects through schemas, categories, or role structures. Schemas are “knowledge
structures that represent objects or events and provide default assumptions about their characteristics, relationships, and entailments under conditions of incomplete information” (DiMaggio, 1997, p. 269). These structures play an important role in conferring cognitive legitimacy – when a candidate (producer) proposal fits into the audience’s (customer’s) schema it is viewed as appropriate (Zuckerman, 1999). Organizations adhere to these institutionalized categorizations and beliefs in order to improve performance (DiMaggio & Powell, 1983; Hannan & Freeman, 1977; Meyer & Rowan, 1977). In this way, the audience members’ knowledge structures both privilege and constrain organizational behavior, thereby influencing a market’s competitive dynamics. Typically, the effects of categorical schemas have been studied during periods of equilibrium – when an audience’s categorical system is well defined, stable, and taken for granted (Hsu, 2006; Phillips & Zuckerman, 2001; Zuckerman, 1999). Recently, however, there has been increased interest in disequilibrium conditions, in particular, when entrepreneurs try to create a new category by offering innovative products.

In these cases, the entrepreneur does not want his offering to fit into an existing category per se. Instead, she tries to get the audience members to incorporate a new category within the existing system. Often entrepreneurial activities fail because they are unable to achieve legitimacy (Carroll & Hannan, 2000; Stinchombe, 1965). Scholars often frame this issue from the perspective of the candidate. The candidate, in this case the entrepreneur, must propose a new category that is novel enough to be distinctive from other categories, but it cannot be too novel because potential customers may not be able to recognize what it is and ignore it (Hargadon & Douglas, 2001).
This framing of the issue implies that during periods of innovation, an audience’s categorical schema constrains the behavior of the candidate in similar ways during periods of equilibrium. However, the state of the audience’s belief system during disequilibrium fundamentally differs from equilibrium. In equilibrium, the categorical system constrains the behavior of candidates because it has integrity and stability (Zuckerman, 1999). Conversely, the categorical system lacks stability during disequilibrium precisely because a new category is being introduced. A new category can change the relationships between existing categories, and audience members may re-interpret existing categories in light of the new category. Without stability, the categorical system cannot impose the same level of constraint as during equilibrium periods.

As a result, from the audience’s perspective, creating a new category is deeply problematic. In order for the audience’s categorical system to be effective it must have integrity and stability. Yet, the introduction of a new category potentially threatens the very stability of the entire system. Critics have identified the issue of accounting for change in the audience’s institutionalized beliefs while still maintaining its integrity as a significant issue in institutional theories (Barley & Tolbert, 1997; Hirsch & Lounsbury, 1997). In many respects, this issue facing the audience is more fundamental than the analogous issue facing the candidate. Since an audience is ultimately necessary to confer legitimacy to the candidate, it does not matter what obstacles entrepreneurs must overcome if it is not possible to explain how an audience’s categorical schema can change.

Shifting focus to the audience also reveals two significant empirical issues in the introduction of a new product category. First, how does the audience determine to which
existing product categories it compares the new product? Given the unfamiliarity of the new product, it is not obvious what should be the proper product comparison. For example, during the commercial introduction of the telephone, different audience members made different product comparisons: urban customers compared the telephone to the telegraph, but rural customers compared it to traditional social practices like talking on a porch (Fischer, 1992). Second, to compare a new product with an existing category presumably requires some knowledge about the new product. But, given the newness of the category, audience members often lack the capabilities or knowledge to make an effective comparison. Who in the audience can even make the necessary product comparisons?

In this chapter, we begin to address these issues by detailing the history of how one particular audience, insurance firms, came to understand and accept a new product category, the computer in the late 1940s through the 1960s. As such, we examine how an audience incorporates a new technological product into its product categories. For insurance firms, the computer represented a new technology, but it was not immediately obvious what it was and how it should be used. To create a product category for the computer, insurance firms made multiple comparisons. One was with tabulating machines, an existing technology heavily used at the time; the other was with the human brain. While the former association highlighted potential information processing uses, the latter association emphasized the decision-making and integrative qualities of the computer. In

---

6 This analysis differs from a traditional focus on the emergence of a new organizational field or form. But, organizational fields are conceptually elastic (McKendrick, Jaffee, Carroll, & Khessina, 2003; Romanelli, 1991) and consequently it is difficult to identify and measure the actual change that takes place. In contrast, actual products have the advantage of being more concrete, recognizable, and measurable.
aggregate, the combination of the comparisons created enough ambiguity to sustain the computer as a distinct category. Relating the computer to the tabulating system put the computer into a context the broader audience could understand, but relating the computer to the human brain also distinguished the computer from this existing technology.\(^7\)

These comparisons, taken collectively, helped distinguish the computer from any given category. As a result, a certain level of ambiguous identity (Padgett & Ansell, 1993) was necessary to sustain the introduction of the new product category. This robust identity contrasts with the need to fit into a product category during periods of equilibrium. How customers compare in innovative situations differs from equilibrium. With a new product, customers compare by focusing on the similarities and differences with product categories, but in equilibrium they are more strictly looking to determine if it belongs in the product category.

No doubt design choices by the computer manufacturers influenced which product comparisons insurance firms made. The physical appearance of IBM’s 650 system, the most successful computer at the time, closely resembled contemporary tabulating systems, to the point that the old tabulator at the center of the system could simply be wheeled out and the 650 processor wheeled into its place. In addition to design elements, market intermediators, in particular industry associations, played an important role in making these product comparisons explicit and focusing the broader audience on the comparison with tabulating machines. Industry associations, such as the Society of Actuaries, LOMA,

\(^7\) By itself, the image of computer as human brain may have created concern that it could potentially replace the very managers who were in charge of justifying its purchase. Establishing a relationship with tabulating systems (or desktop calculators) and identifying uses associated with clerical work would have helped dampen these concerns.
and IASA, had incentives to examine and present information about new business trends in order to reinforce their relevance to their membership. In doing so, these associations leveraged experts who may be more qualified in making categorical comparisons. In this case, the Society of Actuaries, like the other two industry associations, commissioned a report explaining what the computer was and what its potential uses should be, even before the computer was commercially released. These associations played the role of translator, making connections between old and new technologies that any given member may not be able to make. Acting in this role, industry associations helped develop a collective understanding of what a new category is.

Lastly, insurance firms were not just evaluators, but also users. As soon-to-be users, they focused on what function the technology should perform and its potential benefit. Calculating the potential pay-off—a step necessary for justification of a significant purchase or rental—did not necessarily require insurance firms to be able to categorize the product cognitively because this calculation focused on what the product can do as opposed to what it is. Even though many insurance firms did not know what the computer was, they could still think about what to do with it and the potential benefit it brought to the organization (Fisher, McKie, & Mancke, 1983). However, the current uses of existing technology, like the current understanding of it, constrained their benefit analysis. When insurers began to consider what the computer could be used for, they tended to look at how they currently leveraged technology to perform their business functions—tabulating machines to automate administrative tasks and calculating machines to perform computational tasks.
The actual use of the technology also played an important role in sustaining the categorization of the new technology, suggesting that institutional accounts may overemphasize the cognitive and normative aspects of the audience’s belief system. Ultimately for an understanding of the technology to persist, the technical reality of its implementation must support this socially constructed interpretation (Strang & Soule, 1998; Westphal, Gulati, & Shortell, 1997; Zbaracki, 1998). Actual use may reveal characteristics of the new product category that initial categorical comparisons made it difficult to see. As insurance firms began to implement computers, they realized that they underestimated the cost of programming computers—a feature not relevant to tabulating machines. This underestimated cost reduced the potential benefit of the incremental applications they deployed on the new machines. In addition, on-going use served as a mechanism to refine a new product’s categorization, and companies gradually added new applications and began to integrate them during the 1960s. Moreover, the fact that a few insurance firms developed more radical uses of the computer provided examples of using the computer in different ways.

3.2 How insurance firms created a new product category for the computer, 1940’s – 1960s

We analyze the introduction of the computer into the insurance market, incorporating data from both the producer and customer perspectives of the technological change from tabulators to computers. Historical case studies provide some advantages
over cross sectional or even quantitatively focused longitudinal research. Historical analysis is particularly relevant since we are primarily interested in explaining the process of how this new understanding and use developed.

The computer industry has received a fair amount of attention from the producer perspective (Bashe, Johnson, Palmer, & Pugh, 1986; Campbell-Kelly & Aspray, 1996; Fisher et al., 1983; Norberg, 2005), and the discussion of that perspective draws primarily on that secondary literature, supplemented with published materials of the era discussed as well as occasional unpublished materials from the computer archives at the Charles Babbage Institute (CBI) of the University of Minnesota and from the Unisys Collection in the Archives of the Hagley Museum and Library (Hagley). We base the customer perspective on data from several primary sources, both published and unpublished.8 Research in the archives of three insurance firms (Metropolitan Life, New England Mutual Life, and Aetna) provides extensive data on some specific acquisition and use decisions in these firms. The Edmund C. Berkeley papers at the CBI document Berkeley’s early interactions with the computer industry on behalf of Prudential. The CBI also contains transcripts of oral history interviews with early computer representatives, some of which shed light on insurance users. In addition, one of the authors interviewed several individuals involved in this transition in specific insurance or computing firms.

We also leverage a previously untapped source of data around insurance adoption and use of computers—the printed proceedings of annual meetings and special conferences of the insurance associations (including the Society of Actuaries, the Life Office Management Association (LOMA), and the Insurance Accounting and Statistical

---

8 This material was originally gathered for use in a historical monograph by JoAnne Yates.
Association (IASA)). These detailed contemporaneous accounts describe many insurance firms’ thinking and decisions about what computer to adopt and how to use it. Often, they also contain verbatim transcriptions of the question-and-answer period that followed most sessions, revealing further information about customer interpretations and applications. For some firms, frequent conference papers document ongoing use and results. Moreover, committee reports and special publications document activities of insurance association committees investigating possible uses of computers in life insurance. Last, the insurance associations themselves are of theoretical interest in their role of disseminating these different comparisons to individual insurance firms.

We organize the presentation of the historical case as follows. We establish why the computer represented a new product category based upon technological differences with electromechanical tabulating equipment to develop the computer. Because the understanding and use tabulating equipment played such a prominent role in the insurance firm’s understanding of the computer, we briefly review how they used this equipment prior to the introduction of the computer. We then identify how various understandings of the computer emerged, and how the incremental understanding of the computer as a tabulating machine came to dominate the market and shape insurance decisions around adoption and use.
3.2.1 What makes the computer a new product category

To highlight the newness of the computer, we need to introduce tabulating technology, previously used to manipulate large amounts of data. Tabulating technology consisted of various related (but physically separate) devices to punch holes in cards, then sort, count, or perform calculations on the value encoded in some column. In its pre-World War II form, tabulating technology stored data on punch cards created using a card-punching device, manipulated the cards electromechanically to perform one sorting, counting, or calculating operation per pass through a device (an array of specialized devices existed by this time), and recorded the outcome, where appropriate, on a printing device (e.g., a list of sorted data or a total from a calculation). Operators configured the sorting and calculating devices in advance of each operation, using a plugboard and wires; to proceed from one operation to another on the same device, they could replace the pre-configured plugboard with another one or rewire it. Operators had to move cards manually between operations.

Two companies supplied tabulating equipment in the US during the second quarter of the twentieth century: IBM, which focused primarily on tabulating systems and only secondarily on a few other office devices, and Remington Rand, a large manufacturer of many office devices (e.g., indexing systems, adding machines, typewriters), with tabulating equipment accounting for a relatively small portion of its revenues. Although Remington Rand was the larger company, IBM dominated Remington Rand in the tabulating market by around 83% to 17% in 1950 (Norberg, 2005). The two competitors
had different business models, with IBM renting equipment and selling cards, while Remington Rand sold both equipment and cards.

During and immediately after World War II, a series of technological developments in what we now call computers offered radically new ways of manipulating data. Based on the existing economic, technical, and organizational criteria previously outlined, the transition from tabulating machines to computers represented a radical technological change for the incumbent producers. Tabulating systems and computers differed in at least three areas:

- **form of data storage**: early computers stored data for manipulation in magnetic form in variable length records on tape or drums rather than on fixed size (typically 80-column) punched cards;
- **method of data manipulation**: early computers used electronics—vacuum tubes, relays, and circuits—rather than electromechanical mechanisms to manipulate the data; and
- **form of storage for procedures**: computers could store and call programs, or sequences of steps, electronically within the system, while a tabulating machine had to have its circuits physically reconfigured on a plugboard for each process step.

Indeed, the capabilities required to design and manufacture computers—including recording and reading data magnetically, designing reliable vacuum tubes and creating
circuits to connect them, and programming sequences of operations—differed significantly from those required to produce tabulators or other electromechanical business machines.

Underscoring the radical nature of the shift for producers of tabulating systems, in 1950 one of the two manufacturers of tabulating equipment, Remington Rand, chose to buy rather than internally develop the new technology, acquiring Eckert-Mauchly Computer Corporation (EMCC) and its UNIVAC, the most advanced computer being developed for commercial use. In contrast, IBM chose to develop the computer internally. Many were skeptical—a key player in at least one potential user firm, Prudential Life Insurance company’s Edmund C. Berkeley, reported in 1946 that “it seems clear that International Business Machines, Endicott, N.Y., is not to be included in this list [of potential vendors] at the present time, because of their paramount investment and interest in punch card accounting machines, and the great backlog of demand for such machines.”

In addition to the tabulating incumbents, the new technology attracted other entrants, including manufacturers of other office machines (Burroughs and NCR), start-ups (Control Data), and large firms from other sectors (GE, Honeywell, Raytheon, and RCA). Some of the last group had expertise in related technical areas (e.g., RCA with vacuum tubes and magnetic tape), but these players lacked capabilities in programming, as well as any knowledge of the commercial market for such business machines. Burroughs and NCR, like Remington Rand and IBM, had customer bases and knowledge of customer processes in certain areas, but their accounting machines and cash registers were separate

special-purpose devices, and thus farther away from the new general-purpose computer systems than were tabulating systems.

3.2.2 The insurance firms’ use of tabulating equipment

The use of tabulating equipment influenced how the insurance companies developed their interpretation of computers. As an information-intensive business, life insurance had become an early and important user of tabulating technology. By the 1940s, virtually all insurance firms used tabulating equipment, most often rented from IBM but sometimes bought from Remington Rand. Indeed, IBM historians have noted that “Insurance companies were among the largest and most sophisticated business users of punched-card machines in the late 1940s and 1950s” (Bashe et al., 1986). Life insurance firms had generally divided and specialized insurance operations as they grew, and they used tabulating equipment in line with this approach. A typical insurance application performed a single, narrowly defined operation. By the late 1940s, however, some smaller firms had started to use tabulating technology to integrate premium billing and accounting operations, thus reducing data repetition (Cassara, 1952).

During World War II many insurance actuaries worked on large tabulating installations and a few on the new computing technology being developed for military purposes (Moorhead, 1989). Moreover, as the wartime clerical labor shortage was exacerbated by a post-war life insurance boom, insurance firms were seeking technology to
automate more of their processes. From 1948 to 1953, the dollar value of life insurance in force grew 51% and the total number of insurance policies (a better indicator of information processing volume) rose by more than 24% (Bureau of Labor Statistics, 1955). Growth in insurance employment was 12% during the same period, not enough to keep up with the increased data processing load. Thus insurance firms were interested in any technology that promised to restrain its growing clerical labor needs. The post-war period was marked by ongoing interaction between representatives of insurance firms and the groups attempting to develop and produce the new computing technology. These interactions shaped what became the dominant insurance interpretation of the technology as an incremental advance over tabulating technology.

3.2.3 Pre-adoption categorization – the computer as tabulator and brain

Insurers got involved with computers at both the individual and trade association level in the late 1940s. Through this involvement, at least two different understandings of the computer—as tabulator and human brain—developed and coexisted in the insurance community during the late 1940s and early 1950s, though the balance tipped towards the tabulator analogy.

Prudential’s Berkeley was directly involved with the development of the UNIVAC for commercial (as well as governmental) purposes. Berkeley’s views of the computer were revealed in his negotiations with EMCC, in his presentations to various insurance
associations, and in his popular book on computers, *Giant Brains, or Machines That Think* (Berkeley, 1949), published after he left Prudential in 1948. As early as 1947 he presented a paper to the Life Office Management Association (LOMA) meeting in which he referred to the new technology as “Giant Brains” or “mechanical brains that could do calculations that “it would take 10 good mathematicians 6 months to do” (Berkeley, 1947a, p. 116). In the same talk, however, he called the technology “electronic sequence controlled calculating machinery” and said it was “much like a desk calculating machine” (Berkeley, 1947a, p. 116), a machine that a mathematician or clerk could use to perform complex sequences of computations. In the same year, Berkeley presented another paper at the IASA annual meeting (Berkeley, 1947b). In this paper, in discussing potential uses he highlighted a premium billing application, clearly derived from current tabulator applications, suggesting an analogy to a tabulating system.

In his correspondence and negotiation with the designers of the UNIVAC on behalf of his company, he acted on this last understanding of the computer as tabulator. He imagined that Prudential would use a computer system like it used its tabulating installation: for processing large numbers of transactions with a great deal of input, relatively little computation, and extensive output of transactional documents such as card records, and premium bills (Yates, 1997). Thus, he insisted that the computer system include card-to-tape and tape-to-card readers and high speed printers to facilitate incremental, tabulator-like use of computers. His working understanding of the technology, then, was of it as a tabulator—though one that could do multiple steps without human intervention.
Special committees of several insurance associations also actively shaped insurance executives' interpretations of the new technology. The "Committee on New Recording Means and Computing Devices," established in 1947 by the Society of Actuaries, was the committee with the greatest influence. As its influential 1952 report would explain, "life insurance people and electronic engineers were two groups who did not speak each other's language. [...] It became apparent that some medium was necessary to bridge the gap between the two" (Davis, Barber, Finelli, & Klem, 1952). The Society of Actuaries formed its Committee—composed of actuaries from the 1st (Metropolitan Life), 4th (Equitable Life Assurance), and 14th (Connecticut Mutual Life Insurance) largest insurance firms (Best, 1948)—to bridge this gap for the industry, rather than leaving it to each company to bridge the gap individually (Yates, 2005). Its mandate was to study how insurance firms might beneficially use the emerging electronic computing equipment.

The Committee served at least three important functions for the insurance industry. First, it translated and bridged the gap between electrical engineers and insurance people, shaping how most insurance firms would understand computer technology for years to come. The Committee explained it primarily by analogy to punched-card tabulating. The report compared the magnetic spots on tape to punched holes on cards and explained programming as similar to configuring tabulator plugboards. It thus interpreted computing technology not as radically new, but as an incremental extension of the familiar tabulating technology already used by almost all life insurance firms. This view would strongly influence how insurance executives viewed and used the technology.
Secondly, the Committee represented a major potential market segment, insurance firms, to the potential suppliers, conveying to them how firms in this market were likely to view the new technology, and for what purposes they were likely to adopt and use computers. These interactions highlighted the importance of leveraging existing attributes and functions of the tabulating machines into the design of the new computers. For example, the Committee made clear that the life insurance industry, which retained records for as long as a person’s lifetime and operated under regulatory constraints, was not willing to give up their (visible) punched-card records for (invisible) magnetic records. As a consequence, suppliers learned that to market to this segment, they must develop card-to-tape and tape-to-card conversion equipment to make it possible for insurance firms to keep records in punched card form, maintaining some continuity with their existing technology and processes. In addition, the Committee’s testing of possible applications on experimental machines educated the developers about possible insurance applications, giving potential suppliers insight into how to sell to this market.

Finally, the Committee arrived at and promoted a vision of how insurance firms should use the technology as soon as it was commercially available. The Committee considered and rejected computation-intensive actuarial uses such as mortality studies—incremental uses based on using the computer as a powerful calculator—as not extensive enough to allow a computer to pay for itself in any but the very largest firms. Next, the committee considered applying computers to record storage and retrieval to reduce the space taken up by punched cards. They rejected this application, too, since magnetic tape could only be searched sequentially, not randomly, at this time, and magnetic records were
not yet accepted by courts and regulatory bodies as the primary record of a policy.

Ultimately, the Committee concluded that electronic computers could best be used in insurance to perform routine servicing of insurance policies, taking over and integrating several operations (e.g., calculating premiums and dividends, printing premium bills, recording dividends, and performing accounting operations) now handled by tabulators.

The so-called Consolidated Functions approach, presented in the 1952 report, was based on the integrated premium billing and accounting application for which some firms used tabulating technology by the early 1950s. For smaller and more innovative insurance firms already using the tabulator application, it represented an incremental extension that integrated a few more operations, rather than a radical transformation. For those firms still using traditional subdivided operations, however, this application represented a somewhat more radical change in insurance processes. The Committee demonstrated the application’s workability using IBM’s Card-Programmed Calculator (CPC), a transitional system that was not a real stored-program computer like a UNIVAC, and used punched cards instead of magnetic tape, underscoring the more incremental aspect of the application. The Committee’s vision of computer use as embodied in its Consolidated Functions approach, which had both incremental and radical aspects, would influence insurance use for the next two decades—indeed, it would take that long for most firms to achieve the application.

The 1952 report taken as a whole showed some tension between an incremental, tabulator-like approach to using computers and a more radical transformation in insurance processes. The Committee members from Metropolitan Life, John J. Finelli and Malvin
Davis, recognized the potential offered by a more radical use of computers; a paper written by Davis and delivered by Finelli a year later stated that "a basic reengineering of present procedures" would be necessary in insurance before firms could take full advantage of computers (Davis, 1953). In his section of the report, Davis stated several principles for using computers, including integration rather than subdivision of jobs (Davis et al. 1952, p. 22). The Consolidated Functions approach was incremental for some advanced tabulator users but modestly more radical for traditional users. The report’s final summary, however, stressed an incremental, rather than radical, approach to this change—at least in the immediate future:

Realization of their full potential will lead us to the consolidation of many different kinds of work which now are departmentally separated. This, however, is not necessarily a matter of sudden, drastic reorganization. The program indicated is more likely to be the introduction of an electronic computer for one related series of procedures, and the gradual step-by-step adaptation of other routines as we learn at first hand the machine’s capabilities and limitations. (Davis et al., p. 49)

Multiple understandings of the computer—comparisons to tabulators and brains—coexisted in the early interactions between insurance and computing. The interpretation of the computer as a brain implicitly raised the possibility of an entirely new way of doing insurance work, a radical technological change from the perspective of the user. The more common interpretation of computers as similar to tabulators (or to calculators), on the other hand, made the technology seem more familiar and supported a more incremental approach to change.
3.2.4 Early insurance adoption and use of computers

As the first UNIVACs became commercially available in 1954 (several of them had gone to government buyers starting in 1951), insurance firms had to translate their interpretations of computers into rentals or purchases of equipment and into insurance applications. In both hardware and application choice, a large majority of insurance firms chose a more incremental than radical option.

Both of the tabulating incumbents, IBM and Remington Rand, offered computers that were radically different from tabulators as well as devices that were only incrementally different. Remington Rand, through its acquisition of EMCC and its UNIVAC, had taken an early lead in computer technology. The UNIVAC computer system filled a room, with its large central processor, many Uniservo tape drives, and tape-to-card and card-to-tape converters. The room had to be specially prepared, with reinforced floors to support the computer’s weight and a powerful air conditioning system to reduce heat emitted from the many electronic tubes. When Metropolitan Life installed its UNIVAC, the components of the computer were so large that the company had to use a crane to raise them up the outside of its building and swing them in through a window. This computer looked radically different from tabulators, and the glassed-in room Metropolitan Life created for it drew internal and external gawkers. Moreover, its selling price was around $1.25 million, a substantial capital investment. IBM was working to catch up with the UNIVAC technically, developing the large, tape-based 701 (available in 1953) for scientific and defense use and the 702 (available in 1955, but replaced by the
more reliable 705 in 1956) for commercial use. Like the UNIVAC, IBM’s 700-series computers included converters to allow users to use cards or tape as an input/output and storage medium.

Both firms also developed more incremental devices with designs that shared fundamental features of both tabulating and computer technology. IBM intended its 650 model, announced in 1954 and released in 1955, as an interim, small computer to keep its installed base of tabulator users from moving to other computer vendors as it completed its development of bigger, magnetic-tape-based computers (Ceruzzi, 1998). The 650 shared some essential features of the larger, 700 series computers, such as stored program capability (though during operation the program was stored in a magnetic drum rather than on magnetic tape), and some of tabulating machines, such as using cards rather than magnetic tape as its primary long-term storage medium. Table 3.1, adapted from a 1954 IBM presentation at an insurance association meeting, arrays the 700-series computers and the 600-series tabulating equipment on a continuum, placing the 650 in between but slightly closer to the computers than the tabulators. The IBM 650 closely resembled tabulating machines visually, as well. It was housed in a cabinet of the same design and appearance as that of an IBM 604 electronic tabulator (a tabulator using vacuum tubes, introduced in 1948), and it could be rolled into an existing tabulator installation to fit where the 604 had been (Yates, 2005). To insurance firms, it clearly presented a stark contrast with the enormous and alien-looking UNIVAC and IBM’s 700-series computers. The 650 offered an incremental advance on the familiar tabulators, allowing adopters to
Table 3.1 Comparative features of IBM's electronic data-processing equipment

<table>
<thead>
<tr>
<th></th>
<th>604</th>
<th>607</th>
<th>CPC</th>
<th>650</th>
<th>701</th>
<th>702</th>
</tr>
</thead>
<tbody>
<tr>
<td>Announcement Date</td>
<td>1948</td>
<td>1953</td>
<td>1949</td>
<td>1953</td>
<td>1953</td>
<td>1953</td>
</tr>
<tr>
<td>Storage (# of decimal Digits)</td>
<td>50</td>
<td>66 - 162</td>
<td>290 - 930</td>
<td>10,000 – 20,000</td>
<td>20,000 – 8,000,000</td>
<td>2,000,000 – 575,000,000</td>
</tr>
<tr>
<td>Card input/Output</td>
<td>100/100</td>
<td>100/50</td>
<td>100-150/30</td>
<td>200/100</td>
<td>150/100</td>
<td>250/100</td>
</tr>
<tr>
<td>Magnetic Tape</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Page Printer</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Program Control</td>
<td>Wired</td>
<td>Wired</td>
<td>Wired/Stored</td>
<td>Stored</td>
<td>Stored</td>
<td>Stored</td>
</tr>
<tr>
<td>Monthly Rental</td>
<td>$550 Up</td>
<td>$800 Up</td>
<td>$1,175 Up</td>
<td>$3,250 – 3,750</td>
<td>$11,900 – 15,000</td>
<td>$20,000 Up</td>
</tr>
</tbody>
</table>


keep their current tabulating installation room layout, cards, rental model, and normal vendor.

Remington Rand also recognized the need for an incremental hardware solution for customers not yet ready for the large UNIVAC, and it released two smaller, punched-card machines (the UNIVAC 60 and 120) in the early to mid 1950s. Despite the UNIVAC name, however, these machines were developed internally based on Remington Rand’s 409 tabulator and lacked any internal storage capability; thus they were far closer to tabulators than to computers. Although these machines were reasonably successful in the tabulator market, they did not really compete with the IBM 650. Moreover, Remington Rand had difficulties absorbing EMCC and another small computer start-up and devoted inadequate resources to entering the computer market. The resulting delays in the availability and lapses in the marketing of these machines allowed IBM to pull ahead in the market (Norberg, 2005). Both incumbent firms attempted to offer an incremental solution
that closely resembled tabulating machines in design, dimension, and cost but that allowed users to move towards computer technology, as well as a more radical solution, the large, tape-based computer. IBM was far more successful in filling this user-perceived need.

Several other early entrants into the computer market offered small and inexpensive drum-based computers during the early and mid-1950s, but they were slow, harder to program, less reliable, and often required IBM punched-card peripherals for customers desiring card input and output (thus diverting much of the revenue to IBM) (Ceruzzi, 1998). Moreover, the companies producing them (e.g., Consolidated Engineering and Computer Research Corporation) lacked both knowledge about pre-existing uses of tabulating machines and an installed base of customers. They generally were bought up by larger firms attempting to acquire the capability to produce computers. Since these larger firms were not in the tabulator market and had not worked closely with insurance firms, they also lacked knowledge of pre-existing insurance uses of tabulating technology, awareness of their interpretations of computers, and established sales and service relationships with insurance firms. Unsurprisingly, they achieved very little presence in the insurance user market.

Consistent with the predominant pre-adoption understanding, insurance firms overwhelmingly followed an incremental path, with most initially acquiring the more tabulator-like 650 after it became available in 1955 and transferring existing tabulator applications onto it. Although the UNIVAC was available a year earlier, in 1954 and 1955 only four insurance firms—Metropolitan Life, Franklin Life Insurance Company, John Hancock Mutual Life, and Pacific Mutual Life Insurance—took delivery of UNIVACs. In
1956, a panel on the new computing technology at the annual IASA conference included 18 papers on IBM 650 applications and only 3 on UNIVAC applications (IASA, 1956). By the following year, life insurance companies had acquired roughly 50 IBM 650s (Carlson, 1957).

Most insurance firms, even those adopting the UNIVAC, also showed a preference for incremental applications when these offered sufficient projected payback to allow them to justify acquiring a computer for $1.25 million. On paper, at least, a single function could provide adequate volume for payback in very largest firms, but in most firms one function was not enough, and multiple consolidated functions were needed to justify purchase (Swinerton, 1956). The four insurance firms adopting the first UNIVACs followed these two general approaches in choosing their applications: the two large UNIVAC adopters, Metropolitan Life and John Hancock, followed an incremental use pattern, initially converting existing tabulator applications with no additional integration. The two medium-sized firms, Franklin Life and Pacific Mutual, followed a more radical use pattern, transforming their insurance processes by integrating many functions.

Metropolitan Life and John Hancock, both of which had struggled with serious clerical labor shortages for a decade, were large enough that each simply converted a single tabulator application onto the UNIVAC, the most incremental of use strategies. As the largest insurance company in the world, Metropolitan Life could project savings from speeding up and eliminating clerks from the actuarial function. Although two of its actuaries were on the Committee that developed the Consolidated Functions plan, the

---

10 Projections of savings from computer use, not actual savings (which were rarely fully assessed or reported afterwards), shaped purchase and application decisions.
firm’s top executives hesitated to start with such a use, given the company’s investment in the existing record system, the organizational disruption that would be required to integrate previously unintegrated operations, and the potential threat to customer relations around the premium billing application. Consequently, “the company, … although inclined to accept the idea that some sort of consolidated operation was indicated as a long-range objective, still felt that a system should be applied to a localized area as a means of getting started” (Davis, 1953). Its executives viewed the actuarial application as less risky than the Consolidated Functions application, and they calculated that this application would allow them to pay back the computer investment in just four years. John Hancock, the fifth largest insurance firm, had already integrated its premium billing and accounting applications on tabulating equipment, and it justified its purchase by transferring this application directly to the UNIVAC. The integration in this application involved only two functions rather than the expanded set recommended in the Consolidated Functions plan. Thus both John Hancock and Metropolitan Life initially chose an incremental use strategy of converting existing tabulator applications.

In contrast, the two medium-sized firms adopting a UNIVAC, Franklin Life and Pacific Mutual, chose to transform their processes by integrating multiple functions as they computerized. When they chose the UNIVAC and the application based on which they would justify it, neither was large enough to simply transfer an existing function from tabulators to the UNIVAC, and both faced unusual circumstances that motivated their desire to acquire the first computer expected to be commercially available. Franklin Life’s sales were growing at five times the industry rate, and the office force needed to service its
policies was also growing at a rapid rate. By 1951 it faced a space crisis that would soon require building a new headquarters office building (Becker, 1954; Vanselow, 1954). One executive argued that adopting a UNIVAC would slow the clerical growth enough to postpone need for a new building for several years. To show that a UNIVAC would pay off for the firm in four years, he proposed a plan to consolidate operations even beyond what the Society of Actuaries Committee would recommend in the following year. Its use was clearly more radical and innovative than those of the two large firms.

Pacific Mutual took an even more radical approach to computer use. Because of a financial crisis, the California State Insurance Commissioner had reorganized it starting in 1936. When the firm decided to order its UNIVAC in the early 1950s, it had recovered from the worse of the crisis, was growing fairly rapidly, and wanted to cut its still high costs by adopting the new technology. It also devised an integrated application, abandoning its cards for magnetic tape. It ran a daily cycle system that processed every policy every day, updating information as necessary and printing all records out on a high-speed printer. This use required an immediate and radical transformation of the company’s processes. As one of its executives explained,

> It is not merely a means of doing with electronic data processing devices the same work that was formerly done on electric and electronic accounting [i.e., tabulating] machines. It is a sweeping new approach to the problem. Its economies come not so much from a faster, higher powered machine as from the single file concept processed daily to meet the relatively random reference needs of service to our policyowners. (Dotts, 1956).

Although UNIVAC purchasers split in their application strategies, IBM 650 users chose incremental applications for their relatively more incremental computers. The 650 was
much smaller than the UNIVAC, so mid-sized firms could use them for one or two single-function applications, just as large companies used the UNIVAC. Moreover, at this time IBM rented rather than sold its computers, as it did its tabulators. Monthly rental for a 650 was just over $3200, comparable to that for tabulating equipment of similar capacity, so it was much easier to justify a 650 than a UNIVAC. As an automation consultant explained with relation to one small firm’s 650 acquisition, “The selection of the IBM 650 was justified on the basis that it would replace IBM punched card equipment, either installed or on order, with an approximately equivalent monthly rental.”11 Consequently, in the early years, 650 applications were generally direct conversions of existing tabulator applications, though firms often saw the integration of the Consolidated Functions approach as a desired endpoint of a very gradual transformation in insurance processes around computers. For example, Equitable Life announced that it intended “to use 650 machines as an intermediate step to a full 705 installation, by programming, testing and setting up operating procedures for dividends, commissions and several related projects, first for the 650 and then translating these programs to [IBM] 705 programs” (DeVries, 1956).

3.3 Conclusion

This chapter explored how a market comes to understand and use a new product category. We argue that understanding a new product category requires understanding

---

how it relates to a variety of existing categories. These collective comparisons help
differentiate the new product category from any one given category. Consequently,
understanding of the new category is related to which comparisons are actually made and
why; this understanding, in turn, plays an important role in the viability of a new category.
The interplay between producers and customers, mediated by third parties, as well as
historical usage patterns of existing technologies, influence which comparisons are made
and emphasized. Market mediators such as industry associations play an important role in
translating this variety so that members can learn about the differences. An implication of
this view is that traditional radical and incremental distinctions may be too categorical and
misleading. At any particular moment, depending on what comparisons are made and
emphasized, the new technology may be understood by the market as both radical and
incremental. This view emphasizes the paradoxical combination of continuity and
difference during technological change. The strategic dilemma for producers and
customers then becomes how to handle the contradictory notions of maintaining continuity
and achieving discontinuity.
Chapter Four

Dominant use: The social evolution of U.S. manufacturing’s understanding and use of planning software, 1954 - 1996

4.1 Introduction

Over time, how customers categorize and use the technology may change significantly from the initial categorization. Use is on-going and can lead to changes. This chapter explores the dynamic aspects of customer use through the historical overview of how U.S. manufacturers understood and used manufacturing planning software from its inception in the early 1950s through the transition to client/server technology in the mid 1990s. Manufacturing software automates the processes associated with the planning and controlling of product and inventory within the manufacturing process. For example, it determines how much inventory is needed to meet projected demand, and it makes changes to product plans as manufacturing conditions change. This software is a particularly good technology to study use patterns over time. In order to function, it often requires organizational change and integration with other technologies. Thus, its performance characteristics are not well known prior to implementation, increasing the opportunity to learn by using the technology.

During the time period of this study, manufacturers changed how they used this software several times based on shifting product comparisons, experiential learning, and

---

12 The study of usage patterns ends in 1996 when the software market began another significant transition to Internet architectures. Although this sacrifices industry data, from a usage perspective there is still uncertainty about how customers use Internet software.
technological changes. Table 4.1 provides a summary of these changes in terms of the characteristics of the conceptual model. The shifts to the “MRP Crusade” and later to “ERP” represent large manufacturers converging to new dominate uses of the software based on these different understandings and justifications. In the 1960s and 1970s, manufacturers shifted their categorization of computers and applications from tabulating machines to the human brain and their justification from eliminating overhead to improving operational efficiency. This shift coincided with a change in how manufacturers used software, as they began to consider applications like MRP that could help improve decision making to reduce inventory. In response to the marketing efforts of the “MRP Crusade”, manufacturers thus began to converge on this new use, establishing a new dominant use. As manufacturers continued to work with the software, they learned new uses for and limitations of the MRP approach, and customized their own usage accordingly. By the time the disruptive shift to client/server architecture arrived in the

| Table 4.1: Changes in large manufacturers’ use of planning software, 1954 - 1996 |
|-----------------------------------------------|-----------------------------------------------|-----------------------------------------------|
| **Investing Justification**                  | **“Practical Approach”**                       | **“MRP Crusade”**                             |
| Displace Clerical workers and tabulating machines | Operational Cost Containment (e.g., reduce inventory) | Competitive Advantage (e.g., improve customer service and quality) |
| Tabulating equipment, human brain            | Computational                                  | Computational                                  |
| **Understanding**                            | **Deployment**                                | **How Deploy**                                |
| **Functions**                                | **Deployment**                                | **How Deploy**                                |
| Department level functions, accounting pervasive | MRP, MRP II – vertical functional integration | Centralized within division, but decentralized across organization |
| Early push for centralization but became more decentralized | ERP – horizontal functional integration | Centralization with some local processing autonomy |


early 1990s, there was variety in how manufacturers used the software, especially between the highly centralized approach in small manufacturers and the decentralized deployments in the large manufacturers. Thus, the change toward reengineered, centralized applications was radical for large manufacturers but not for small manufacturers.

Historical details show that learning by using the software generated many of these shifts in how manufacturers used the software. After an extended period, manufacturers learned that current use practices did not yield the desired return. In some cases, this learning extended the life of the software and in others it helped identify alternative uses. Market intermediaries played an important role in diffusing this learned knowledge and helped shape new uses of the technology. Interestingly, after repeated experiences of poor return on investment with this technology, manufacturers continued to identify new uses rather than abandon using the software.

4.2 Data Sources

Many historical accounts of technological progress and industrial change primarily focus on the technology, innovators, and the firms who comprise the industry. Customers get cursory attention, typically through references to sales and market size. Yet, sales numbers do not capture how customers use the technology. I used the conceptual framework described in Chapter 2 to organize data collection to focus on how the manufacturers understood, justified, and deployed the applications. Manufacturers have
openly discussed software implementation projects from the selection of the software to its implementation and use. At times, as in the case of General Electric’s implementation in its Lexington Kentucky plant, the coverage is quite extensive and broad. These discussions are found in trade association proceedings, trade magazines, and industry analyst reports. In particular, the American Production and Inventory Control Society (APICS), a professional educational society targeting the manufacturing planning professionals, publishes presentations about software use from its yearly conference proceedings. I used the proceedings from 1960-1996, with some minor gaps, to capture the general use of planning software. Other sources of information include *Manufacturing Systems, Harvard Business Review, and Journal of Inventory and Production Management* (an APICS Journal) and various articles from ABI-Inform and Factiva.

A potential concern of just considering manufacturing oriented sources is that they do not capture the perspective of other potential users within the organization. Even though planning software targets manufacturing users, technical professionals were responsible for implementing and maintaining this software. To capture their perspective, I gathered information from the Systems and Procedure Association, which represented the “systems men” often responsible for designing and maintaining these applications in the late 1950s and early 1960s. Secondary sources, in particular, the technology historian Thomas Haigh’s work on systems men (2001) proved invaluable. I supplemented this data with information from technically-oriented trade magazines such as *Datamation, Computerworld, Mid Range Systems,* and *Software Magazine.*
Other important users included the accounting department. Early in the introduction of the computer, the responsibilities for justifying, maintaining, and managing the computer reported through the accounting function. Although controllers did not use planning software on day to day bases, they significantly influenced which functions were automated and were interested in costing output from these systems. The Controllership Foundation administered a series of surveys to document what applications manufacturers (and other industries) developed in the mid 1950’s. This group also provided detailed reports and bibliographies. Other important sources of the accounting perspective included accounting professional journals such as *The Journal of Accountancy*.

Market intermediaries also played an important role in shaping how manufacturers used planning software. This group of actors, in particular consultants, trade associations, and industry analysts, influenced how customers interpreted and used planning software. I was fortunate enough to gain access to the archival records of three significant firms – Advanced Manufacturing Research (AMR), IDC, and Michael Hammer’s consulting group. To supplement this information, I also interviewed a small number of consultants involved in the implementations of planning systems.

Lastly, it is important to document the technical history of the software and the software vendors. Since software integrates with other technologies, the technical history is quite broad. Changes in hardware, data management systems, programming languages, communications technology are well documented in other historical accounts. I leverage these sources wherever possible. Sources of information about the software vendors themselves included: several books written about executives of leading software firms,
recollections from IEEE Annals of the History of Computing, SEC submitted information such as 10-Ks and S-1s, and Wall Street Reports from 1982-present.

4.3 The setting: U.S. manufacturing and planning software

Manufacturing organizations typically consist of the business functions that design, build, sell, and distribute goods as well as more horizontal functions like accounting and human resources. This historical study is mostly interested in the administrative functions that plan and control the production and inventory processes. These functions address the following types of questions: Given demand, what inventory is needed to manufacture the products? When and how much of the inventory is needed? If some thing goes wrong with the production process, e.g., a machine outage, what changes can be made to minimize the disruption? In general, the manufacturing planning and control function tries to optimally satisfy particular customer demand with the most efficient level of inventory and operations possible. For a slightly more detailed description of the differences in manufacturing styles and a brief history of the manufacturing production planning function leading please see Appendix 4A.

Manufacturing planning software automates the processes associated with planning and controlling product and inventory within the manufacturing process. This software, for example, determines how much inventory is needed to meet projected demand, helps the manufacturers make changes to product plans as manufacturing conditions change, and
manages all the data associated with manufacturing products. Today, this software is integrated into a broader suite of functions such as accounting and human resource management in software packages known as enterprise resource planning (ERP). Well known examples of a software vendor in this are SAP and Oracle.

Technically, software consists of the programs that manage the user interface, data and system requirements, and the process logic that automates the business process. Planning software cannot work on its own and must integrate with other technologies such as hardware, database management software, and operating systems. Given this interdependence, I take a systems view (Hughes, 1983; Tushman & Murmann, 1998) of planning software, which includes both the components internal to the software itself and these external components required to make the software run. The main technical unit of analysis is the software system design (Henderson & Clark, 1990) – how the different system components interact with each other to perform the planning software’s tasks. Essentially, there were two distinct architectures during this time period. Originally, planning software was designed as a host-based system which means all the processing happened on a host computer, typically a mainframe or minicomputer. By the early 1990s, this architecture changed into a distributed architecture called client/server, which distributed the tasks between different components such as mainframes and personal computers. For a more complete description of the technical characteristics and significant technical changes during this period, please see Appendix 4B.

With this basic understanding, let us turn to the historical account. A summary of the key historical events is described in Appendix 4C.
4.4 1950s: A “Practical Approach” to use of planning software

Manufacturing firms were some of the earliest adopters of business computers. By 1959, manufacturing represented 42.4% of all computers outlays in the private sector in the U.S. (Phister, 1979, p. 444). Given the high cost of ownership, large manufacturers were the early adoptors. By 1967, 100% of the large manufacturers had a computer, but only 2.18% of the whole industry had made a purchase (Phister, 1979, p. 447). However, this did not mean that smaller manufacturers were not exposed to early uses of the computer. To the extent that smaller manufacturers used computers, they initially rented time from a service bureau. Smaller manufacturers would more substantially enter the market for computers and software later with the introduction of cheaper computing in the forms of mini and micro computers.

In order to make these computers useful, large manufacturers initially had to develop applications. In the earliest stages, independent software packages did not exist and computer hardware vendors did not offer unique application programs. Hence, the business community originally called the technology “computer application”, signifying the close relationship between the computer hardware and the tasks it was programmed to perform. Historians believe that the term “software” emerged in the late 1950s (Haigh, 2002). The use of the term became more prevalent in technology oriented trade magazines such as Datamation by the middle of the 1960s. However, the technical community seemed to adopt the term “software” well before the manufacturing business practioners (and perhaps all business practioners). The first use of the term “software” I am aware of
in the manufacturing literature was in a 1964 American Production and Inventory Control Society (APICS) presentation by Herbert Grohskopf, a manager of Scientific and Computer Applications at the Lummus Company (Grohskopf, 1964, p. 66). However, he was a member of the technical community as opposed to the business community of production planners and manufacturing managers. The use of the term “software” in the APICS proceedings becomes much more common in the early 1970’s which coincides with the introduction of pre-packaged software applications. In the 1950s and into the 1960s, manufacturers developed most manufacturing applications themselves, often with the aide of consultants. Therefore, the use of “computer application” is not insignificant because it signals that manufacturers thought of software as closely interrelated to the computer. Its function was to make the computer useful.

From a use perspective, it is important to understand what kinds of applications the manufacturers initially developed, how they decided upon these particular applications, and how they deployed these applications. From 1954-1957, the Controllership Foundation conducted surveys of how large firms used computers, providing a broad view of early manufacturing applications. Overall the survey identified 68 different applications developed. Table 4.4.1 lists the top applications deployed by establishment (not firm, i.e., a firm may have multiple installations) from 1954 – 1957 as well as some summary statistics about the total number of applications installed per computer. Despite the diversity of applications, accounting applications were the most pervasively deployed. Payroll applications were developed in 75% of all implementations and cost accounting in 46%. In fact, of the 189 implementations, only 12% did not deploy any accounting
function at all. By comparison, manufacturing applications, such as material production requirements and inventory control, were implemented much less frequently, 33% and 32% of the time respectively.

Technological constraints at the time certainly influenced this usage pattern. Technically, these applications were architected in a way that all the transaction and data processing occurred on a host machine. Today, we call this application architecture host-based. In addition, these applications were proprietary to the hardware system because they used different and incompatible operating and data management systems.

Applications written for an IBM 650 did not run on the UNIVAC and even the IBM 1401. This changed for IBM in the 1960s with the introduction of the IBM 360 product line.

---

Table 4.4.1  Some computer applications developed in U.S. manufacturing, 1954 – 1957

<table>
<thead>
<tr>
<th>Application</th>
<th>1954</th>
<th>1955</th>
<th>1956</th>
<th>1957</th>
<th>Total</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payroll</td>
<td>2</td>
<td>38</td>
<td>43</td>
<td>59</td>
<td>142</td>
<td>1</td>
</tr>
<tr>
<td>Labor</td>
<td>2</td>
<td>26</td>
<td>26</td>
<td>42</td>
<td>96</td>
<td>2</td>
</tr>
<tr>
<td>Cost Accounting</td>
<td>2</td>
<td>31</td>
<td>21</td>
<td>33</td>
<td>87</td>
<td>3</td>
</tr>
<tr>
<td>Enq.</td>
<td>2</td>
<td>18</td>
<td>22</td>
<td>34</td>
<td>76</td>
<td>4</td>
</tr>
<tr>
<td>Employee</td>
<td>2</td>
<td>20</td>
<td>20</td>
<td>28</td>
<td>70</td>
<td>5</td>
</tr>
<tr>
<td>Material Prod.</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td>27</td>
<td>62</td>
<td>6</td>
</tr>
<tr>
<td>Inventory Control</td>
<td>1</td>
<td>15</td>
<td>18</td>
<td>27</td>
<td>61</td>
<td>7</td>
</tr>
<tr>
<td>Sales Rev.</td>
<td>2</td>
<td>9</td>
<td>14</td>
<td>30</td>
<td>55</td>
<td>8</td>
</tr>
<tr>
<td>Total</td>
<td>3</td>
<td>45</td>
<td>59</td>
<td>82</td>
<td>189</td>
<td></td>
</tr>
<tr>
<td>Avg. Apps/Site</td>
<td>8.6</td>
<td>6.4</td>
<td>7.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Controllership Foundation Surveys, 1954 - 1957
which supported interoperability between different products within the IBM family but not between hardware from different vendors (Fisher et al., 1983). Although significant advancements would be made at the component levels of a software system, the host-based architecture would remain until the late 1980s.

Since the application was tightly coupled with the computer, the technical constraints of the computer itself may have influenced which applications were deployed. By today’s standards, early business oriented computers were technologically crude. Particular issues related to running business applications included available memory, data and program management, and user interfaces. Lack of memory impacted the amount of computations and procedures a computer could manage, limiting the computational intensity of the applications (Ceruzzi, 1998). Loading and unloading a program also took significant time, restricting the number of applications that could be run in a given day. Early business computers also lacked sophisticated database management systems, requiring programmers to manage data and the program logic simultaneously (Haigh, 2001). In today’s applications, managing data integrity and retrieval is distinct from the business logic of the program, but initially hardware interface, data management, and application logic were all intertwined together. Typically, data was stored in a file structure, making it difficult to relate data items, such as component part information, together. It was not until the 1960s that hierarchical database structures supported more complex data relationships. In addition, the data storage, magnetic tapes and drums, only permitted sequential access to the data (Ceruzzi, 1998). Processing through the data
sequentially significantly slowed data retrieval. Data storage and management limited the amount of data complexity a computer could manage.

Finally, although the computer used tape and electronic means to process data and the programs within the computer, getting data into the computer still required card input devices. Similarly, from the output perspective, terminal interfaces did not exist, so the main outputs for these computers were interfaces into card sorters or printers. In a sense, these systems were only partially automatic often requiring manual intervention to transfer data between devices, as evident in the list of input/output devices in the Controllership Foundation survey. Flow charts of computer system activities also showed the manual interaction and the role of additional peripheral devices (see, for example (Laughna, 1960)). Given the design limitations of the computers and the significant costs, many applications were managed in a batch mode meaning that transactions were accumulated and processed together in “batches”(Cortada, 2004). Batch processing limited the amount of individual transaction changes that could occur during the processing of the batch, resulting in separate error and exception handling.

The minimal memory, poor data and program management, and no user interfaces of the computer at the time certainly favored applications that managed repetitive data processing such as payroll. However, managing the complex data relationships in a bill of material would have been extremely difficult given the poor database management tools and sequential data retrieval restrictions. Yet, as Tables 4.1.1 shows manufacturers developed these types of applications, at least a third of the time. In addition, despite the constraints to run multiple applications on the same computer, large manufacturers tended
to automate multiple applications per installation. From 1955-1957, the average large manufacturer that implemented computer applications developed 8 applications per installation, cutting across 3 different functions. For example, International Harvester implemented 9 applications in 1955 at their Farmall Works facility in Rock Island, Illinois, including 3 in accounting, 3 in manufacturing, 1 in each of engineering, human resources, and sales (Courtney, 1958).

Despite these technological constraints, however, it may have been more cost effective to deploy several applications on one machine. The financial commitment to new computers was significant, ranging from $60-200K per month in absolute terms depending upon the size of the computer (Yates, 2005). To make the computer investment cost effective, it may be necessary to deploy several applications on one machine. However, consultants at the time argued that computers could be cost effective even if only payroll (of a certain size) was implemented (Higgins & Glickauf, 1954). In addition, even when purchasing less expensive and powerful computers, such as the IBM 650, manufacturers still implemented multiple applications. From the Controllership Survey data, the mean number of applications on the IBM 650 was 5.8, only 2 applications less than the mean of larger computers like the UNIVAC. If manufacturers deployed more applications to offset costs, we would expect that smaller machines would support fewer applications. Technological and cost constraints alone do not sufficiently explain why manufacturers tended to deploy information intensive applications across several business functions.

In fact, it is not entirely clear how well manufacturers understood these technical limitations in advance of purchasing the computers. At the time of commercial release,
manufacturers were uncertain about what the new computer was and which applications should be developed. When General Motors was faced with a computer purchase, management went from department to department seeing for what they could possibly use it (Fisher et al., 1983). In comparison, as noted in the previous chapter, the insurance industry developed an understanding of some of these limitations prior to the commercial introduction of the computer through commissioned studies of the computer. Insurers significantly underestimated the cost of operation – in particular the time and effort required to program and maintain an application. Tabulating machines could not store procedures so they did not have to be programmed. Insurers learned about the cost of programming as they deployed and used the applications. As a result, a fuller account of the initial deployment patterns should take into consideration how manufacturers understood and justified their purchase of computers and applications.

General Electric's (GE) purchase and implementation of computer applications in 1954 provides a typical and well documented example of this process. In 1954, General Electric (GE) purchased one of the first commercially available UNIVACs to run business functions in its new Major Appliance plant in Louisville, Kentucky. GE advocated a "practical approach" in which well known and functioning business applications were converted to the UNIVAC from tabulating machines, with the goal of adding more complex problems as they became more familiar with the computer. Roddy Osborn, a manager of the Business Procedures Section at the Plant, involved in the project described the general thinking as to "first taking care of such obvious needs as eliminating the
Figure 4.4.1: GE’s understanding, justification, and deployment of computer applications, 1954

<table>
<thead>
<tr>
<th>Processes</th>
<th>GE’s Action</th>
<th>Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding</td>
<td>Emphasized comparison to tabulating applications</td>
<td>Described “innovation” in comparison to tabulating machine tasks, but recognized “mechanical brain”</td>
</tr>
<tr>
<td>Justification</td>
<td>Cost reduction</td>
<td>“Possible savings can be evaluated in terms of: (a) salaries, (b) “fringe benefits” of clerical workers being displaced, and (C) the related costs, if recoverable, of occupancy and furniture and equipment depreciation” (p. 104).</td>
</tr>
<tr>
<td>Deployment</td>
<td>Automate “routine dog work” of pre-existing systems</td>
<td>“The idea of making a drastic jump from present operations to a completely new system controlled by a mechanical brain will cause him to throw up his hands in despair” (p. 101).</td>
</tr>
</tbody>
</table>

Source: (Osborn, 1954)

drugery of office work (Osborn, 1954, p. 101)” and then moving on to more complicated problems.

Osborn’s description of the implementation considerations for these applications provides a glimpse of how GE understood, justified, and deployed these initial applications, which is summarized in Figure 4.4.1. Similar to insurance firms, GE interpreted computer applications by making several product category comparisons – to tabulating machines/other manufacturing equipment as popularized by the consultant John Diebold’s vision of a fully automated factory (Diebold, 1952) and to the “human brain.” These different comparisons focused on different applications: the machine comparison favored deploying the computer to process data to give managers more time to make
decisions on their own; whereas, the human comparison favored using the computer to
improve human decision making itself. Even though GE recognized that computers had
“Great Brain” like qualities, they initially focused on thinking about the computer in
relation to the tabulating machine. They described the innovations of the computer in
terms of the reduction of human to machine intervention required to process a task and
manage exceptions. To justify the UNIVAC’s purchase, GE calculated the potential cost
savings in comparison to tabulating machines. Specifically, GE measured benefit in terms
of eliminating clerical workers and did not consider operational improvements such as
inventory reduction:

And, in computing the breakeven point, savings were evaluated only in
terms of salaries, space rentals and equipment depreciation applicable to
those clerical jobs eliminated in four limited routine applications. No value
was given to such intangibles as more prompt reporting, the ability of
management to make more informed decisions quicker, and reduced
investment in inventory. Had these factors been included, a break-even
point could be reached with fewer applications and less computer time
(Osborn, 1954, p. 100).

GE also shied away from using the computer to automate managerial decision
making processes because they were concerned about the organizational response to the
new technology. In particular, Osborn thought it was important to get buy in from the
various functional departments who would use the applications’ output, and it was not
clear how these departments would react to the new technology. Osborn was particularly
conscious of the functional managers’ concern that if the computer application did a really
good job, it could replace the department and potentially him. Therefore, he emphasized
that rather than “making a drastic jump from present operations to a completely new
system controlled by a mechanical brain” (Osborn, 1954, p. 101), GE chose to “convert”
eexisting clerical functions to the new machine. In addition, he argued that the computer
application should be deployed in a centralized manner – various departments would
provide the data to a central processing group who would perform the tasks and then return
the output to the department, usually in the form of a report.

Focusing on clerical routines and centralizing administration of the computer
applications managed the implementation risk in two ways. First, centralizing processing
can take advantage of data and task interdependencies between these functional units,
lowering the cost of ownership. Second, the functional department still retained control of
the data and making the important business decisions based upon this data such that there
was “no usurping of management prerogatives (Osborn, 1954 p., 106). Functional
managers would be more comfortable using the computer applications. Based on these
guidelines, GE decided to deploy four applications: payroll, material scheduling and
inventory control, order service and billing, and general and cost accounting.

Specifically, Osborn described the functionality of their material requirement and
inventory control applications in this way:

Material scheduling and inventory control involve the “explosion” of a
production schedule into detailed material requirements by days and weeks
for any given period. Inventory balances are maintained by posting receipts
and withdrawals daily. In addition, open orders on vendors are recorded
and adjusted daily. Proposals to change production schedules can be
analyzed to determine the extent to which they can be satisfied by present
inventories and inventory commitments (Osborn, 1954, p. 102).
This “explosion” and maintenance of inventory balances essentially consisted of the clerical tasks required to process material requirements for production. These applications processed the information about the various component materials required to produce an end product and the schedule required to build it. For large discrete manufacturers (see Appendix 4A for a discussion about discrete and process manufacturing), the bill of material typically contained this information. The bill of material showed the dependent relationships of component parts as well as quantities required for production. Material production requirement applications would manage the information on these bills of materials and help keep it updated as the product was produced. In fact, as mentioned, one of the first commercially developed applications was IBM’s bill of material processors in 1959, which leveraged a hierarchical database design to help manage these relationships. Inventory control managed the information about the inventory balances for each inventory item which was used to manage production as well as for costing purposes. Therefore, the manufacturing applications tended to process information needed to manage the production planning as opposed to automating the decision processes of doing the planning itself. This process still lay in the hands of the manufacturing department.

GE functional managers seemed to buy into the belief that the centralization of managing computer applications can still support departmental control of the data and decision making. In a 1955 report commissioned by the Controllership Foundation, several GE functional managers argued:

The decentralization of operation and executive control thereof does not necessarily imply a similar decentralization of services. In other words, it is
possible to process data centrally, with the control of the operations based on this data in the hands of the individual departments.

As an example, the UNIVAC installation at Lousiville processes detailed payroll and inventory records furnished to it by department in that geographical area. The details from which the process is performed, the resulting tabulations and summaries of these data are all returned to the individual departments for their action. The UNIVAC simply serves as a central processing organization (Paxton, Vinson, Wichman, Oswalt, Lewis, & Hurni, 1955, p. 62).

Finally, Osborn recognized that consultants play an important role in the purchasing and implementation process. In fact, for this project, GE worked with Arthur Andersen. It is informative to see how GE used the consultants. Osborn described the consultants’ role as educators and facilitators in the implementation, with the specific tasks of “training personnel, arranging management orientation meetings, and planning for conversion of initial applications (Osborn, 1954, p. 106). Osborn preferred using consultants in this role instead of computer firms, perhaps to avoid becoming too reliant on any one vendor. Thus, consultants became engaged with the implementation of computer applications early in the adoption of computer. The consulting industry during this time also grew significantly. In 1957, a trade magazine listed 42 firms under the grouping of consulting services, but by 1967, there were 226 firms under programming services alone (Bromberg, 1967). A manufacturer cited statistics that from 1955 to 1966, the total number of consulting firms in the U.S. had grown by more than 900 to 2600 and billings had increased from $200 million to $725 million (Goffredo, 1969). Eventually, some consultants, including Arthur Andersen would even develop their own solutions, and over time their roles would expand from educators and implementers to in many cases the
purchasers. As we shall see, the shifting dynamics of the “consumption junction” and competition among these intermediaries influenced the manufacturer’s usage pattern.

In summary, faced with the uncertainty of the new technology and how it would be received, GE opted for a “practical approach” which conservatively focused on meeting minimal thresholds of success as opposed to developing a new system that fully optimized the not well understood benefits of the computer. GE’s practical understanding and justification in addition to organizational concerns influenced how they initially deployed the computer applications. Manufacturers justified their purchase based upon the total cost reduction in replacing tabulating machines and eliminating clerical workers. As a result, manufacturers chose to automate applications that replicated well documented and systematized functions. In order to minimize the risk of failure, manufacturers deployed these applications in a manner to minimize organizational change. The common term to explain this use of applications was “automation,” reflecting the focus on displacing costs, programming repetitive tasks, and separating data processing from decision making. It fit in with the broader theme of automation of the manufacturing process with numerically controlled machinery, popularized by Diebold’s vision of the automatic factory (Diebold, 1952).

This practical approach would dominate the way manufacturers thought of and used computer applications throughout the 1960s and into the 1970s. For example, an educator described how Fairchild used computer applications as follows:

... all of this [production] data is cranked into the computer, the computer cranks out reports and paperwork automatically, and the work is divided so that each material specialist has a segment of the computer’s output, and he
analyzes it, manages it, makes decisions, and tells the computer what to do. (Ammer, 1960, p. 162)

Similar to GE, Fairchild thought of the computer in relation to other processing equipment and not in relation to human decision making. They described the computer’s function to “crank out” reports, just as manufacturing equipment grinds out products. In addition, the deployment of the computer application centralized the processing of information, but distributed control of decision making with the functional departments. Lastly, Fairchild measured the benefit in terms of cost reduction, in this case, reducing the materials department from 135 clerks to 50 (Ammer, 1960).

However, in the 1960’s an alternative use of computer applications based upon a different understanding and justification emerged.

4.5 1960’s: Alternatives to the “Practical Approach”

In the 1960s, some manufacturers began developing applications that automated managerial decision making instead of clerical information processing. One of the first decision-oriented applications within manufacturing was called material requirements planning or MRP. MRP helped the production planner calculate material requirements based upon dependent relationships within a product’s configuration, production requirements, and end-product demand. In contrast, the earlier material requirements application described at GE simply generated information about the inventory
requirements for the production planner who would then make any calculations without the aid of the computer. It is believed that in 1961, J.I. Case, under the supervision of Joseph Orlicky, implemented one of the first MRP solutions on a computer. During this same time period, other manufacturers such as John Deere and American Bosch also implemented computer-based MRP systems (Lilly & Smith, 2001). Initially, the total number of MRP implementations remained relatively small, totaling only 150 by 1971 (Orlicky, 1975), but by the 1980s it became the dominant use of manufacturing applications. Therefore it is important to determine why an alternative use of a computer emerged and in particular why it was MRP.

A good place to start is to consider the technological changes that occurred during this time period. Although there were significant changes at the component level, especially in programming, data storage, and user interfaces, the over all architecture of the software products remained the same. Planning software still had a host-based architecture. Nevertheless, it is possible that the component level changes led to an alternative use.

Specifically, new generations of computers, in particular, the solid state computers with disk drives introduced in the early 1960s allowed for random access to data and increased data storage, greatly improving data retrieval capabilities of computers (Ceruzzi, 1998). On the software front, hierarchical database management systems from computer manufacturers and independent software providers were commercially available in the late 1960s. This software abstracted some of the data management requirements from the business logic and provided additional structure to manage data relationships, facilitating
more robust data schemas and data processing. Processing power increased such that these new systems did not have to run in “batch” mode but supported real-time processing (Chandler, 1997). In addition, COBOL was introduced as a programming language that facilitated writing the part of the software program that automated the business processes. The introduction of a well defined programming language and data management applications meant that the layers of the software technology stack (see Appendix 4C) were becoming distinct and easier to manage. Lastly, by the end of the 1960s, terminals were introduced which enabled business users direct access to the computer and data manipulation.

Quality improvements in these dimensions – non sequential data access, real-time processing, and direct user interface – allowed for more decision-oriented uses of the computer. For instance, more sophisticated data storage and management could support the non sequential and repetitive data querying aspects of the decision making process. Determining the material requirements of a complex bill of materials required pulling data multiple times from different information sources. But, these improvements could also better address the needs of processing repetitive information tasks. For example, terminal interfaces could reduce the time and effort required to correct data errors. The technological improvements were not tied to a specific use of the computer. In other words, it was not as if once manufacturers recognized these specific technological innovations, they decided to implement decision-oriented applications like MRP. In fact, it appears to be the opposite. Some manufacturers were committed to implementing MRP, regardless of what the computer was.
Rather, a more complete explanation of the emergence of decision oriented applications recognizes that manufacturers began to shift their understanding and justification of computer applications. Through extended use of existing computer applications, manufacturers were learning that their early conservative use of the computer applications, although highly successful in some cases, generally did not produce the expected returns. Consultants conducted studies on the return on investment of early computer use, helping diffuse this performance information. Learning about the inefficiencies did not lead manufacturers to abandon the use of computer applications, perhaps because of the significant investment both financially and organizationally. Rather, it encouraged some practitioners to re-emphasize using computer applications to manage decision making processes that were earlier ignored with the practical approach.

One well-cited consultant study was done by the management consulting firm McKinsey & Company in 1963 and reported in a *Harvard Business Review* article (Garrity, 1963). They surveyed 27 companies in 13 different industries and found that only 9 had financial success. The McKinsey report generally blamed the poor financial performance on management issues. It argued that effective leadership, soundness of planning and control tools, operations management involvement, and the caliber and role of the technical staff explained the difference in performance (Garrity, 1963). The problems with the computers were not technical in nature, but in the conservative, practical uses customers had created for it.
McKinsey argued that:

In the average company [poor financial return], the computer has been used on a much narrower front. Only one or two divisions are participating in the computer system program. Typically, the computer systems program is restricted to routine office functions such as payroll, billing, and inventory record-keeping (Garrity, 1963, p. 8).

In some cases, executive management views the computer as merely a super accounting machine and turns it over, as a matter of course, to the traditional data processing department (Garrity, 1963, p. 174)

According to this logic, applying the computer to GE’s pre-existing and well automated systems should be avoided because they do not generate the necessary benefit. Those advocates who did not share the initial “practical vision” of GE would use these studies to further their argument for using the computers in more imaginative ways.

Consultants and practitioners also began to expand their justification of computer applications to consider operational benefits: reducing investment in inventory, increasing inventory turns, and improving production efficiency. For example, in 1967, Harold Plant, an administrator for manufacturing markets at RCA, identified potential computer investment savings in inventory management, shop labor expense, reduction of purchase parts, overdue orders, salaries, and maintenance expense (Plant, 1967). According to this logic, applying the computer to well automated standard applications should be less of a priority because they do not generate the necessary benefit (Glaser, 1967; Orlicky, 1961). Automating payroll does not help with lowering inventory levels and shortening order lead time. Instead, manufacturers should consider addressing those processes that may have more significant impact on the bottom line. Computer applications should assist the
planner in making decisions about what inventory levels to carry and when new inventory should be ordered rather than simply manage inventory records. To make these uses more salient, practitioners, such as Joseph Orlicky of J.I. Case and later IBM, re-focused attention on the computational characteristics of the computer by emphasizing its more human like qualities (Orlicky, 1961). In 1969, he published the book *The Successful Computer System* which argued that a computer “amplifies man’s intellect” (Orlicky, 1969, p. 9). Note it does not replace man’s intellect – the manager is still needed but the computer could be a really useful tool to make him smarter.

Therefore, in the 1960’s, an alternative understanding, justification, and deployment emerged (see Table 4.5.1). Where the practical approach understood the computer application in relation to tabulating equipment and justified using the application to automate repetitive clerical tasks on the grounds of cost reduction, this new use understood the computer application in relation to the human brain and justified using the application to automate managerial decision making processes on the grounds of increasing operational benefit. The specific application that manufacturers began to deploy was MRP.

It is believed that Joseph Orlicky implemented the first MRP application on an IBM machine in 1961 at J.I. Case, a farm equipment manufacturer in Wisconsin. Joseph Orlicky himself would become an important figure in promoting MRP in the United States. He came to the United States shortly after WWII from Prague. From his earlier book on the use of computers, he clearly saw using the computer as a tool to more effectively plan and control material requirements within a manufacturer. Soon he began working in
Table 4.5.1  Comparison of “Practical Approach” with new MRP use

<table>
<thead>
<tr>
<th>Use characteristics</th>
<th>Practical Approach</th>
<th>New MRP use in 1960s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding</td>
<td>Focused on similarities with tabulating machines</td>
<td>Focused on similarities with human decision making processes</td>
</tr>
<tr>
<td>Justification</td>
<td>Cost reduction</td>
<td>Operational efficiency</td>
</tr>
<tr>
<td>Deployment</td>
<td>Automated “routine dog work” of existing applications</td>
<td>Automated decision making processes routines in planning tasks</td>
</tr>
</tbody>
</table>

production control at J.I. Case and he was an early participant in APICS, helping found the Minneapolis chapter. In 1962, he left J.I. Case to work for IBM as a Manufacturing Industry Education Manager and an Industry Consultant, where he was instrumental in teaching IBM clients about MRP as well as helping develop IBM’s own software solution. Later, in 1975, he wrote the book *Material Requirements Planning: The New Way of Life in Production and Inventory Management* which would become a key reference for MRP style planning.

MRP helps production planners determine when new material is needed. This was not a new calculation, as at the time, many manufacturers were using a different technique, the re-order point (See Appendix 4A for a brief history). Re-order point essentially determines the inventory level at which additional inventory for the part needs to be re-ordered. It operates using the data of historic use and forecast to calculate economic order quantity and safety stock requirements for each inventory part individually. Over the
years, mathematical techniques had been applied to this technique to improve forecasting and safety stock calculations. Similar to re-order point, MRP was initially interested in determining when new material was needed. In this sense, the first uses of MRP were to replace the re-order systems.

There were sharp contrasts between MRP and re-order point systems:

The second basic method [MRP] is to calculate requirements for sub-assemblies and parts (collectively called components) based on the needed quantities of the higher-level assemblies, usually finished products, in which they are used. Demands for the finished product assemblies are determined from forecasts, backlogs of orders or both and modified by existing inventories to prepare a “Master Schedule” of production requirements by time periods which provides the input to “material requirements planning” to determine component needs (Plossl & Wight, 1971, p.3)

Unlike the re-order point method which calculates re-order points for each part individually, MRP makes the distinction between the demand for the end-product and the inventory requirements of the components required to build the product. This basic distinction between independent demand for the final good (or sub-assemblies) and dependent demand for its component parts separates MRP from re-order point. Independent demand must be forecasted and can function with re-order point logic, but dependent demand is calculated, not forecasted. Orlicky is famous for saying, “Do not forecast demand when it can be calculated” (Plossl & Wight, 1971, p. 10). In essence, MRP schedules when the inventory is needed to meet production requirements set by the master schedule. This scheduling involves using end-item due dates with standard lead-time offsets to calculate the date requirements for each dependent item in the product’s bill
of material. In theory, MRP synchronizes the inbound flow of component parts with production requirements to minimize inventory holding and lead-times in the production equipment. Such inventory planning requires significantly more calculations and data from several different places, including bill of material, production schedules and routings, forecast, and back-log. In addition, material requirements change frequently with changes on the shop floor and customer orders. Often the MRP calculations needed to be re-run to respond to these changes. One method was to simply re-generate the entire plan, but Orlicky argued for net change MRP which would reduce the re-running of the MRP logic to only those areas affected by the change.

Therefore, the proposed new use was not to automate the already existing re-order point decision process. Rather, Orlicky and others proposed automating an entirely new managerial decision process to determine material requirements for production. As mentioned, Orlicky’s MRP system was designed specifically with the computer in mind (Orlicky, 1975). The MRP process includes many tedious calculations and may have to be re-run. Production environments with complex product bill of materials with many sub-assemblies made manual requirements calculation impractical. The computational speed of the computer made these calculations much more feasible. But, why was there a need for a new approach in the first place? And, why was the proposed alternative MRP in particular? An entirely different approach to material planning could have been proposed. For instance, during the same time period in Japan, manufacturers were exploring material
management techniques, eventually called Just-in-time, that were more reactive and flexible than rationalized and planning oriented (Cusumano, 1989).\textsuperscript{13}

At the time, there was concern that the re-order point procedures and material expediting practices were not sufficient given the changing economic, competitive, and production environment. The U.S. manufacturing environment started to experience a series of economic and competitive changes that increased focus on production and inventory processes. After 1954, the U.S. economy experienced several recessionary periods (see Appendix 4A) which encouraged manufacturers to more efficiently manage the inventory and production process. With these fluctuations, however, forecasting demand and improving customer service became more important. To respond to these changes, the production process itself increased in complexity. Manufacturers introduced new products more frequently and the products became more complicated. For example, in a 1960 APICS presentation, a Chrysler planner detailed how the variations of the number of components of a car, e.g., engine type, color, trim, increased the complexity of production planning requirements (Laughna, 1960). He estimated that production planning required 10,000 individual computations, 5,000 individual schedules, and a total 55 pages of synchronized data. It was not clear how re-order point logic could manage this complex production environment.

\textsuperscript{13}The MRP phenomenon appeared to be localized to U.S. and Europe. Why this is the case is an interesting question, but beyond the scope of this dissertation. However, it is important to point out that similar to the U.S. the development and adoption of Japanese manufacturing principles were heavily influenced by users, in particular Toyota (Cusumano, 1989). The influence of the Japanese JIT systems on the U.S. interpretation and use of MRP will be discussed shortly.
In addition, management was beginning to view the manufacturing function more strategically and less as a cost center. In a series of *Harvard Business Review* articles dating back to 1966, Wickham Skinner of Harvard Business School emphasized manufacturing as an essential part of corporate strategy (he later wrote a popular book based on these articles (Skinner, 1978)). Within manufacturing itself, the production planning position began to take on more of a managerial role. By the late 1950s, those professionally responsible for planning and maintaining production and inventory control were still staff positions. A group of practitioners started the professional society American Production and Inventory Control Society (APICS) in Cleveland, Ohio in 1957 to help promote production and inventory planning personnel and to establish more standards in education of production and planning techniques. APICS deployed a regional structure hoping to attract individuals at the local level as an executive office was not established until 1968. This allowed for local members, such as Joseph Orlicky, to have significant influence.

Two other prominent figures in APICS were George Plossl and Oliver Wight. George Plossl worked at Stanley Works in production control and joined IBM as a consultant briefly in the mid 1960s. Oliver Wight worked in inventory control at Raybestos and subsequently moved to Stanley Works. He too joined IBM briefly in 1965 as an educator and consultant, but left with George Plossl to form a consulting company. After this brief partnership, Wight formed a consulting company in his own name which still exists today. He also helped establish APICS Fairfield County chapter in 1959. Both
men were frequent speakers at APICS events and were highly engaged in the broader education efforts, helping compile dictionaries of terms and bibliographies of references.

Perhaps most notable and significant was their efforts to promote a more integrated view of production and inventory control. They argued for effective planning and control based upon the use of timely information in their 1967 book entitled *Production and Inventory Control: Principles and Techniques*. For many, this book became a bible for production and inventory control practitioners (Lilly & Smith, 2001) and APICS would continue to update this approach in a series of production and inventory control handbooks.

In the book, they argued:

Working through an information system, planning, measuring actual performance against the plan and then presenting information to line managers who must take corrective action, production and inventory control’s function is to reconcile these objectives to meet the overall profit goals of a company (Plossl & Wight, 1967, pp.2-3).

Central to production and inventory control procedures was an “information system” that enabled planning and feedback mechanism to make “corrective” management decisions to meet “overall profit” goals. It should be noted that by “information system”, Plossl and Wight did not necessarily mean a computer system. They desired mass appeal for their principles and recognized that not every manufacturer could afford a computer system. So, information system simply meant well organized, documented, and accurate information about products, orders, and the production process. Orlicky’s MRP approach fit in with these more general and fundamental principles of production and inventory management.
espoused by Plossl, Wight, and APICS. MRP’s procedures to rigorously plan and control production material requirements throughout the manufacturing lifecycle exemplified what Plossl and Wight thought of as a well functioning information system.

MRP also synthesized the use and perspectives of the dominant users of early production planning systems. Essentially, there were three primary users of early applications: manufacturing personnel, accountants, and “systems men”. Each user had a different use and perspective about the purpose of the planning application. Manufacturing primarily used the output to complete its daily tasks of making sure there is enough inventory to fulfill forecasted demand and to respond to issues during the production process. As mentioned, much of this decision process was not automated because the planning systems only accurately recorded the production information. In contrast, accounting was interested in inventory data for costing, budgeting, planning, and financial statement preparation. These different uses created tensions.

Controllers often complained that production inventory data was not accurate enough for their costing purposes. An independent consultant interviewed 50 controllers in Michigan in 1960, with the general observation:

The production control records are inadequately kept, inaccuracies are rationalized, the production control people are close to the physical situation in the plant and merely use the inventory figures as broad guides or trend indicators. Much reliance is placed on specific observation, special counts and other methods for knowing what decisions to make relative to scheduling. Obviously, the major exceptions in the area were the companies having well planned systems on their data processing equipment ... The basic information contained in the inventory records is the same for both purposes. However, the manner and purpose for which it is used is different in production control and accounting. (Casenhiser, 1960, p. 132)
In many respects, the accountants had the upper hand in this debate. They were heavily involved in the purchase of the computer and the data processing groups responsible for programming and operating the computer system reported up through the accounting department. For example, computer uses for accounting typically were reserved for more convenient times. Richard Lilly, an IBM consultant and later manufacturing software entrepreneur, recalled how in the early 1960s while working for Norton in Worcester, Massachusetts, his team would have to travel to New York City on a Monday so they could use the IBM 1410 to test and run their bill of material explosions from the end-of-day Tuesday to 6 A.M. on Wednesday (Lilly & Smith, 2001, p. 16). Later, Wight suggested that production planning professionals needed to get more involved in application selection and implementation to make sure that their needs got met (Wight, 1965).

Nevertheless, accounting’s use of manufacturing information influenced MRP. Underlying the notion of an effective “information system” was the need for accurate data and having controls in place to maintain data integrity. Embedded in the notion of calculating material requirements and establishing feedback routines from actual production was an appeal to having an audit trail of the process taken and the changes made within the MRP system itself. It was not prudent to rely upon the tacit experience of manufacturing personnel. In fact, the accounting influence was significant enough that when manufacturing personnel began to understand the differences and effectiveness of Japanese Just-in-time production system in the 1980s, they identified the issues of the existing MRP systems as being too accounting oriented.
In contrast, “Systems men” had more of a technical than functional use of these computer applications. This group was responsible for designing and programming these applications. In the late 1950s and early 1960s, they started to push for “total integrated management control systems”, partly as a means to advance their position within the organization (Haigh, 2001). Some went to the extreme to argue that firms should redesign their business processes to become more integrated. Such business process “re-engineering” would not take hold until much later in the 1990s.

A less radical version argued that manufacturers should use system designers to build “total management control systems” for the company (Crouch, 1960). The vision of the “total management control systems” automated the process of supplying information to management as opposed to replacing decision making with computers. In this respect, this system view is not much different than what GE envisioned the role of its data processing center (Osborn, 1954). Similar to Osborn’s use of system, the total systems view preserved the sensitivity toward potentially replacing managerial functions. The main difference between Osborn’s system and the system designer’s total system was that Osborn thought of the system in data processing terms. The integration was based upon taking advantage of common data input/output requirements between functions, e.g., sharing inventory records between finance and manufacturing, to reduce the data processing costs (Osborn, 1954). He was less interested in the potential benefits of sharing data in the decision making process.

Beyond leveraging common data elements, the total management system would integrate information allowing the manufacturer to quickly respond to supply and demand
changes (Crouch, 1960). Information systems were meant to keep data managers informed at all times by providing timely information to help make decisions instead of providing periodic reports based on historical data designed from pre-determined data requirements. The “information pyramid” – the different planning, execution, and control decisions a firm must make and the structure of a management system to provide this information – became a symbol of the total systems movement ((Haigh, 2001), see (Head, 1967) for a critical discussion of the pyramid).

In practice, however, the total systems movement failed to materialize in the 1960s and 1970s due in part to sharp criticism. One of the most vocal was Harvard Business School professor, John Dearden, who voiced his issues through a series of Harvard Business Review articles, culminating with the well known 1972 article, “MIS is a Mirage” (Dearden, 1972). Dearden criticized the very concept of system professionals and insisted that management did not have uses for real time data as the systems view suggests (Haigh, 2001). In practice, implementing a system proved very difficult and firms began speaking against such an approach, e.g., Westinghouse openly talked about how it decided not to move to an integrated computer system (Fritz, 1969).

Despite this criticism and lack of broad implementation of total integrated systems, this view influenced how manufacturers talked about and interpreted the computer. Discussions of computers during the 1960s and 1970s began to make the distinction between data processing and supplying information to make decisions (see for example, (Greiman, 1969; Kosla, 1969; Ringness, 1968; Schwartz, 1969; Whitney, 1968). Diagrams of system designs at APICS proceedings showed increasing functional integration as well
as information flows to support planning and control decisions (Plant, 1967). The systems view also promoted a more centralized view of managing data through functional/data integration and increased corporate involvement in processing information (Withington, 1969). At a more general level, Plossl and Wight’s arguments for an integrated view of production and inventory control reflected the systems view. Effective MRP systems included all three elements of the “information pyramid” – planning of material requirements with feedback from execution and controls in place to make effective management decisions. In fact, Dearden, despite his criticisms, pointed to integration in manufacturing as an example of potentially successful integrated systems at the departmental level (Haigh, 2001).

In summary, during the 1960’s manufacturing introduced MRP as an alternative to the recording keeping manufacturing systems deployed in the 1950s. MRP represented a new use of the computer that focused on automating the production planner’s decision process. Its use was intended to make the manufacturer more efficient. MRP itself developed in response to the changing competitive environment and the efforts of manufacturing personnel to increase its influence within the organization. MRP principles reflected the influence of other users of computer applications – accounting’s interests in accurate data and process control, and the system men’s desire for integrated information systems.
4.6 1970’s: The MRP Crusade – establishing a dominant use

At first, only a handful of manufacturers developed MRP solutions. Orlicky estimated that by 1971 there were approximately 150 and by 1975, 700 installations (Orlicky, 1975). By 1984, the industry analyst group IDC estimated that approximately 21,000 of this system or its next generation, MRP II had been installed (IDC, 1985). Essentially, MRP became the new dominant use of planning software. How did this happen? The heavy marketing campaign of the MRP technique, affectionately known as the “MRP Crusade” by the professional society APICS played a significant role in establishing the MRP as the new dominant use.
Table 4.6.1: Early adopters of MRP systems

<table>
<thead>
<tr>
<th>Company</th>
<th>Size</th>
<th>MFG Style</th>
<th>Exp with MRP</th>
<th>Reg or Net</th>
<th>Freq</th>
<th>Computer Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black &amp; Decker</td>
<td>&gt; $75M</td>
<td>Make-to-stock</td>
<td>3 Y</td>
<td>Net</td>
<td>Daily</td>
<td>IBM 360</td>
</tr>
<tr>
<td>Data Control Systems</td>
<td>&lt; $25M</td>
<td>Make-to-order</td>
<td>1 Y</td>
<td>Regen</td>
<td>Weekly</td>
<td>IBM 1130</td>
</tr>
<tr>
<td>Dictaphone Corporation</td>
<td>$25-74M</td>
<td>Make-to-stock</td>
<td>4 M</td>
<td>Regen</td>
<td>Weekly</td>
<td>Honeywell</td>
</tr>
<tr>
<td>General Railway Signal Company</td>
<td>$25-74M</td>
<td>Both</td>
<td>2 M</td>
<td>Net</td>
<td>Weekly</td>
<td>IBM 360</td>
</tr>
<tr>
<td>Markem Corporation</td>
<td>&lt; $25M</td>
<td>Make-to-order</td>
<td>6 M</td>
<td>Regen</td>
<td>Weekly</td>
<td>IBM 360</td>
</tr>
<tr>
<td>New Britain Machine Company</td>
<td>&lt; $25M</td>
<td>Both</td>
<td>3 M</td>
<td>Regen</td>
<td>Weekly</td>
<td>IBM 360</td>
</tr>
<tr>
<td>Perkin-Elmer Corporation</td>
<td>$25-74M</td>
<td>Make-to-stock</td>
<td>4 Y</td>
<td>Net</td>
<td>Daily</td>
<td>IBM 360</td>
</tr>
<tr>
<td>Twin Disc, Inc</td>
<td>$25-74M</td>
<td>Both</td>
<td>5 Y</td>
<td>Net</td>
<td>Daily</td>
<td>IBM 360</td>
</tr>
</tbody>
</table>

Source: Selected from Plossl and Wight, 1971, p.12

Entering the 1970s, the APICS organization still had a heavy regional structure. An executive office was not established until 1968. In 1965, membership was only 5,000 practitioners, but momentum began to change in the early 1970’s (see Figure 4.6.1). The group helped market Plossl and Wight’s book about the foundations of production and inventory control in 1967. In 1971, APICS sponsored Plossl and Wight to conduct a report explaining what MRP was and to provide case examples of MRP systems running on computer systems. The report carefully identified the MRP principles in relation to reorder point methodology and provided schematics of what master production schedules look like. Perhaps most importantly, it included 8 case studies of firms ranging in size and manufacturing style running MRP solutions mostly on IBM machines (see Table 4.6.1). These case studies even included sample material requirement report outputs showing the time-phased characteristics and in one case the code modification to IBM’s PICs system.

This report became the basis of information for the “MRP Crusade”, launched shortly thereafter. APICS formed an alliance with industry consultants, where consultants
would help lead seminars much like the one outlined in the 1971 report to educate production planners about the virtues of MRP planning. The consultants did not charge APICS but they could charge each participant (Lilly & Smith, 2001). APICS began to market these seminars, led by prominent consultants such as Orlicky, Plossl, and Wight. Educational videotapes were also made available through APICS.

This marketing campaign spread the good word of MRP planning techniques throughout the manufacturing community, generating significant growth in APICS membership in the 1970s and early 1980s (See Figure 4.6.1). In addition to these marketing efforts, APICS established a certification program in 1972, called Certified in Production and Inventory Management (CPIM). The intent of the certification program was to establish a minimum level of professional standards and understanding of integrated production and inventory control principles and MRP practices. Initially, there were four modules to the exam – forecasting, shop floor control, MRP, and inventory planning. By 1975, 91 people had achieved CPIM, increasing to 3459 by 1980 and over 52,000 by 1995 (Noujaim, 1997). In 1973, APICS extended its certification program to include a more advanced exam called Certified Fellow in Production and Inventory Management (CFPIM). By 1975, 50 practitioners had been certified, increasing to 916 by 1980, and to over 2,000 by 1995 (Noujaim, 1997).

In addition to these APICS certification programs, consultants began to establish benchmarks to help manufacturers assess their MRP practices. In 1977, Oliver Wight’s consulting practice began publishing an ABCD checklist that allowed companies to benchmark their manufacturing planning and control processes. This checklist would ask
Figure 4.6.2: Establishing MRP as the new dominant use

American Production and Inventory Control Society

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1957</td>
<td>McKinsey Report identifies poor management and use as source of poor returns on computer</td>
</tr>
<tr>
<td>1963</td>
<td>APICS Forme</td>
</tr>
<tr>
<td>1971</td>
<td>APICS Report on MRP usage kicks off “MRP Crusade”</td>
</tr>
<tr>
<td>1972</td>
<td>APICS Certification program started</td>
</tr>
<tr>
<td>1973</td>
<td>Oliver Wight starts “Standard” MRP check list, later establishes “ABC” list to evaluate manufacturers</td>
</tr>
<tr>
<td>1980</td>
<td>3459 APICS practitioners certified; APICS membership is 46,000</td>
</tr>
</tbody>
</table>

questions about production and planning methods and organization while providing performance levels for high performing firms (the A) through lowering performing firms (B-D). These certification programs and benchmark questionnaires helped establish standards in MRP practices and applications.

By the mid 1970s, MRP became the dominant way manufacturers used computer applications:

The APICS MRP Crusade has vanquished the infidels and brought true religion to the land. Holdouts still exist and some of the converts are Protestants, but time-phased quantity and due-date planning based on higher-level time-phased demands, forecasts, and customer orders has become widely accepted (Dorman, 1974, p. 8)
Figure 4.6.2 summarizes the events that lead to shift from the practical approach to the MRP approach to using planning software in the 1960s and 1970s, highlighting the important role APICS and consultants played in the process.

4.6.1 The Minicomputer and a new market segment

By the early 1970s, minicomputers became commercially available, expanding the market potential for manufacturing applications to smaller manufacturers. In many cases, manufacturers used minicomputers for technical purposes in engineering departments (Bresnahan & Greenstein, 1999), but they also used them to run administrative applications like MRP. In fact, ASK Group, a MRP software solution for HP minicomputers, would become one of the largest HP minicomputer reseller by the 1980s (Kurtzig, 1994). Because of their lower price point, smaller manufacturers could now afford purchasing a computer.

In terms of number of potential customers, small manufacturers represented a much larger market. Table 4.6.1.1 shows the number of establishments by number of employees over select years from 1947 to 2002. Although not a perfect proxy for firms, the trend in the number of establishments provides some indication of the changes in the demographics. From 1947 to 1977, the number of establishments grew at a 1% CAGR, with 0% CAGR since then. The growth profiles between these groups are slightly
Table 4.6.1.1: U.S. Manufacturing Establishments by number of employees, select years, 1947-2002

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5-9 employees</td>
<td>70,362</td>
<td>106,961</td>
<td>112,636</td>
<td>117,514</td>
<td>112,269</td>
<td>140,626</td>
<td>113,120</td>
<td>116,855</td>
<td>120,094</td>
<td>124,115</td>
<td>141,992</td>
<td></td>
</tr>
<tr>
<td>20-49 employees</td>
<td>40,004</td>
<td>42,802</td>
<td>48,207</td>
<td>47,266</td>
<td>45,426</td>
<td>49,802</td>
<td>51,622</td>
<td>55,779</td>
<td>55,779</td>
<td>55,779</td>
<td>51,900</td>
<td></td>
</tr>
<tr>
<td>50-99 employees</td>
<td>18,666</td>
<td>21,901</td>
<td>21,754</td>
<td>22,066</td>
<td>24,923</td>
<td>26,523</td>
<td>26,707</td>
<td>27,270</td>
<td>28,241</td>
<td>28,241</td>
<td>25,983</td>
<td></td>
</tr>
<tr>
<td>100-249 employees</td>
<td>14,319</td>
<td>15,647</td>
<td>16,732</td>
<td>17,614</td>
<td>19,762</td>
<td>20,800</td>
<td>21,546</td>
<td>21,281</td>
<td>21,963</td>
<td>21,963</td>
<td>20,346</td>
<td></td>
</tr>
<tr>
<td>250-499 employees</td>
<td>5,553</td>
<td>6,032</td>
<td>6,640</td>
<td>6,639</td>
<td>7,749</td>
<td>8,032</td>
<td>8,364</td>
<td>7,974</td>
<td>7,795</td>
<td>7,795</td>
<td>6,963</td>
<td></td>
</tr>
<tr>
<td>500-999 employees</td>
<td>2,728</td>
<td>2,836</td>
<td>2,957</td>
<td>2,942</td>
<td>3,460</td>
<td>3,481</td>
<td>3,677</td>
<td>3,323</td>
<td>3,113</td>
<td>3,279</td>
<td>2,720</td>
<td></td>
</tr>
<tr>
<td>1000-2499 employees</td>
<td>1,431</td>
<td>1,463</td>
<td>1,463</td>
<td>1,375</td>
<td>1,639</td>
<td>1,527</td>
<td>1,480</td>
<td>1,308</td>
<td>1,290</td>
<td>1,290</td>
<td>1,187</td>
<td></td>
</tr>
<tr>
<td>2500+ employees</td>
<td>504</td>
<td>533</td>
<td>533</td>
<td>544</td>
<td>674</td>
<td>582</td>
<td>581</td>
<td>482</td>
<td>421</td>
<td>342</td>
<td>316</td>
<td></td>
</tr>
</tbody>
</table>


different as well, large establishments (measured as over 500 employees), typically associated with larger firms, decreased by 1% CAGR from 1977-2002, while smaller establishments stayed relatively the same over the same time period. This growth profile corresponds with economic recessions and increased global competition as well as the more recent outsourcing trends. Not surprisingly, the smaller manufacturer market potential is much greater than the large manufacturer market (in number of firms, not necessarily revenue).

Since smaller manufacturers generally have less complex operations with less people, one may presume that they required less functionality in their software. In some sense this is true. Historically, smaller manufacturers were organized in less complex ways than large manufacturers (Scranton, 1992), but smaller manufacturers have become more complex organizationally over time. By the 1970s, small business did not necessarily have less complex production system requirements. For example, compare a small electronics manufacturer’s description of its manufacturing requirements in the late 1960s (Janesky, 1969) with Chrysler’s description (previously mentioned, (Laughna,
1960)). Both manufacturers faced variations in customer needs which increased the complexity of the production requirements. For example, Chrysler had to contend with planning different engine types, color, and trim for the same chassis. These variations greatly increased the number of active products which needed to be planned for. The small electronics manufacturer had 200 active products – 50% of which have variations yielding 3,000 different products. These products averaged 500 components which required managing over 11,000 parts in the stock room (Janesky, 1969). Similarly, the Chrysler planner estimated that production planning required 10,000 individual computations, 5,000 individual schedules, and a total 55 pages of synchronized data (Laughna, 1960). The difference in complexity between Chrysler and the small electronics manufacturer is a difference in degree not kind.

However, smaller manufacturers did operate with fewer resources than large manufacturers. Less slack in labor and financial resources meant that employees often performed multiple functions. As a result, higher level management tended to get more involved in the day to day manufacturing operations, resulting in more executive participation in computer and software purchases. From a software management perspective, horizontal functions such as data processing often were administered by a very small group or simply added responsibilities to existing personnel. In contrast, large manufacturers tended to have full departments to manage these processes. For example, one small manufacturer in 1965 described how the computer application responsibilities fell on the shoulders of the assistant office manager. These were added responsibilities, and she simply attended computer and programming training to learn how to manage the
computer (McCartney, 1965, pp. 97-8). In contrast, when GE decided to implement computer applications in its Lexington Kentucky plant, it conducted psychological testing of its systems group to determine who was qualified to program and operate these machines and hired outside consultants for help (Osborn, 1954).

The differences in size of manufacturer generated differences in how manufacturers deployed the solution. Given the resource constraints, smaller manufacturers tended to deploy the solutions in highly centralized fashion with significant involvement from upper management (Johnson, 1999). In many cases, all MRP functionality was run on one computer. Large manufacturers, in contrast, used the minicomputer in a much more decentralized way. Large manufacturers deployed MRP systems locally and did not necessarily integrate them at the corporate level. One plant’s MRP system did not necessarily integrate with another plant’s MRP system, which in some cases may be running on an entirely different hardware platform.

Therefore, when smaller manufacturers entered the market they did not introduce a significantly different functional use of planning software. The main difference was in how they deployed the software within the organization (see Table 4.6.1.2 for a comparison). This difference in deployments, however, was significant enough to create a new market segment. Many entering software providers in the 1970s purposively targeted smaller manufacturers. And, over time, how small manufacturers used planning software followed a distinctly different pattern than large manufacturers.
Table 4.6.1.2: Comparison between large and small manufacturers

<table>
<thead>
<tr>
<th></th>
<th>Large Manufacturers</th>
<th>Small Manufacturers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Understanding</td>
<td>Justification</td>
</tr>
<tr>
<td>MRP Crusade</td>
<td>“Software System”</td>
<td>Operational Efficiency</td>
</tr>
</tbody>
</table>

4.6.2 Pre-packaged software

By the late 1960’s, pre-packaged software became a possible alternative to developing applications in-house. Pre-packaged software means that the customer purchased an application in which the program logic had already been written. Lawrence Welke’s ICP Software Directory in 1967 identified 458 different programs from 150 firms, most of the administrative systems were payroll and finance (Lilly & Smith, 2001). In 1968, *Datamation* dedicated most of its October issue to packaged software with articles about acquiring software, the economics of the software market, legal issues, and user perspectives, including Ford. Significant entry into the software market industry including manufacturing software, however, did not occur until the early 1970s.

Software historians identify IBM’s unbundling decision in 1969 and the introduction of the minicomputer as important catalysts to this growth (Campbell-Kelly, 2003; Steinmueller, 1996b). No doubt these events had an important impact – unbundling software from hardware made it more economically feasible for independent software
vendors to charge a price. In an effort to keep hardware prices down, many minicomputer vendors partnered with software vendors to resell the hardware with the software already installed, called turnkey solutions. Beyond these two events, growth within the manufacturing software market also corresponded with the “MRP Crusade” in the early 1970s.

Specifically within manufacturing software, one of the first “pre-packaged” applications was IBM’s Bill of Material Processor in 1959 followed by its Production and Inventory Control Systems (PICS) in 1966, which Orlicky, Plossl, and Wight, helped develop. However, these applications were more like specifications than finished products. In 1973, IBM introduced its Communications Oriented Production and Inventory Control System (COPICS). This product was comprehensive in scope, consisting of twelve applications areas extending beyond core MRP applications: engineering and data control, customer order servicing, forecasting, master production schedule planning, inventory management, manufacturing activity planning, order release, plant monitoring and control, plant maintenance, purchasing and receiving, stores control, and cost planning and control (IBM, 1973). Within these functions, IBM emphasized:

-a high stress on the use of system for planning. Tactical and operation planning receive the bulk of the attention including such things as setting manpower levels, determining material requirements and production schedules and stating inventory policies. ... Emphasis is also given to control in that most managers feel that if they had the capability to more closely monitor their areas of responsibility, they could do a better job (IBM, 1973, p. 269).

IBM also architected the product to foster integration between these different work functions:
Most of today’s manufacturing control systems have been developed independently of the other systems with which they have to interface. Departments like Production Control, Accounting, Sales, or Engineering have typically developed their own systems autonomously, with the intent of solving their own specific problems rather than those of the entire company. The incompatibles systems that result make comparison between various areas of the business difficult, and lead to duplication of effort in systems development, programming and maintenance of company files.

The COPICS framework stresses the relationship among the various portions of the system— that is, on “systems integration”. No company will attempt to implement all of COPICS at one time, but will attack a portion at a time (IBM, 1973, p. 270).

In many respects, IBM’s COPICS products foreshadowed the enterprise resource planning (ERP) products introduced in the early 1990s. In fact, at a high level, ERP did not introduce significantly new functionality not anticipated by COPICS and both products emphasized cross-functional integration. But, IBM was correct in recognizing that most customers in the early 1970s would not implement all of COPICS at one time; as we shall see, by the 1990s this would change.

Also, as IBM pointed out in the quote, most software vendors targeting production planning and control only focused on developing MRP-type functionality. In fact, the MRP crusade and the consulting evaluation criteria provided a list of functional requirements for these programs. Many entering software firms validated their products against the principles espoused during the MRP crusade and used APICS meetings as marketing opportunities. For example, Sandra Kurtzig, the founder and CEO of ASK Computer—a software vendor that targeted smaller manufacturers—explained:

... I attended a talk given by a guy named George Plossl, who’d written a book called Manufacturing Control: The Last Frontier for Profits. An avuncular man in his late fifties, Plossl argued that manufacturing could be
made into a science through the computerized material requirements planning (MRP). Much of the manufacturing "science" he referred to in his talk I had already incorporated into the Farinon program. It was a pleasant surprise to learn that guided by only common sense and the needs of my customers, I'd unwittingly stumbled onto manufacturing's leading edge (Kurtzig, 1994, p. 60).

Indeed, most firms that entered the market designed their pre-packaged solutions around the MRP principles.

It is also interesting to note the different kinds of firms that entered the manufacturing software packaged markets. As mentioned, many hardware vendors, such as IBM and Honeywell, entered the market prior to the 1970s. Some minicomputer vendors like Hewlett-Packard also sold their own systems, but many just relied on turnkey solutions in which independent software vendors resold the minicomputers with their software. Another logical entrant was the consultants who worked on many different implementation projects building manufacturing expertise. Two firms who built significant practices in manufacturing were Arthur Andersen (now Accenture) with its MAC/PAC product and Rath & Strong, with its Production and Inventory Optimization System (PIOS). Less obvious entrants were service bureaus, such as ADP who developed services specifically for manufacturing, and customers themselves who wanted to recover some of the development investment costs. Examples of customers include Martin Marietta and Boeing. The independent vendors can be divided into those that entered with a focus on manufacturing applications, e.g., ASK Group, JD Edwards, versus more diversified firms from other functional areas such as SAP, MSA or McCormick & Dodge or from the systems level market such as Cincom, Cullinet, and Oracle. Although the
independent software vendors made up the majority of the market, many of these other forms persisted well into the 1990s. For example, Arthur Andersen and IBM did not exit until the mid 1990s.

In fact, IBM soon gained considerable market share based in part with its early involvement in developing manufacturing software and the strong presence of its hardware in manufacturers. By 1984, the industry analyst group IDC estimated that IBM had 50% market share. Since then no other software vendor has achieved that much market share; current market leader SAP was estimated to have approximately 39% market share as of 2003. Although the top several market leaders historically have had over 50% of the market, the overall market is not very concentrated. Historically, the top ten firms have captured about 60-70% of the market, but currently, the market does appear to be consolidating especially among those firms who target smaller manufacturers.

During its formation, the market could be segmented in several different ways. Since the initial packaged software depended upon proprietary hardware and operating systems, vendors concentrated on developing applications for a particular hardware platform. Early on many firms concentrated on IBM and as minicomputers became more popular others wrote to DEC or HP. Target manufacturing size also distinguished between competitors. Those who targeted large manufacturers also tended to have solutions for smaller manufacturers, e.g. IBM’s COPICS versus MAPICS solution. Those who targeted smaller manufacturers tended not to also target larger firms but many would sell into divisions of large firms. For example, some of ASK Computer’s early customers were large firms like Boeing but they focused on smaller firms in general. It is not clear if firms
initially targeted different manufacturing styles, but eventually in the small manufacturer market vendors usually focused on specific and related manufacturing styles. For example, ProfitKey targeted make-to-order manufacturers; whereas, Datalogix targeted process manufacturers. Vendors who target larger firms tend to provide solutions to multiple kinds of manufacturing.

From the customer perspective, the first hurdle for a manufacturer interested in considering packaged software was obtaining information about who sells these programs. Initially, some hardware vendors such as IBM and industry associations had lists of software vendors (Glaser, 1967). After 1967, they could consult Larwence Welke’s ICP software directory, essentially a yellow pages listing of software products and companies with brief descriptions and contact information (Haigh, 2004). In 1973, Oliver Wight’s consulting practice went one step further by providing evaluation of MRP software package’s completeness with respect to standard MRP practices at the time. Another important avenue of information was word of mouth among manufacturers. The purchase process usually involved talking with, even visiting references. APICS meetings were also a good source of information and in the 1970s software firms became active participants in presentations. Later, software firms would set up demonstration and informational booths at expositions held in conjunction with APICS meetings and at other conferences.

Once aware of this alternative, manufacturers then went through a lengthy process of picking a software solution. In 1973, Sundstrand, a large industrial manufacturer purchased a software package for two plants in its Aviation Division. One participant in this process, George Hoyt, summarized their experiences in the 1975 APICS Conference
Proceedings. Sundstrand typified the evaluation process at large firms ((Hoyt, 1975), see also (Glaser, 1967; Head & Linick, 1968; Noling, 1971; Pantages, 1968) for similar arguments). Before any software packages were evaluated, Sundstrand considered whether they should make or buy the application. At least initially, purchasing software packages was framed in comparison to custom developing an application. Similar to how GE justified the purchase of the computer by savings relative to existing data processing cost, Sundstrand analyzed the packaged software benefits relative to developing the application themselves. These benefits included shorter installation, less development cost, re-allocation of over burdened data processing personnel, better utilization of computer resources, and even tax advantages (because purchased software could be capitalized but development cost must be recorded as they occur – a feature software vendors would later use as reason why firms should buy new software instead of fixing Y2K issues themselves). There was the additional belief that software vendors utilized the most advanced technologies and techniques and employed better programmers. Some were concerned about potential problems from the data processing group who was accustomed to getting software for free and had a desire to build these systems themselves (Head & Linick, 1968).

And, Sundstrand anticipated using pre-packaged software in similar ways as a custom developed application – to automate existing business functions. Evaluating software packages in many cases replicated the process of building an application from scratch. As any good system man of the time would tell you, to develop an application required specifying the functional requirements first. This entailed talking with users and
analyzing the current procedures. Minor changes and improvements typically were tolerated as these requirements were converted into design specs. Programmers would then convert these design specs into actual application code, which was tested and then presented to the user. Similarly, it was believed that in order to properly evaluate software, the manufacturer must specify in advance the detailed functional requirements of the system. System analysts (as they were beginning to be called) and eventually consultants were used to conduct an analysis of the functions to be automated by the software packages often producing a Request for Proposal (RFP)\textsuperscript{14} with the detailed requirements. This analysis usually involved interviewing or documenting what actual potential users did, getting direct involvement of users in the buying process.

Manufacturers would send this RFP to potential vendors. Beyond functional requirements, firms typically asked for technical specifications, references, product release schedules, and financial information. Since application vendors developed solutions specific to proprietary hardware, the selection of packaged software was still closely tied with hardware preference. Based on responses the to RFP, manufacturers would short list the number of vendors. The short listed vendors typically were invited to provide a demonstration of their software before final selection was made (for an example of this process see the walk through of an MRP implementation at Oster in the early 1980s - (Savage & Schuette, 1982)). The ultimate decision usually fell upon some combination of data processing, the controller, and perhaps some operational representative. In a 1985

\textsuperscript{14} I have been unable to determine whether the concept of RFPs originated with software purchasing. Discussions of buying computers do not include such specifications, but it is possible that firms used RFPs for other types of purchases before software.
survey of manufacturers, industry analyst group IDC found that the information technology organization was responsible for the purchase and use of the manufacturing systems in 55.9% of the firms (IDC 2665). Thus, it was commonly believed that the most rational way to decide between software vendors was to understand in advance what exactly the software was supposed to automate. This belief persisted well into the early 1990s.

Because packaged software was already programmed and inevitably no software could perfectly match functional requirements, manufacturers faced a new consideration – whether to conform to the software specification or to customize the software. The advantage of conforming functionality to a software package was that it reduced implementation time, risk, as well as simplified conversions to new versions of the software. But, changing existing processes to conform to a general process was the main disadvantage of conforming. Conversely, customization increased implementation risk, made upgrades more difficult, and was difficult to contain, but it preserved the business processes of the manufacturer. Sundstrand did extensive analysis trying to determine the optimal trade-off between customization and implementation cost and return on investment, ultimately deciding that limited customization was best (Hoyt, 1975).

The preference not to change the software persisted well into the 1980s as reflected in responses to APICS surveys (Hobbs, 1985). For instance, at Oster, they estimated that only 25-30% of processes document required redesign and cautioned against changes:

The main reason for using a standard package without changes is to avoid the traditional problem of data processing becoming the “eye of the needle” on the project schedule … There is also the problem of the vendor supporting on-going maintenance and
future enhancements, if the package has been modified. (Savage & Schuette, 1982)

Packaged software was meant to control existing functions and not to redesign these processes. The rational behind limited redesign was the cost of change. Note how Oster expressed this cost only in technical terms – the increased dependency on IT and concern about support from the vendor. Much like GE’s initial analysis of computers in 1954, Oster only evaluated organizational change in terms of the programming change cost and did not consider the potential business benefits of changing a business process.

After purchasing, the next phase focused on getting the software to work within the organization. Manufacturers began to call this process the “implementation” in contrast with the previous “development” of applications. The implementation process involved a series of technical and data conversion exercises, education of the users about the systems, and conversion to the new system. In many cases, the MRP techniques were not well understood within the firm, requiring manufacturers to send a group of management to educational workshops offered by consultants. Larger firms would bring these consultants on-site. In fact, Oliver Wight formed what he called “The Proven Path” – a standard methodology to implement MRP systems. Larger firms would also hire outside consultants to help plan and implement software. Firms such as Andersen Consulting, Ernst & Young, EDS started to build significant implementation businesses.

The two most underestimated areas were preparing the data for the new application and getting the users to change work habits to conform to the system. Packaged software has its own requirements which required data to be in a certain format and structure. To
prepare the data required a significant amount of effort cleaning existing data and converting it to the new format. To the extent that any data would be exchanged with other applications that were either developed internally or provided by another vendor, e.g., a payroll application, an interface would have to be built to help convert the data into the proper structure. Consultants often would spend significant amount of their time building interfaces. On the people side of the equation, users did not get the proper training and documentation after the implementation of the system by the technical personnel. This usually made for a difficult transition. It was soon learned that getting operational people involved earlier in the process helped increase use and decrease resistance among the user community.

From a deployment perspective, packaged software tended to be implemented at the functional and department levels. Several factors contributed to this. In MRP’s case, even though it was integrative, it was only so within manufacturing as Dearden originally recognized in his critique of the total systems view. Initially, MRP integrated the primary functions around planning and controlling the material flow within a production process with limited interfaces with accounting and purchasing information. Prescolite, a division of U.S. Industries, implemented IBM’s bill of material processor (BOMP) and inventory requirements planning (IRP) software in 1971. Their use exemplified the business-based justification and implementation processes of the early adoption of MRP applications (Stevens, 1973). The company justified the switch from re-order point to MRP based upon potential business cost improvement: its impact on inventory investment, reduction of expediting and late-orders, and improved service level. They chose IBM based upon price
performance, functionality, and availability of IBM engineers. The firm made the decision to decentralize the deployment at the plant level in order to further engage manufacturing personnel to use the system. Implementation involved a significant systems design phase and outside help from consultants and recognized the difficulty of obtaining data accuracy.

In summary, acquiring a packaged solution represented a new kind of product purchase for large manufacturers accustomed to developing applications. Manufacturers believed that the best way to purchase software was to understand in advance exactly what the software was supposed to automate. Because no pre-packaged software could perfectly match functional, organizational, and technical requirements, manufacturers faced the new consideration of how much customization of the software should be done. Nevertheless, manufacturers essentially used pre-packaged software in similar ways as the internally developed applications. Their understanding, justification, and use of a pre-packaged software application mirrored the developed applications.

4.7 1980’s: Variations of a dominant use

In the later part of the 1970s and 1980s, there were important variations in how manufacturers used MRP software. Manufacturers started to expand MRP functionally, incorporate new manufacturing techniques, and modify MRP to support different styles of manufacturing. Like the 1960’s, the main source of these changes came from the experience of using MRP applications. Exposure to alternative manufacturing techniques, in particular the Japanese just in time (JIT) system, also encouraged manufacturers to re-
examine the MRP approach. Table 4.7.1 counts the total number of presentations at APICS that focused on these topics, showing the increased interest in these issues as manufacturers became more experienced with MRP.

Similar to the early computer in the 1960s, the results of MRP installations were getting mixed reviews in the 1980s. One frequently cited survey conducted by two academics, John Anderson and Roger Schroeder, in early 1980 showed that less than 10% of companies surveyed achieved full benefit from the MRP systems, while 30% got some good benefits, and over 50% obtained little or none (Anderson & Schroeder, 1984). The Anderson and Schroeder study blamed management for the poor return on investment, just as the McKinsey study had in 1963. In particular, the MRP study pointed out that many manufacturers did not use the application functionality to its fullest potential (Anderson & Schroeder, 1984). This review suggested that more could be learned and done.

Oliver Wight emphasized a similar point in his 1981 book *MRP II: Unlocking America’s Productivity Potential*. He argued that MRP applications can help manufacturers learn more about production problems and how to more effectively manage and control them. In his book, Wight identified two particular areas of learning—scheduling material requirements and integrating with other functions within the organization (Wight, 1981). Recall that one of the new features of computerized MRP applications was the computationally intensive logic that combined production schedules with material requirements to schedule when material was needed. Wight explained that as manufacturers used these systems, they began to learn more about scheduling. In particular, manufacturers learned that to make the MRP scheduling logic more valid it
needed to consider capacity constraints and to get information feedback from production activities. Original MRP logic assumed standard lead times and capacity (Orlicky, 1975), but often these assumptions were not valid and created infeasible plans. As a result, capacity planning was added as an additional step in the production process. For example, Prescolite identified capacity planning as an additional area to add functionality to its MRP implementation (Stevens, 1973). In addition, to improve planning, manufacturers learned the value of providing feedback from the execution of the material schedules into the planning process. Because of the feedback, this approach was called “closed loop” MRP planning. Wight credited a small manufacturer Markem as the first to implement this type of system in 1969 (Wight, 1981, p. 53).

Wight also believed that the information within the MRP system had value to other functions, in particular, finance, marketing, and engineering. As had been previously discussed, there has been a long history of interfaces between finance and manufacturing. Finance required information from manufacturing to prepare costing, budgeting, and financial statements. Historically, accounting had viewed manufacturing data as too
inaccurate for this financial analysis, but the implementation of MRP solutions had improved data accuracy and made this interface more apparent at the functional level. Yet, it was not always entirely clear who obtained the rights to this data and who should develop the systems to manage it (Nolan & Knutsen, 1974). Proponents of MRP systems believed that the production schedules provided these numbers to run the business and advocated broader business planning capabilities that integrated finance, marketing, purchasing, and distribution. They called this system manufacturing resource planning, or MRP II, which essentially was a closed loop MRP system with this additional business planning functionality. Wight explained MRP II as follows:

Manufacturing Resource Planning (MRP II) is a game plan for planning and monitoring all of the resources of a manufacturing company; manufacturing, marketing, finance, and engineering. Technically, it involves the closed loop to generate the finance figures... The requirements are also costed out, schedule receipts and planned order ... are costed out, and a projected component inventory balance is calculate. ... Manufactured items can be broken out of the plan, and converted to capacity requirements in standard hours by work center. These can be costed out to show the amount of labor as well as the amount of material that needs to be purchased to meet a given path. That was the first step in developing MRP II – Instead of having one set of numbers for operating system in manufacturing and another set kept by the financial people, once manufacturing people have numbers that are valid, the financial people can use these to get their numbers (Wight, 1982, pp.6-7)

This level of integration went further than the data interfaces discussed at GE in the 1950s. MRP II promoted integrated planning between the two groups, suggesting they operate off one set of numbers. Incidentally, these numbers came from the manufacturing function, not surprising given that Wight was originally a planner himself. Plossl even argued that accounting system needed to change their cost systems because operating systems have
changed with MRP systems. The tension between whose system should be the basis of planning – finance or manufacturing – continued from the initial uses of planning software (Plossl, 1987).

In promoting MRP II, Wight published a companion book targeted for executives, called *The Executives Guide to MRP II* (Wight, 1982). This book lacked the technical details but explained the MRP II concept within a broader business framework. Appealing to the executive audience identified a new potential user within the firm. Although executives were certainly involved in purchase decisions, they were active more at the general approval level. By expanding the scope of MRP to relate to other functional areas, these systems potentially could become of more interest to executives operationally. Wight justified these systems along the now standardized business cost minimization approach. The benefit of an MRP II system was in its ability to meet production schedules at the lowest possible costs.

In addition to extending the core functionality of MRP, manufacturers also tried to apply MRP systems to industries for which the system was not originally designed. Orlicky recognized that MRP was essentially designed for discrete manufacturing that operate in a make-to-order and make-to-stock production environments (See Appendix 4A for distinction (Orlicky, 1975)). Most of the early adopters fit this mode, but by the late 1970s there was more experimentation in other industries – in particular, process industries and aerospace & defense. In applying MRP to these new industries, manufacturers learned about potential deficiencies in the MRP solution. The main issue with aerospace/defense was that the production process, although discrete in nature, was run as projects. MRP
systems originally were designed to operate off of customer orders or forecasts and did not have strong project management capabilities. Process manufacturing issues included the continuous nature of the production process, having highly variable units of measures, dealing with variability of the components and by-products as the product is produced, and managing recipes instead of assembled parts. Thinking about vertical expansion further made differences between manufacturing practices more apparent, solidifying the belief that MRP addressed only certain manufacturing segments. Interestingly, many software vendors expanded their product offerings to address aerospace/defense’s unique requirements, in particular, adding project management capabilities. For example, Andersen Consulting designed a system specifically for this industry. In contrast, the process industry given its relative size remained underserved. Some firms like Datalogix did develop process industry specific solutions and SAP did enter the U.S. market by initially selling to process manufacturers.

Outside of these learning experiences, U.S. manufacturers became increasingly aware and experimented with alternative production management techniques. One such techniques was Computer Integrated Manufacturing (CIM), which had its roots in the 1970s (see Harrington, 1973). Like MRP II, this approach promoted increase integration. In practice CIM advocated vertical functional integration within the manufacturing process as opposed to the horizontal business planning integration found in MRP II. Vertical integration entailed connecting the administrative planning function (such as MRP) with the automation of the shop floor machines and engineering. In particular, CIM called for tighter interfaces between the engineering software, CAD/CAM drawings, and the product
information in the MRP systems. Think of a CAD/CAM drawing feeding into a bill of material file.

In the 1980s, several large manufacturers, such as General Motors and John Deere, attempted to develop the protocols and interfaces between the manufacturing equipment, data collection equipment, engineering computers, shop floor computers, and eventually the administrative computers (as chronicled in *Manufacturing Systems* magazine). Such integration took closed loop to a new level as data collection about materials and equipment were supposed to be fully automated. In contrast with the MRP II approach, CIM emphasized execution over planning. However, the lights out factory never really materialized on a large scale.

Perhaps more significant was U.S. manufacturing’s increased exposure to the Japanese Just-in-time (JIT) model. With Japan’s success in the automotive and electronics industries, U.S. manufacturers increasingly became aware of this technique. Developed in Japan and applied in practice in the early 1970s at Toyota, JIT provided an alternative view on production and inventory control than the MRP logic developed in the United States (Cusumano, 1989). Although both approaches try to eliminate waste and improve customer value, JIT favors flexibility and continuous change based on a pull method of inventory control called Kanban and MRP focuses on developing a realistic production plan which pushes inventory through a production system. For MRP oriented solutions, the manufacturing problem was framed as how to minimize cost by most efficiently meeting production schedules. This meant generating effective production and material schedules as well as responding to changes in production. In contrast, JIT framed
the production problem in terms of maximizing customer value with the idea that customers could get good quality even while the manufacturers minimized its cost.

Essential to JIT is the concept of continuous improvement – business functions and processes could and should change when necessary. Under JIT, each employee had local control and responsibility toward these goals leading to decentralized decision making; however, the logic of MRP II centralized decision thinking through one set of numbers and books.

In 1985, APICS commissioned a study in which practitioners traveled to Japan to learn and evaluate Japanese JIT production techniques. Although today the distinctions between JIT and MRP are quite clear, at the time manufacturers did not make such sharp distinctions when they first encountered JIT. One common interpretation viewed JIT as a broad management philosophy and Kanban and MRP as two different methods to achieve it (see for example, (Andreas, 1987; Brooks, 1985, 1987; Fuller & Brown, 1987; Raeker, 1987). Often, Kanban and MRP were viewed in direct competition. Kanban is an inventory control technique that pulled inventory through the production process rather than pushed it through with MRP material schedules. In some respects, Kanban resembled the re-order point systems MRP was meant to replace. Some, however, argued that they can and should co-exist (Andreas, 1987). Many Kanban systems did not require computerization. JIT, in some respects, was a threat to the computer and software industry. It responded by highlighting differences and arguing that both were needed – MRP provided a long-term plan whereas Kanban did not ((Raeker, 1987), who worked for an MRP software firm). Regardless of the compatibility of Kanban and MRP, JIT was
viewed as a more general management approach that provided a rational logic to run a manufacturing business. Thus, it was believed that MRP could be compatible with JIT if certain changes were made. The idea was that firms were already committed to their MRP systems, but JIT identified potential improvements to several areas in the execution level. (Bourke & Sandras, 1985; Raeker, 1987).

Aside from these changes in manufacturing practices, this period experienced significant technological changes in each area of the technology stack—hardware, communications, and software. Most of the technological changes promoted integration across the heterogeneous technical environment. These technologies were called “open” to contrast with the proprietary technology since the introduction of the mainframe.

In operating systems, there was a push toward providing a system to could operate across heterogeneous hardware environments. The operating system UNIX replaced the need for hardware vendors to develop a proprietary operating system. UNIX was developed on a DEC machine at Bell Laboratories dating back to 1971, became commercially viable in the early 1980s. Also during the 1980s, IBM was developing a standard, called System Application Architecture (SAA) and released in 1988, to enable interoperability on different platforms and operating systems. Unlike UNIX, this standard applied to different platforms and operating systems within the IBM product family. Although UNIX became the dominant interoperable operating system, the SAA architecture impacted the many software vendors who wrote to the IBM platform.

During the 1980s, standards to facilitate communication and data transmission between different computers also became more commercially viable. TCP/IP, a series of
protocols to manage data transfer, is perhaps best known as the backbone of the Internet, started its shift from defense and government use to commercial use in the 1980s. In 1985, the Internet Architecture Board held a workshop on TCP/IP for the computer industry, to help popularize the protocol for commercial use. Other protocols such as IBM’s System Network Architecture (SNA) were introduced during this time period. These protocols focused primarily on communications across wide areas, but equally important was local communication among computers, called local area networks or LANs. LAN technology was introduced in the late 1970s and enable several machines to be integrated together to take advantage of a network computer or even multi-user processing. In a certain sense, LANS supported some of the first distributed processing among computers. IDC estimated that in 1985, less than 6% of PC's communicated over LAN, but by 1990 it grew to 35% (IDC 5204).

At the database management layer, relational database management software became commercially available in the 1980s. Until that time, most applications used a hierarchical data management structure. Recall that one of the limitations of hierarchical database systems found in most host-based systems was that it was difficult to integrate different indexes. This data management structure made it difficult to relate information between programs, but relational data management systems support relationships between data by supporting indexing at the data level at each table. In addition, relational databases supported a standard querying language called SQL which facilitated data extraction and manipulation. Because these data management systems supported relationships between the data and provided a common access language, this more robust data management
system enabled firms to centralize their data management while preserve local control of using the data. The value of this centralization at the schema level is that it helps maintain data integrity.

Ironically, perhaps the most significant technological event in the 1980s the microcomputer, or PC (introduced to businesses in 1982), did not integrate technologies together, but promoted further decentralization. The microcomputer put computing power right on a manager's desk. In fact, many microcomputer manufacturers targeted individuals directly because the cost was low enough not to require substantial budgetary approval (Chandler, 1997). The first applications developed for the PC were personnel productivity solutions such as document editing and spreadsheets. The low cost of the PC also further expanded the manufacturing software market by enabling even smaller manufacturers who could not afford, nor need a minicomputer. Another important hardware development was IBM's release of the AS/400 in 1988. In a certain respect, the AS/400 contrasted with the open minicomputers because the AS/400 was a fully integrated box with proprietary operating system and data management software. Despite the fact that it was proprietary, it was very reliable and IBM aggressively partnered with a wide range of software vendors. It quickly became one of the most successful mid-range computers in the market. By 1994, IBM had sold close to 300,000 units (Hill, 1996).

The introduction of the PCs further promoted decentralization while also increasing concerns about data integrity. PCs enabled business users to do their own data analysis without fully depending upon the information technology staff. In the mid 1980s manufacturers started using spreadsheet for manufacturing purposes (see (Johnson, 1985)),

-134-
and spreadsheet provider Lotus even advertised special manufacturing plug ins for its product. For example, production planners could download production information into a spreadsheet and do their own contingency planning outside of the systems. However, data manipulation outside of the application threatened the data integrity of the system. If the planner did not upload his altered plan, the existing MRP II application would have the wrong information. Since the IT groups still had responsibility for data integrity of the systems, this localized use created concern.

In summary, the variations in use during the 1980s were of a different kind than in the 1960’s. The introduction of MRP in the 1960’s represented a new understanding, justification, and deployment for software applications; whereas, the variations in the late 1970s and 1980s extended the current use of MRP software while maintaining the same use. In the 1980’s, manufacturers continued to justify software purchases based upon potential cost savings and to view software in the sense that it automated business functions. They also used the same overall selection process of identifying the functionality upfront and then selecting a software system based on best matching that functionality. For large manufacturers, the deployment of the software also tended to be decentralized. In essence, these variations made incremental changes to the same use as opposed to proposing a new use. In Rosenberg’s language, the learning by using that occurred during the 1980’s principally generated embodied knowledge where in the 1960s it lead to disembodied knowledge.

However, there were some important subtle differences in the process of buying software. As the functionality expanded with business planning integration from MRP II,
a broader set of managers got involved in the software selection and implementation process (Funk, 1986). In addition, it was recommended that management training about MRP practices and principles occur earlier in the process so they will be better informed during the selection process (Gray, 1985). Consultants were also getting involved earlier in the process. Historically, consultants were used simply as advisors during the selection process and got more involved during the implementation phase. But, by the early 1980s consulting firms started offering a new service called “systems integration”. This was a rapidly growing business and by mid 1980s IDC estimated that the system integration business was $9.8 B with large professional service firms and the Big 8 accounting firms capturing the most market share (IDC, 1987b). This service helped clients more effectively combine the different hardware, software, and communications to better support the information needs of their client. Consultants got engaged in the purchase decision to ensure that the software would support the developing architectural standard they were helping the manufacturers build.

These changes in how purchasing was done would prove to have significant effects on how the different customer segments interpreted and reacted to the technological transition to distributed computing.
4.8 1990’s: Client/Server and a new dominant use, ERP

All the technology, service, and manufacturing practice variations that emerged by the early 1980s culminated with a significant technological change in manufacturing software architecture by the late 1980s. The new architecture, called client/server, took advantage of the integrative technologies, such as interoperable operating systems, relational databases, and communication protocols, to distribute transactional processing across multiple hardware platforms. It was believed that this architecture would be more cost effective because it enabled more processing across cheaper machines. For the software vendors this technological change represented a significant and disruptive transition. It required learning new technological skills and changing the product’s architecture. By the late 1980s, several new client/server applications entered the market and by the early 1990s, most pre-existing software firms recognized the need to change and were investing heavily to make it. In general, software vendors perceived this transition as technical and did not anticipate any changes in how customers would use the software.

Various consultants, industry analyst groups, and other invested market intermediators also influenced this transition. Industry analyst groups, such as Gartner Group, IDC, and AMR, asserted their presence by explaining the significance and issues associated with this technological change as well as by assisting manufacturers in negotiating software purchases. Gartner Group even coined the term ERP (Enterprise Resource Planning) which would replace the MRP II systems manufacturers had been
implementing. In consulting, client/server provided a standard architectural platform for system integrators to recommend and implement at customer sites. From the management perspective, consultants offered business process reengineering (BPR) services, encouraging firms to radically change their business processes to become more cross-functionally integrated and to better service customers. The reengineering charge at the business practice level found a receptive audience in large manufacturers because of the increased competition and an economic downturn in the early 1990s. Consultants, in general, continued the trend of becoming more inserted in the application decision process as opposed to the historical focus on implementation support.

From the manufacturers’ perspective, the transition to client/server was a mix of reusing existing technologies and developing new uses for others. At the hardware level, manufacturers often did not simply replace existing hardware such as mainframes with new workstation technology, but leveraged existing hardware in the new client/server architecture (Bresnahan & Greenstein, 1996; Bresnahan & Saloner, 1997). However, on the software side, manufacturers were much more willing to implement completely new applications. Reengineering principles argued for complete functional integration beyond just the planning layer implemented by MRP II systems. As a result, existing software systems did not fully support this new type of processing. Often larger firms would turn to the newly named ERP applications to configure these new business processes. Ironically, the functionality coverage of these new systems was not necessarily any broader than the pre-existing systems.
In addition to the influence of reengineering, large manufacturers changed how they purchased software, expanding from a cost based to value based justification. Because reengineering required changing business processes, often software was selected on a more general idea of support of new business processes rather than the historical approach of matching software functionality to a detailed list of functional requirements. Lastly, these large manufacturers did not use this software to automate existing systems, but they were solutions to help configure new business processes to achieve sustainable competitive advantage.

Smaller firms did not change their usage patterns of planning software as dramatically as large manufacturers. Small manufacturers were attracted to client/server because of the supposed lower cost of ownership. However, because their deployments of MRP software were more centralized than larger manufacturer’s installations, smaller manufacturer’s core processes already had some level of integration. Thus, the call to leverage software to reengineer the business did not make much sense, especially given the cost (AMR, 1995a). The proposed new use that was radical for large manufacturers was much less so for smaller manufacturers. Consequently, they did not change their use of software as they adopted the new client/server solutions.

Figure 4.8.1 summarizes how each customer segment understood, justified, and deployed client/server solutions in relation to the previous period. It shows how large manufacturers made significant changes in each of these dimensions of customer use, but small manufacturers did not. To better understand the details of these differences requires further explanation of changes in the technology and the role of the market intermediators.
4.8.1 The technological discontinuity: From host-based to client/Server architectures

By the late 1980s, the technical changes at the component levels of the software stack (see Appendix 4B) culminated in a substantial architectural change within manufacturing software. Since the first applications released by IBM in the late 1950s and early 1960s, software applications had been optimized to run on a particular proprietary operating environment. As discussed, these systems were called, “host-based”. A host-based system processes all the transactional and data management on a host machine, originally a mainframe and later on minicomputers or microcomputers. Since each host hardware vendor had its own operating system and infrastructure software, applications were written to specific hardware platforms. Originally, business and IT professionals interacted with these machines through “dumb terminals”, which did not process...
transactions. As mentioned, with the introduction of the PC, several firms began experimenting having these machines manage a different set of transactions that would then interface with a different machine.

In contrast, client/server architectures distribute transaction processing (see Figure 4.8.1.1). A distributed architecture, in contrast, splits the transaction processing of an application between different machines, enabling concurrent processing, multitasking, and better utilization of hardware resources. For example, consider the order entry process which requires writing the customer order, checking inventory availability, and placing the manufacturing order. In a host-based system, all the transaction processing would occur on one machine which would batch this transaction with other customer orders and process them all at a prescribed time. In a distributed environment, logically different machines
(physically it could be the same) would share the processing of the order in real time. One machine may focus on data manipulation whereas another would manage the process logic and presentation management. Typically, client/server systems operate on separate machines connected through a network. The client, usually a microcomputer, sends requests to the server which processes the requests and replies back to the client. In client/server architecture the client is not “dumb” but may do some data verification. For instance, the logic that prevents a production planner to put a certain inventory value for a given component part may actually reside in the client.

Even though the transition from host based systems to client/server was well recognized, the change was significant and difficult. As the name implies, this technological transition represented what the technology management literature calls an architecture innovation (Henderson & Clark, 1990). It required learning new operating and database systems as well as reworking how business transactions were processed. The transition to client/server introduced new components such as UNIX operating systems, relational databases, graphical user interfaces, and new communication protocols. For example, by splitting the transaction processing between machines, software engineers had to change how the application processes transactions and build additional safeguards against interruptions. In fact, the relationship between these technical components changed as well. In the host-based design, the hardware component was the core of the technical system as it controlled many of the other infrastructure layers. Each enterprise application was written specifically for that platform. In the client/server design, the enterprise application is somewhat hardware agnostic as it runs across multiple
machines. This agnosticism shifts the core of the system from hardware to the infrastructure layer. Lastly, the changes in components and the relationship between these components required substantial rewriting of program business logic of the enterprise application. One software vendor, SAP, estimated that they spent over a billion dollars making this technical transition (Plattner, Scheer, Wendt, & Morrow, 2000).

Eventually, the new client/server solutions came to be known as enterprise resource planning (ERP). That said, it is interesting to point out that the planning functionality within the ERP system was essentially the same as the MRP systems. In fact, many software vendors initially thought of this transition simply as a costly technological transition – porting the functionality from a host-based architecture to client/server. Early versions of the new client/server products typically had less functionality than the older host-based solutions (Plattner et al., 2000). From a functional perspective, ultimately, the main difference between ERP and MRP II applications was in the breadth of the functional integration. MRP II essentially was vertically integrated within the production function with some cross-department interfaces at the planning level. ERP, on the other hand, was integrated across departments at both the transaction and planning levels. For example, when the production plan identified a need to order more inventories, a purchase order could be generated, and once completed the general ledger would be updated appropriately. Prior to ERP, each of these steps would be handled by different systems.
4.8.2 Changes in Intermediators: Industry Analysts, Consultants, and BPR

In addition to the technical changes, there were significant changes in the intermediators and the roles they played in mediating the relationship between customers and producers. Where APICS played a significant role in establishing MRP as the dominant use of planning software in the 1970s and 1980s, industry analyst groups and consultants played more of a role in the transition to client/server computing. In interpreting this change to distributed computing, intermediators collectively focused on cross-functional integration. In general, their concept of integration expanded beyond the planning level integration found in MRP II to include integration across business functions at the transaction level. A common phrase uttered during this period was “knocking down the information silos” between functions. Each intermediary group had their own spin on what this integration was, its significance, and what manufacturers should focus on.

Industry analyst groups focused on technical solutions; whereas, consultants concentrated on the process reengineering the business.

Industry analyst groups played an important role in identifying technological trends and customer usage patterns during this transition. One of the first industry analyst groups, IDC, was founded in 1964 to provide analysis of trends, buying patterns, market size for a variety of technologies. Clients could use these reports to help make more informed technology decisions. By the early 1980s, IDC was conducting specific analysis on the manufacturing software industry. Another important analyst group in technology, Gartner, was founded in 1979. Along with providing sector and trend reports, Gartner also
provided direct advisory services for clients. AMR, which focuses specifically on the manufacturing market and in particular software issues, was founded in 1986. Collectively, industry analysis were interpreting what client/server was technically, how firms were using this new technology, how software vendors were integrating this new technology, and potential new visions for the software.

Like any other market, industry analysis is competitive. These firms compete for client services by marketing their knowledge of technological trends. One way to demonstrate this knowledge is to coin terms that help describe the trend. Both AMR and Gartner provided acronyms to describe the potential change in client/server software applications. In 1991, Gartner issued a report developing the notion of enterprise resource planning (ERP) which promoted cross functional integration by leveraging client/server technology. Gartner even advertised this concept in manufacturing trade magazines (see add in October 1992 issue of Manufacturing Systems). Around this time, AMR also coined the term Customer Oriented Manufacturing Management System COMMS. This concept also promoted the idea of integrating functions around improving customer service – an important theme of BPR. For whatever reason, the ERP acronym took hold and the next generation software became known as ERP software.

It is important to recognize that the name ERP came from market intermediators. As a name, ERP departed from the MRP and MRP II naming convention. When Wight named the next generation MRP solution MRP II, he wanted to emphasize the importance of manufacturing in the planning process (Wight, 1982). In contrast, the industry groups
consciously wanted to move away from the manufacturing centric naming convention.

ERP was meant to invoke a radical break from previous systems.

APICS was still very active during this time period, but these other groups inserted themselves as key advisors on technical and managerial issues related to the technological transition. In addition to identifying trends, AMR and Gartner got involved in the manufacturer's purchase decision by providing advice on pricing and negotiation tactics. As a result, the industry analyst groups became an important constituent not only for information about technological trends, but also for directly influencing the purchasing decision. In 1991, APICS initiated another certification program called Certified in Integrated Resource Management (CIRM). CIRM initially included five modules – customer and products, logistics, manufacturing processes, support functions, integrated enterprise management. This educational program certified a broader range of functions than the production and inventory control focus of CPIM. Integrated Enterprise Management also became an important topic at APICS proceedings. In 1992, it had its own topic heading with over 25 presentations.

Consultants also experienced significant growth during this time period. The systems integration businesses that started in the 1980s continued to flourish as consulting groups continued to added personnel. IBM entered the service market in a significant way with its new service oriented strategy developed in the late 1980s and early 1990s. From a service offering, BPR was introduced in the early 1990s, popularized by Michael Hammer’s famous Harvard Business Review article, “Don’t Automate, Obliterate” (Hammer, 1990) and his subsequent best selling book with John Champy in 1993.
(Hammer & Champy, 1993). At the same time, other consultants like Tom Davenport were publishing articles about reengineering and process management (Davenport & Short, 1990).

In some respects, BPR integrated traditional management consulting concerns about strategic issues with traditional technology consulting concerns about developing and managing a robust technical infrastructure. An important facet of the reengineering message connected the investment in technology with process change. Hammer argued to “Treat geographically dispersed resources as though they were centralized” and “Capture Information once and at the source”, but he also recognized to “Have those who use the output of the process perform the process” (Hammer, 1990, pp. 109, 110, 111)). The concept of cross functional integration in ERP paralleled the consultant’s desire for BPR. However, the connection between ERP and BPR was not explicit until later. By 1996, Michael Hammer was conducting reengineering seminars specifically on ERP and in particular SAP.

The BPR message closely resembled the total systems movement started in the late 1950s. Recall Wanner’s 1958 presidential keynote address to the systems men where he called for systems men to help organizations redesign their business processes. However, there were some important differences between the BPR in the 1990s and the proposed reengineering in the late 1950s. Although BPR called for integrated management, it also argued for decentralized control of its processing. In contrast, the total systems movement called for centralized control of the integrated system. In addition, this movement came
from third party advisors and not internal IT professionals which made it more palpable within the organization.

In many respects, BPR's juxtaposition of data centralization and process decentralization more closely resembled the JIT philosophy of pushing responsibility to the point of where the work actually is performed. One way to describe this approach is centralization of the data but decentralization of the work. Client/server architectures enabled such a work process design. Distributing work to the client and the improved networking capabilities allowed for client/server applications to support geographically dispersed teams. In addition, it enabled the business user to use the application from the desktop and the scale of the system supports more users which in turn can support the decentralization of work. A relational database system helped solve the problem of the inability to relate between data in the hierarchical database. As such, applications built on this technology can capture the information at the source through the decentralization of work, store it locally in tables, but control it centrally through an effective data schema.

From a market size perspective, worldwide BPR sales grew rapidly from $230M to $750M in 1992 (Newmann, 1991; Rosati, 1992). In 1995 Andersen Consulting added 8,700 consultants and Deloitte & Touche acquired International Consulting Solutions which specialized in SAP implementations (Mullin, 1996). In 1994, the CSC Index found that roughly 78% Fortune 500 were involved in re-engineering and the Institute of Management Accountants' Controllers Council in October 1994 found that 60% of 2200 American companies interviewed were engaged in reengineering (Micklethwait, 1996).
Figure 4.8.2.1 shows the timeline of important events in the development of business process reengineering.

4.8.3 Different customer responses to client/server

At first, customers did not change their software usage pattern as they adopted client/server applications. Studies in 1991 and 1992 from various research groups indicate that roughly 35-45% of the Fortune 1000 engaged in client/server projects (Gantz, 1992; Van Kirk, 1992). In fact, respondents expressed concern about data security with the distribution of data processing to the client and most firms were inclined to minimize
organizational change by implementing the solutions one department at a time (Gantz, 1992). With the introduction of reengineering principles in the early 1990s, a change in use began to emerge among large manufacturers. Rather than automate existing processes, reengineering principles argued for complete functional redesign. This change required firms to think about what these new process should look like (Stevens, 1994).

Soon large manufacturers started making significant changes in how they justified, understood, and deployed planning software. From a justification perspective, large manufacturers started to include analysis of the software’s impact on customer value and strategic advantage. Manufacturers began considering how planning software could help improve customer value by reducing delivery times and improving quality and service in addition to increasing inventory turns and lowering the level of on-hand inventory. An IDC report echoed this new justification logic:

With BPR, performance gains are not based on the design of the system but on the way the system interacts with the business and people. An organization now justifies its need for a system overhaul based on business requirements and new enabling technologies. ... Existing processes radically restructured, results measured in productivity, time to delivery, employee optimization can be used for quality initiative, departmental/enterprise standards addressed (Rosati, 1992).

Reengineering raised an interesting question related to software purchasing – should a firm redesign its processes before or after the software was selected? Historically, firms preferred to design the system before software selection in order to determine the functional and technical requirements to evaluate the software products (see Sections 4.6 and 4.7). During earlier periods, manufacturers certainly recognized that some process
change was necessary, but they preferred to only have minimal changes. Recall how Sundstrand argued that at most 25% of the business processes should change. In contrast, business process reengineering proposed that manufacturers should avoid automating existing “cow paths” and re-design new processes to automate. Therefore, it was not clear at what point a manufacturer should purchase the software. Often software was purchased before the new processes were designed:

Classical reengineering approach will suggest that the design of the new solution be completed to the required level of detail before beginning the design of the enabling information system solution. In reality however, this may not be the practical choice. Many companies get impatient with the long drawn out reengineering program and end up selecting the information system to be implemented, somewhat prematurely. And, then they learn to simultaneously reengineer the business processes in the context of the systems solution. This may indeed be a more effective approach, if coordinated effectively, due to the long time taken for developing a custom solution. ... Well designed integrated solutions may in fact offer rich ideas that may speed up the reengineering design and implementation activity (Ravikumar, 1994, p. 20).

Rather than completely design these new processes first and then buy a software package, this consultant suggested that manufacturers should purchase software concurrently with the process design. As a result, large manufacturers started to deploy software in a different way. Rather than automate pre-existing business processes, software configured these new business processes. This use of software and technology in general could help manufacturers do things differently in order to achieve some level of competitive advantage. As one consultant commented:

Technology is most effective when used to enable business to be conducted differently. It should enable us to focus resources on activities that add value to the products and services we supply to customers (Clay, 1993, p. 69).
The change in the process through which large manufactures purchased software had additional ramifications. Recall that earlier manufacturers would create request for proposals (RFP) that detailed precisely what the software needed to do (see section 4.6). Often in the RFP process, manufacturers would document what the business process did in order to identify the software requirements. Manufacturers would base their decision in part on the vendors' ability to satisfy these detailed requirements. However, with BPR, these processes were not well understood at the time they solicited software vendors. As a result, these RFPs only provided a general sense of the functional specifications the software vendors would need to satisfy. Since consultants were heavily engaged in the broader reengineering effort, they also were much more involved in the selection process with the customers. This was much different than GE originally suggested how manufacturers should use consultants. More than advisors and educators, consultants played a more prominent role in making the software purchase itself, making IDC ponder if consultants were becoming a new class of software customers (IDC, 1987b).

The actual deployment of client/server ERP software was also decidedly different. Rather than use it at the local division or geographical level, many top-tier firms expected to get enterprise-wide information from the systems. Advancements in communications and data management technology enabled firms to decentralize business processing while centralizing data control. Don Klaiss, a

15 This difference in how RFPs were written and used during the sales process was pointed out to me in an interview with Thomas Berquist who was a management consultant in the early 1990s.
vice President of Manufacturing Products for Oracle described this kind of deployment in the following way:

They [enterprise resource planning software] support an expanded view of the enterprise, helping to break down the walls that separate local operations from those of other business units, customers, and trading partners. You can access engineering data, supply-and-demand information and profitability measures from all business units that are part of the enterprise. (Inglesby, 1993).

In contrast, smaller manufacturers had a distinctively different experience with the transition to client/server (see Figure 4.8.1). They did not go through the same transformation in understanding, justification, and deployment because the proposed new uses of ERP software closely resembled how smaller manufacturers already used MRP software. Recall that smaller manufacturers began purchasing software products after the standardization to MRP and MRP II. Although not as integrated at the transactional level as ERP, MRP and MRP II integrated cost accounting with manufacturing, procurement, and some customer ordering functionality. It also captured the core business planning process of a manufacturer. The deployment of this software was centralized in the sense that it was the main, if not only, software used by the firm. Consequently, when the smaller manufacturers began purchasing enterprise software, they tended to buy integrated applications on one type of hardware (AMR, 1995a, b). MRP was the mission critical business processes. ERP simply expanded the functional breadth. Smaller firms did not think of ERP as that different from MRP, nor did they envision it doing anything differently.

In addition, as mentioned IBM released the AS/400 in 1988. This new minicomputer contradicted the new open theme of distributed computing in that the
AS/400 was a fully integrated box with proprietary operating system and data management software. Despite the fact that it was proprietary, it was very reliable and quickly became one of the most successful mid-range computers in the market. Many software firms that targeted smaller manufacturers specialized in AS/400 deployments, e.g. J.D. Edwards, and System Software Associates, and IBM’s Mapics. Because of limited resources, many smaller manufacturers were attracted to reliability and implemented these MRP solutions. However, the proprietary nature of AS/400 made it difficult to integrate with other client/server solutions.

Lastly, as large firms began adopting these technologies, the cost of implementing reengineering sky rocketed with the substantial data conversion, process design, and change management (Interview with Thomas Berquist). These costs with little perceived benefit did not compel smaller manufacturers to change its use from automation to configuration. As such, although smaller manufacturers continued to purchase client/server ERP solutions, they used the applications to automate in a highly centralized manner.

4.9 Conclusion

It is instructive to map out the various patterns of how manufacturers used planning software. Figure 4.9.1 illustrates the usage pattern for large manufacturers. The Y axis
measures the variety in how large manufacturers actually used the software. High variety means that large manufacturers understood, justified, and deployed the software in multiple ways; whereas, low variety signifies convergences on a particular understanding, justification, and deployment. So, low variety signals the presence of a dominant use. The illustration is meant to be conceptual so the actual shape of the line and exactly how high or low is should not concern us here.

Based on the history, initially there was low variety as manufacturers used initial planning software in a conservative manner, in many cases replicating pre-existing tabulating applications. By the 1960’s, variety in use increased as some manufacturers began to experiment with using the software to help production planners actually make better decisions as opposed to automating record keeping. In the early 1970’s, this new use became dominant, largely through the efforts of the “MRP Crusade”. Shortly
thereafter, variety began to increase slightly, mostly in how manufacturers deployed the software as their understanding and justification remained consistent with earlier periods. The introduction of client/server technology corresponded with increased variety as manufacturers began to use ERP software to configure new business processes in an attempt to improve customer service. By the mid 1990s, this new use became dominant.

One of the interesting aspects of this history is that there is not just one commercial introduction of a new technology, but two – the computer and client/server. There were also multiple settings of dominant uses. It is instructive to compare these different episodes. For instance, when manufacturers first experienced the business computer in the mid 1950s, they were very conservative in their understanding and use of it; however, when they experienced new client/server technology in the 1990s, they more quickly developed different understandings and uses of the technology. In both cases, manufacturers made product comparisons in order to understand the technology. These differences show that despite the constraints of a pre-existing knowledge schema, the introduction of a new technology need not be incremental.

When manufacturers first encountered computer applications, manufacturers focused on the similarities between the computer and primarily tabulating machines (recall they also compared the computer with people). The emphasis on similarities encouraged manufacturers to use computer applications to perform the same kinds of tasks as the tabulating machine. In contrast, during the introduction of client/server technology, large manufacturers did not make multiple product comparisons, but focused on the differences between client/server and the host-based MRP applications. Prior to the introduction of
client/server, manufacturers were learning about the limitations of MRP applications. The comparisons of client/server applications with MRP then focused on how client/server was different from these exposed limitations of MRP.

The type of product comparison – whether it is similarities or differences – in addition to the number of comparisons influence whether customers initially adopt radical uses of a new technology. Multiple product comparisons may help differentiate a new product category, but focusing on differences within a comparison may lead to new uses more quickly. Another important difference between the computer and client/server adoption was that client/server was a second generation technological change. In essence, it was a change to a pre-existing product category as opposed to the introduction of a new product category. As a result, the core category to which the new technology was compared was well understood; it was the dominant use. This facilitated comparison, in particular the identification of differences. These insights imply that second generation technological changes may in fact lead to more radical adoption of new uses than first generation technological changes.

Another important feature of the usage pattern was that different customer segments had different patterns. Figure 4.9.2 adds the small manufacturers’ usage pattern. This pattern is remarkably different than the large manufacturers’ despite the fact that each segment essentially used the same technology. Small manufacturers entered the market much later than large, but essentially developed the same understanding and justification. The major difference between the two was in how they deployed to software – small manufacturers tended to deploy the software in a highly centralized manner; whereas, large
Figure 4.9.2 Comparison between small and large manufacturers usage pattern

![Graph showing comparison between small and large manufacturers usage pattern.](image)

manufacturers, in a decentralized manner. Small manufacturers did not significantly change their use of planning software when encountering client/server technology, but large manufacturers did.

Finally, the pattern of use is decidedly different than the pattern of technological change. Figure 4.9.3 adds the pattern of technological change. Variety in technical terms is measured similarly. High variety means that there are multiple design options; whereas, low variety means there is a dominant design. Recall that the unit of technical analysis for this study was software system design. Essentially during this time period there were two different design approaches, host-based architecture until the late 1980s and then distributed architecture. Thus, the high variety in the 1980s represents the introduction of this new technology, which soon converged to the distributed architecture design. The early software design is a bit complicated given that initially software was so closely tied
with hardware. The early high variety reflects the multiple hardware platforms, but this converges quickly to the IBM platform in the beginning of the 1960s with IBM’s commercial success and the release of the IBM 360. The introduction of the minicomputer and PC are indicated because they introduced new platforms; however, from an architecture standpoint, the software design remained essentially the same.

One drawback of explaining variation solely in terms of the technical aspects of software design is that it does not address variation in the software functionality. Even though software vendors may use the same software design, the functionality that they provide in the applications may vary significantly. For example, as mentioned, in the early 1970s, IBM believed that an advantage of its COPICS product was its breadth of functionality and integration of the different applications. Most software at the time only
focused on one particular function, such as MRP. That said, the “MRP Crusade” did help
to standardize the functional requirements for the initial MRP solutions.

Yet, it is possible that the pattern of variety in application functionality offered by
the vendors may approximate the pattern of customer use. In addition, the difference in
functionality between different customer segments may explain the difference in their
usage pattern. Although important, this dissertation does not systematically address this
alternative view. To do so requires gathering marketing information about products like
COPICS over time as well as application documentation. However, as discussed, the
functionality of the application in use is not fully determined by the functionality offered
within the product. Customers often customize the software product to process company
specific functionality. As a result, variety in software functionality may result from
customer specific application which may differ from the pattern of functionality offered
within the product.

Presenting the historical account in this way helps show the decoupling of the
pattern of how manufacturers used planning software from the pattern of technological
change. For instance, there are periods like in the early 1960s where the variety in
customer use increased while the technical characteristics remained essentially unchanged
(there were changing as noted in section 4.4, but in incremental ways). Similarly, one
customer segment, small manufacturers, did not change their use of software even though
there was a significant technological change to distributed computing. The fact that these
processes are different suggests that to fully explain the significance of the technological
change on the software vendor's population dynamics requires addressing both the
customer usage and technological change patterns. This is the task of the next chapter.
Appendix A to Chapter Four – A brief overview of the U.S. Manufacturing Sector

The U.S. Census Bureau defines the manufacturing sector as those “establishments engaged in the mechanical, physical, or chemical transformation of materials, substances, or components into new parts” (see http://www.census.gov/epcd/naics02/def/NDEF31.HTM#N31-33). There clearly is wide variety in how this transformation is done. Assembling the different components to build a car differs significantly from mixing the right ingredients to make aspirin. To get a better sense of what the manufacturing is, this appendix outlines the different styles of manufacturing. It also provides a brief history of the manufacturing planning function prior to the introduction of the computer in the 1950s. Lastly, this appendix identifies the demographic, organizational, and macroeconomic changes in U.S. manufacturing during the period of historical analysis. The preconditions of the manufacturing function and the changes in U.S. manufacturing influenced how manufacturers thought of and used planning software.

Highlighting the different ways to classify types of manufacturers helps clarify what manufacturing is. A good place to start is with how the U.S. government categorizes different industrial firms. In 1938, the government established the Standard Industrial Classification (SIC) system to classify establishments by their major industry and business activity (In 1997, the North American Industry Classification System (NAICS) replaced the SIC system). At the industry level, these classification systems distinguish manufacturing from other industries such as agriculture, mining, services, finance, and transportation. Manufacturing is further categorized into 17 groups (in the 1987 SIC
system) based upon business activities and type of products manufactured: food, furniture, petroleum refining, metal, electrical equipment, paper, and transportation equipment. Each of these major groups is further classified into more specific product categories. For instance, tobacco is segmented further into four groups - cigarettes, cigars, chewing and smoking tobacco and snuff, and tobacco stemming and redrying. In essence, this classification system identifies the different product markets that manufacturers target.

For our purposes, this is interesting because building a product like an airplane is fundamentally different than the synthesizing of chemical compounds to make laundry detergent. Related to the industry classification system is the style of manufacturing. At a high level, there are discrete and process styles of manufacturing. Process manufacturing converts raw materials into a finished good through a continuous production flow or in batches. Examples include oil refining, chemical synthesis, and paper. Discrete manufacturing assembles an engineered product by combining physical discrete sub-components. Examples include automobiles, computers, and toys. These process differences affect how the manufacturer plans and controls the production and inventory processes. Process manufacturing typically involves the mixing of a limited number of ingredients (called recipe management), continuous automation of the production process with tight capacity constraints, different levels of units of measures (material is stored in kilograms, but the recipe only uses a trace amount to produce), and variation in the ingredients as the final product is produced. In contrast, discrete manufacturing typically deals with the assembly of many different discrete units and a higher labor component but it is less concerned about degradation and by-products during the production process. As a
**Table 4A.1: Distribution of U.S. Discrete/Process Manufacturing, select years, 1950-2000**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td>26%</td>
<td>31%</td>
<td>32%</td>
<td>32%</td>
<td>32%</td>
<td>34%</td>
<td>34%</td>
</tr>
<tr>
<td>Process</td>
<td>53%</td>
<td>55%</td>
<td>55%</td>
<td>54%</td>
<td>54%</td>
<td>53%</td>
<td>53%</td>
</tr>
<tr>
<td>Other Mfg.</td>
<td>21%</td>
<td>15%</td>
<td>13%</td>
<td>15%</td>
<td>14%</td>
<td>13%</td>
<td>13%</td>
</tr>
</tbody>
</table>

Source: Based on categorizing manufacturing industries into discrete and process from industry breakdown found in (Cortada, 2004, p. 77). This is somewhat consistent with IDC’s breakdown which estimates that roughly 60% of manufacturers in the U.S. are process.

As a result, management of material requirements and scheduling the production process is much more of a significant issue for discrete manufacturers; whereas, the management of the production process itself is vital to process manufacturers. If one were to map the SIC industry classification establishment census statistics to style of manufacturing, the distribution of discrete and process oriented manufacturing has remained fairly stable at 53% process, 32% discrete, and 15% other (see Table 4A.1).

More specifically, some firms make products to a forecast (called make-to-stock) while others, to specific orders (called make-to-order). In many cases, make-to-stock involves repetitive production of similar items (computers) and make-to-order produces highly engineered products (airplanes) in a job-shop environment. The differences in manufacturing style also have implications for how the manufacturer plans and controls its production operations. For example, the aerospace and defense industry typically manufactures its customized make-to-order products as projects instead of producing to a scheduled forecast as in the more repetitive automobile industry. Running manufacturing as projects requires assigning labor to different project tasks, tracking completion times,
and reporting interim production reports. As a result, a software solution that targets a
highly repetitive process manufacturer such as Coca-cola will not be well suited for a
custom order and project oriented discrete manufacturer such as Boeing.

4A.1 Manufacturing operating processes and use of technology pre-computer

Prior to the introduction of the computer in the 1950s, how manufacturers managed
production and inventory processes was in flux.\(^1\) Unlike accounting procedures which
were fairly stable by the 1950s (Johnson & Kaplan, 1987), production planning lacked
standardization, formalization, and documentation. Practitioners distinguished between the
“formal” and “informal” systems (see (Wight, 1974, pp. 5-10) for a discussion on
formal/informal systems). The formal system essentially established production
requirements and what material was needed to manufacture the goods. However, in
production, the manufacturing environment constantly changes. Customers change orders,
a machine may break, people get sick, or an expected inventory order may not arrive. As a
result, there also existed an “informal” system that managed the flow of material through
the production process. Manufacturers had “move men”, “stock chasers”, “expediters”
with “hot lists” who essentially expedited inventory as production requirements changed.

\(^1\) I am not familiar with any historical account of production and inventory management comparable to
Johnson and Kaplan’s analysis of accounting management (Johnson & Kaplan, 1987). Consequently, this
discussion uses secondary sources. Pre-scientific management history is based upon secondary historical
accounts, in particular, (Chandler, 1977) and (Scranton, 1992) as well as descriptions provided by
practitioners, most notably, (Plossl & Wight, 1967). Scientific management practices are well documented, in
particular in (Aitken, 1985). I have taken the liberty to described production and inventory control as
integrated, which is a modern view. Wight argues that production and inventory control developed along
different paths and it was the scientific management movement that integrated the two together.
This system was informal because it operated on intuition and experience and often changes made on the shop floor were never recorded in the formal planning system. Therefore, the production planning process is inherently different than accounting processes because it must react to a changing environment. Accounting processes “account” for what already occurred, processing a set of given transactions based upon predetermined business logic such as commission rates or tax rates (which still is very complicated to program).

By the 1950’s, there was some formalization of the “formal” planning process. Historically, the responsibility of managing inventory levels and the flow of material and completion of orders historically fell upon the foreman. The Scientific Management movement at the turn of the 20th century helped shift the responsibility of the production and inventory planning and control from the foreman to a planning department. In Shop Management, Frederick Taylor insisted that the foremen should report to the planning department (Taylor, 1911). A vivid example was the production planning department at the Watertown Arsenal in 1908 - 1915. The planning room was the “nerve center” and included 20 men: the leader or master mechanic, route-sheet clerk, production clerk, rate-setting department, move men, cost-keeping, and a “window man” who time stamped job cards (Aitken, 1985, pp. 131-4). The responsibilities of the planning room included maintaining assembly diagrams, preparing, monitoring, and filing route sheets for the production process, procurement of component parts, cost record keeping, and expediting the movement of inventory. Adoption of scientific management was not restricted to government. Chandler briefly described similar production planning practices at the
manufacturers Remington and Yale and Towne during the same time period (Chandler, 1977, pp. 275-279).

However, it is not entirely clear how widely the manufacturing community adopted scientific management practices and specifically the production planning department by the introduction of the computer in the 1950s. GE and other large manufacturers had these departments but it was likely that many smaller manufacturers still managed the process on the shop floor (Plossl & Wight, 1967). It is important to note that although scientific management helped establish a production planning group, this group was essentially clerical in nature. During this time period, the manufacturing function was generally viewed as a cost center. Not until the 1960’s did manufacturers begin to consider production an important element of corporate strategy. In a series of Harvard Business Review articles dating back to 1966, Wickham Skinner of Harvard Business School emphasized manufacturing as an essential part of corporate strategy (he later wrote a popular book based on these articles (Skinner, 1978)). Also during this time, the professional society, American Production and Inventory Control Society (APICS), emerged in part to increase the profile of production planners as an important member of the management team.

The primary technique planning departments were using to plan and schedule production was called “re-order” point (Orlicky, 1975). This methodology uses historical forecasts at the individual part level to determine the inventory level when new inventory needs to be re-ordered for the part. Using historic use data and forecasts, re-order point calculates the order quantity and safety stock requirements. Over the years, mathematical
techniques have been applied to this technique to improve forecasting and safety stock calculations. Economic order quantity (introduced in 1915) as well as more sophisticated re-order point (1934) calculations improved the stock replenishment process; Gantt charts (around 1910) were applied to machine loading, Program Evaluation and Review Techniques (PERT - 1958) and Critical Path Method (CPM - 1957) were applied to project planning; and operations research emerged after WWII to provide a flurry of techniques. Although these advances improved efficiencies and performance, they neither addressed the fundamentals of how production planning should be organized and controlled nor the informal system of expediting material. As discussed in the chapter, materials requirement planning (MRP) approach introduced in force in the 1970’s would offer an alternative way to plan and schedule production requirements.

From a technology perspective, firms applied electro-mechanical equipment to help manage the formal planning system. By the 1950’s many larger manufacturers used Kardex or punch card systems to record re-order points for parts, records of purchased parts, and inventory levels (see (Niece, 1957) – a contemporary book on production techniques that describes the use of these systems). In many ways these file systems contained the information originally managed in Taylor’s planning department and were used for production planning purposes. The informal system of expediting continued to be managed through experience, not Kardex files, and rarely ever was documented in the formal system. So, even with the card systems in place, it was hard to maintain the accuracy given the changing conditions. In addition, many manufacturers lacked essential documentation for effective material planning. In particular, many manufacturers did not
have a formal bill of materials – a listing of parts and subassemblies and quantities needed to manufacture a product (Plossl & Wight, 1967). Without this documentation, it was difficult to accurately and consistently calculate re-order points for products.

Therefore, when computers were first introduced in the mid 1950s, the production environment lacked systematic documentation and data accuracy that made conversion to the computer attractive. To the extent that manufacturing applications were converted, they tended to be the formal aspects of the manufacturing planning process which had some documentation and procedures.

### 4A.2 Macro factors influencing the manufacturing market

During the period of historical analysis, the U.S. manufacturing environment started to experience a series of economic and competitive changes that increased focus on production and inventory processes. Between 1954 and 1980, there were 4 recessionary periods (see Table 4A.2.1). These recessions affected demand as well as inventory and labor costs, encouraging manufacturers to more efficiently manage inventory and the production process. With these fluctuations, however, forecasting demand became more difficult and customer service became more important.

In addition to economic downturns, increased global competition exposed U.S. manufacturers to alternative production management techniques. Most prominent was the exposure to the Japanese Just-in-time systems in the mid-1980s. Primarily through Japan’s successes in the automobile and electronics industry, U.S. manufacturers began to identify and consider the differences between the U.S. MRP style and Japan’s JIT. In 1985, APICS
Table 4A.2.1: Various U.S. Business Cycles, 1953-1996

<table>
<thead>
<tr>
<th>Peak Date</th>
<th>Trough Date</th>
<th>Contraction Period (Months)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1953 - July</td>
<td>1954 - May</td>
<td>10</td>
</tr>
<tr>
<td>1957 - August</td>
<td>1958 - April</td>
<td>8</td>
</tr>
<tr>
<td>1960 - April</td>
<td>1961 - February</td>
<td>10</td>
</tr>
<tr>
<td>1969 - December</td>
<td>1970 - November</td>
<td>11</td>
</tr>
<tr>
<td>1973 - November</td>
<td>1975 - March</td>
<td>16</td>
</tr>
<tr>
<td>1980 - January</td>
<td>1980 - July</td>
<td>6</td>
</tr>
<tr>
<td>1981 - July</td>
<td>1982 - November</td>
<td>16</td>
</tr>
<tr>
<td>1990 - July</td>
<td>1991 - March</td>
<td>8</td>
</tr>
</tbody>
</table>


commissioned a study to in which practitioners traveled to Japan to learn and evaluate
Japanese JIT production techniques. JIT also became a large and dominant discussion
section at the 1980’s conference proceedings. Rather than control manufacturing through
highly rationalized production plans, the Japanese JIT system pulled inventory through the
production process with the intent to eliminate material in the production process as well
as the time and resources to build the product.
Appendix B to Chapter Four – A brief overview of planning software technology

The technical design literature describes technology in terms of nested hierarchical sub-systems (Baldwin & Clark, 2000; Murmann & Frenken, 2004; Tushman & Murmann, 1998). A computer, for instance, consists of nested systems of data storage, processors, and operating software. The hierarchical systems image is particularly useful for a technical discussion of planning software because it consists of several related components and must interface with other systems in order to operate. Planning software requires hardware to run on, databases to manage the data, and networking applications to distribute the processing of the application.

As discussed in the chapter, the main technical unit of analysis is the software system design. We can think of software design as the different ways these component systems relate to each other in order to function. Technically, enterprise software consists of programming code. Within the software program, there are three interrelated components: code to manage the user interface, to manipulate the data and interface with the hardware, and to process the business logic. The user interface refers to how the user interacts with the software program. Today, we know it as the graphical user interfaces that we see on web pages, but originally there was not electronic access. At first, punch cards and reports were the main interaction users had with software programs. That data management code administers the data that supports the business transactions. For instance, a planning application requires information about product structures, inventory levels, customer orders, purchase orders, etc. The data management code helps structure the various data elements to ensure that it can be retrieved in an efficient manner while also
preserving the integrity of the data. The business logic processes the tasks necessary to complete a transaction within the system. It establishes the business rules the users must follow, for example, purchase orders over a certain amount require approval from a manager.

As mentioned, these components do not work unless they integrate with other technology in the system. Hardware, such as mainframes, minicomputers, or personal computers, are required to store the data and process the transactions. However, application software does not interface directly with the hardware; rather, the operating system mediates the hardware machine code and application code. Communication hardware and software as well as transaction processing software helps manage distributed transactions. Figure 4B.1 identifies the various components of the software design stack as well as different vendors who supply solutions at each of these levels.

The software system design refers to how the software program relates to these other components in order to process its business transactions. During the time period of the study, there essentially were two distinct design approaches. The first was called host-based architecture because all transaction processing happened on a host machine, typically a mainframe. The second architecture emerged in the late 1980s and was called distributed architecture (the particular form of this approach that took hold was called client/server). Unlike the host-based architecture, this architecture supported processing across multiple hardware platforms. Typically, one server would manage the database requests; another, the business transactions; and the third, the interface layer.
This change in architecture represented the culmination of many different changes at the component level. Figure 4B.2 outlines some of the significant technological changes at each level, highlighting when client/server took hold. For example, at the operating system level, UNIX emerged which could operate across a heterogeneous hardware components. Prior to UNIX, each operating system was proprietary to the hardware solution. At the database level, relational databases emerged that facilitated linking disparate data tables together at the data element level. This more sophisticated data management centralized control of data integrity but allowed for distributed processing of
the data itself. Currently, software system design is working through yet another architectural change – the Internet.

Perhaps the best way to understand the technological changes is to see how the software interfaces have changed over time. Figure 4B.3 shows SAP’s user interface during the host-based era and later with client/server. During the host-based era, the terminal did no processing of the data. The interface was very simplistic, only allowing for sequential data updates using a cursor. The second screen shows the interface of a client/server version of SAP’s software. In this case, the terminal did some basic data
Figure 4B.3: SAP's user interface

R/2 Terminal Screens (Green Screen: 1970s & 80s)

R/3 version 4 (Graphical: early 1990s)

Source: http://www.sapdesignguild.org/resources/r3_history.asp

editing like ordering data such that it could be presented in a drop-down box. This screen is much more graphical in nature and less sequential.
Appendix C to Chapter Four – Key historical events in the evolution of use

Table 4C.1 provides a summary of the key historical events among the significant factors that affected the manufacturer’s usage pattern: technology, market intermediaries, software vendors, and customer actions.

Table 4C.1: Summary of key historical events

<table>
<thead>
<tr>
<th>Year</th>
<th>Technology</th>
<th>Intermediators</th>
<th>Software Vendors</th>
<th>U.S. Manufacturers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1952-1959</td>
<td>1952: FORTRAN created by team at IBM</td>
<td>1957: American Production &amp; Inventory Control Society (APICS) started in Cleveland, OH</td>
<td>Late 1950s: IBM releases Bill of Material Processor (BOMP)</td>
<td>Mfg considered a cost area</td>
</tr>
<tr>
<td></td>
<td>1954: UNIVAC commercially released</td>
<td></td>
<td></td>
<td>Use reorder point production process</td>
</tr>
<tr>
<td></td>
<td>1956: 1st computer to use disk drive: IBM’s 305 RAMAC</td>
<td></td>
<td></td>
<td>1958: Mfg’s purchased 42.4% of computer systems</td>
</tr>
<tr>
<td></td>
<td>1959: IBM releases solid state computer: 1401</td>
<td></td>
<td></td>
<td>1959: P. Bacigalupo (IBM) issues net-change MRP at American Bosch</td>
</tr>
<tr>
<td></td>
<td>Codasyl formed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>Technology</td>
<td>Intermediators</td>
<td>Software Vendors</td>
<td>U.S. Manufacturers</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------</td>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1966: IBM releases PICS</td>
<td>Production and Inventory Control:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Principles and Techniques</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1968: Mfg's purchased 33.6% of computer systems</td>
</tr>
<tr>
<td></td>
<td>DEC releases minicomputer: PDP-8</td>
<td>1969: O. Wight forms education/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IBM releases IMS database technology</td>
<td>consulting company</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Late 1960's: CRT terminals begin to</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>replace teletype</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>Technology</td>
<td>Intermediators</td>
<td>Software Vendors</td>
<td>U.S. Manufacturers</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>1970: Edgar Codd (IBM) creates relational database model</td>
<td>1971: Database Task group publishes standard called Codasyl approach</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1971: Database Task group publishes standard called Codasyl approach</td>
<td>1972: C developed at Bell Laboratories</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1974: IBM releases SNA protocol</td>
<td>1979: Oliver Wight starts MRP II ABCD evaluation methodology</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1975: J. Orlicky publishes book on MRP</td>
<td>1979: Relational Software (now Oracle) releases first commercially available SQL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>Technology</td>
<td>Intermediators</td>
<td>Software Vendors</td>
<td>U.S. Manufacturers</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>----------------</td>
<td>-----------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>1983</td>
<td>Unix announces will standardize different flavors of UNIX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1986</td>
<td>Com. Availability of distributed database management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1988</td>
<td>IBM release AS/400</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>Technology</td>
<td>Intermediators</td>
<td>Software Vendors</td>
<td>U.S. Manufacturers</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>----------------</td>
<td>------------------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>
| 1990-2000 | 1993: Windows NT Released  
1994: Netscape releases Navigator web browser  
1995: Java programming language released | 1991: APICS begins certification in CIRM  
SOP 91-1 issued providing revenue recognition guidelines  
1992: Gartner coins term ERP  
1993: Reengineering the Corporation published  
1997: SOP 97-2 issued superseding SOP 91-1 | 1994: IBM exits the market with the sale of MAPICS  
2000: Microsoft enters the market with the purchase of Great Plains | Early 1990s: Large manufacturers begin reengineering businesses |
5.1 Introduction

Once we recognize that change in how customers use a technology may be orthogonal to changes in the technology itself, the next logical questions is do these changes in use influence the population dynamics of an industry, i.e., the number and types of firms we see in an industry? This is the central question of the chapter.

To begin to answer this question, I use details from the historical case to help explain the entry/exit pattern of the planning software industry. The planning software industry is a particularly good case to study these issues. First, this industry exhibits the anomaly originally presented to introduce this research: an established firms successfully transition through a discontinuous technological change. In fact, one established firm, SAP, has become the market leader in ERP software. Second, one customer segment, large manufacturers, significantly changed its usage of software after the technological change to client/server; whereas, the other segment, small manufacturers, did not. There were different patterns of use in response to the same technological change. This feature of the case enables for a comparison between the entry/exit rate of these two different segments to begin to isolate the effects of customer use.
To build my argument, I first consider alternative explanations from prevailing evolutionary-based theories of industry evolution and theories of why firms fail in the face of disruptive technological changes. I find that although these explanations provide some additional insight, they are not consistent with the historical facts of the case. I then develop a more historically accurate explanation which more strongly considers changes in how manufacturers used the software. In particular, I argue that the emergence of a new dominant use, even without a corresponding technological change, can lead to significant new market entry. In addition, established firms are less likely to fail when customers use a new technology in similar ways to how they used the old technology. More generally, firms — established or not — are more likely to survive the introduction of a new technology if they are able to identify and adapt their products to how customers develop uses for that product. These insights help explain why established firms may persist through significant technological changes.

5.2 Entry/Exit patterns of planning software industry

Figure 5.2.1 shows the entry/exit pattern for planning software from 1954 through 1996 (See Appendix 5A for a description of the coding). Like many technology industries, the planning software industry exhibits an inverted U shape. Early on in the industry many firms enter, but soon thereafter the exit rate is greater than the entry rate. Accounts of the population dynamics of an industry are particularly interested in explaining inflection
Figure 5.2.1: Entry/exit pattern for U.S. manufacturing planning software market, 1954 – 2003

Source: See Appendix 5A

points in the model (Geroski, 2001). In this particular industry, there are two such periods: the significant increase in market entry in the late 1960s and early 1970s and the increase in market exit in the late 1980s. I shall address each in turn.

One possible explanation of the early market entry is related to the product lifecycle: at industry birth, high levels of market uncertainty about user preferences and technology performance make it attractive for firms to enter the market (Utterback &
Abernathy, 1975). Because firms initially have different technical capabilities (Klepper, 1996; Nelson & Winter, 1982), they typically pursue different types of product innovation. As a result, during this period of market entry, there are variants in the technical design. According to this view, during market entry we should see variety in both demand and technology design. However, the historical details of the software case do not support this explanation. The increase in market entry occurred in the 1970s, more than 10 years after the first planning software market entered in 1958.\footnote{It is entirely possible that I have incorrectly specified the beginning of the industry. A case could be made that the industry really began in the 1970s with the introduction of pre-packaged software. Typically, industry studies define industry emergence with the commercial introduction of the first product. However, to do so does not capture the fact that manufacturers started to develop applications in the 1950s and hardware vendors offered some application specifications as early as 1959. Therefore, it is not entirely clear what the first product is. This raises some interesting theoretical and methodological questions about locating the beginning of an industry. Answering such questions are best reserved for future research.} From Figure 4.9.1, it is evident that significant market entry occurred during the ‘MRP Crusade’ – the establishment of a new dominant use. Thus, there was not high market uncertainty; rather market preferences were becoming more specified. And, from a technology perspective, the software design had already standardized on the host-based architecture. As a result, during this time period different software firms were not proposing alternative technical designs; rather, they were providing technically similar solutions based upon the host-based architecture. (see how the variety in technical design is decreasing in Figure 4.9.3).

Another possible way to explain the market entry is that demand was increasing – there was an outward shift in the demand curve attracting firm entry. According to this explanation, as new firms enter, customers begin to perceive an improvement in technology quality, leading to a takeoff in sales which induces further firm entry (Agarwal & Bayus, 2002). If this account is correct, we should expect to see an increase in sales that
Table 5.2.1: Shipments of different kinds of computers, 1975 - 1990

<table>
<thead>
<tr>
<th>Year</th>
<th>Mainframe</th>
<th>Minicomputer</th>
<th>Personal computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>6,700</td>
<td>26,990</td>
<td></td>
</tr>
<tr>
<td>1976</td>
<td>6,750</td>
<td>39,320</td>
<td></td>
</tr>
<tr>
<td>1977</td>
<td>8,900</td>
<td>56,780</td>
<td></td>
</tr>
<tr>
<td>1978</td>
<td>7,500</td>
<td>68,340</td>
<td></td>
</tr>
<tr>
<td>1979</td>
<td>7,200</td>
<td>81,250</td>
<td></td>
</tr>
<tr>
<td>1980</td>
<td>9,900</td>
<td>105,870</td>
<td>486,000</td>
</tr>
<tr>
<td>1981</td>
<td>10,700</td>
<td>121,990</td>
<td>905,000</td>
</tr>
<tr>
<td>1982</td>
<td>10,600</td>
<td>128,000</td>
<td>3,775,000</td>
</tr>
<tr>
<td>1983</td>
<td>9,980</td>
<td>146,800</td>
<td>7,623,000</td>
</tr>
<tr>
<td>1984</td>
<td>11,330</td>
<td>205,400</td>
<td>9,670,000</td>
</tr>
<tr>
<td>1985</td>
<td>10,910</td>
<td>190,800</td>
<td>8,828,000</td>
</tr>
<tr>
<td>1986</td>
<td>10,990</td>
<td>198,200</td>
<td>9,816,000</td>
</tr>
<tr>
<td>1987</td>
<td>11,200</td>
<td>205,800</td>
<td>11,130,000</td>
</tr>
<tr>
<td>1988</td>
<td>11,540</td>
<td>218,100</td>
<td>12,380,000</td>
</tr>
<tr>
<td>1989</td>
<td>11,890</td>
<td>227,700</td>
<td>13,500,000</td>
</tr>
<tr>
<td>1990</td>
<td>12,130</td>
<td>232,000</td>
<td>14,560,000</td>
</tr>
</tbody>
</table>

Source: (Steinmueller, 1996a)

corresponds with the timing of firm entry. However, that is not the case either. At a macro level, demand for using computers and computer applications was growing rapidly during the 1970s. For example, Table 5.2.1 shows the shipments of computer equipment growing significantly during this time period. However, growth in investment in computer hardware did not necessarily translate into a significant sales takeoff in pre-packaged software. Recall that historically manufacturers developed computer applications either internally or through the assistance of consultants. In the 1970s, there still was a clear preference for internal development. Table 5.2.2 shows the growth of software spending across the U.S. economy, distinguishing between spending of pre-packaged applications, customer development with the aide of consultants, and own, in-house development. During the 1970s, although spending of software packages grew, it remained a relatively small percentage of total spending. Firms were still focused on developing applications as
opposed to buying them. More specifically, Figure 5.2.2 illustrates the sales pattern for the industry.\(^{18}\) Note that sales takeoff does not occur until the early 1990s, when the

---

\(^{18}\) Note that the sales figures do not go all the way back to industry inception. This is because the early entrants mostly were hardware firms who mainly did not charge for their solutions. In the 1970s, vendors started to charge for the software, but I am not aware of any industry group that tracked this information at an industry level. Indications from the individual sales experiences indicate that although the market was
number of firms is actually decreasing, not increasing. The timing of this takeoff also corresponds with the significant increase in software package buying across the U.S. economy (see Table 5.2.2). Thus, although demand for software in general was increasing during this time period, it was not the case that demand for pre-packaged planning software was quantitatively increasing.

Yet another possible explanation is that market entry helped established the new independent software firms as a legitimate kind of business to perform this function. Most growing, the growth rate did not appear to be any different than in the 1980s. Brief industry histories from the analyst accounts in the 1980s corroborate this assumption.
of the firms that entered during this time period were independent software firms, which represented a new organizational form. According to population ecologists, the additional entry of a new independent software firm would help legitimatize these new software firms (Carroll & Hannan, 2000; Hannan & Freeman, 1977). However, although independent software firms were the dominant type of firm to enter, they were not the only kind. Consultants, hardware, and even manufacturers themselves continued to enter. Also, considering that the macro software spending data indicates that customers preferred internal development, it is not clear that they thought independent software vendors were the legitimate organizational form to provide this service.

More importantly, independent software firms faced some institutional hurdles that were not resolved until well after the period of market entry. A series of *Datamation* articles in 1968 pointed out how difficult it was to interpret software companies. In some respects, they resembled products companies because they licensed software programs that firms would implement. But, software products were not tangible. This had legal ramifications for patentability, tax implications for R&D initiatives, and revenue recognitions issues from the accounting boards and the SEC. For example, software companies viewed their code as an asset like a piece of manufacturing equipment and wanted to amortize its development over time. However, the returns on software were unclear, suggesting that its development should be expensed as incurred. Because software often required a fair amount customization to get it to function within a company, it was not clear how to recognize this revenue – at the time of purchase or when the customization work was completed. To represent themselves in these important issues
with the government, software firms formed the Association of Independent Software Companies, which merged with the Association of Data Processing Service Organizations (ADAPSO) in 1972. In particular, ADAPSO supported the software industry’s attempts to prevent state governments from defining the purchase of software as a taxable entity (Haigh, 2005). ADAPSO was also active with the R&D issues as considered by FASB. Other important regulatory events included: the passage of the Copyright Act in 1976 which enabled software firms to copyright software code, the 1985 FASB 86 ruling which provided guidelines for R&D expenditures, and SOP 91-1 and SOP 97-2—two rulings that explained software revenue recognition guidelines in 1991 and 1997 respectively. Thus, it took some time for the government and customers alike to recognize independent software vendors as a distinct and legitimate kind of business. Legitimization of these vendors was not just a spillover from increased market entry, but it required government lobbying and active participation from industry groups.

In contrast to these explanations, a more historically accurate explanation of increased market entry during the 1970s accounts for the “MRP Crusade” and IBM’s unbundling decision. No doubt IBM’s decision to unbundle software from hardware made it economically viable for independent software firms to enter the market. In addition, the rapid growth of more standardized mainframes and minicomputers during this time period (see Table 5.2.1) also stimulated software sales.

However, the unbundling decision alone is not sufficient to explain the increase in firm entry. For example, there was significant firm entry in other software areas prior to IBM’s unbundling decision. In the mid 1960s, Lawrence Welke began to compile a
software directory (called the ICP) for firms interested in purchasing pre-packaged software. In 1967, prior to IBM’s unbundling decision, he identified over 150 different firms who provided a variety of software solutions, including system software, file maintenance, and accounting (Haigh, 2004; Lilly & Smith, 2001). In fact, if one were to plot the entry/exit patterns for accounting software market (per Lawrence Welke’s directory), the increase in market entry would occur before IBM’s unbundling decision.

In addition to IBM’s decision, the “MRP Crusade” played an important role. The significant event that occurred in the commercial market was not a quantitative expansion as the sales takeoff thesis suggests, but a qualitative change in how customers valued and used the software. The introduction of a new dominant use that reflected different preferences created the market opportunity for software entrepreneurs. From the previous chapter, recall that MRP valued solutions that supported improved decision making as opposed to systems that automated record keeping. This shift in understanding and justification created different customer needs and requirements. And, APIC’s “MRP Crusade” helped establish this as the dominant use among manufacturers.

Software entrepreneurs during this time period recognized the role of APICS and the benefits of the standardized MRP approach it promoted. In many cases, early pre-packaged software firms initially provided application development services to manufacturers primarily because access to start up capital was restricted. In the late 1960’s, the stock market started reacting negatively toward service provider stocks over concerns of long term viability – a burst of a technology bubble similar to the Internet (Campbell-Kelly, 2003). In fact, it would take close to ten years for a software firm to go
public. In addition, the technology venture capital business was just beginning to emerge. Against this negative investment sentiment, it was difficult for entrepreneurs to raise initial money. So, providing services to custom develop planning applications generated revenue to help fund the development of pre-packaged applications. APICS’ crusade to educate manufacturers about MRP helped establish more standard practices which software entrepreneurs could leverage in their product offering. A good example of this is ASK Computer, which was founded by Sandra Kurtzig in 1972 (for a detailed account of the firm history see her book, (Kurtzig, 1994)). Initially, the firm did customer development for various manufacturers, but soon discovered similarities between how these firms operated and leveraged APICS’ definitions of MRP to help standardize its product offering. As a result, many software firms used APICS as a form of marketing, frequently presenting at APICS conferences and exhibiting their solutions.

The most successful at this form of marketing was IBM, who quickly gained close to 50% market share in the MRP market. IBM’s strength in hardware certainly contributed to its success, but in addition, IBM actively participated in APICS’ “MRP Crusade”. At some point, they even helped fund it and provided some of the videotape materials (Lilly & Smith, 2001). Most significantly, IBM hired many of the early thought leaders in MRP – including Joseph Orlicky, George Plossl, and Oliver Wight – into its manufacturing consulting and education groups. This group educated manufacturers on production and control techniques. The “fathers” of MRP and production control also helped develop modules for IBM’s initial manufacturing solution, PICS. And, then Plossl and Wight left
IBM to form consulting groups that advised manufacturers on MRP practices and software selection. It was no surprise that IBM’s solution matched the market’s preferences well.

As a result, the emergence of a new dominant use, MRP, helped more clearly define the manufacturers’ requirements. More clearly specifying these needs led to significant new market entry.

Let us now turn to the period of market exit in the late 1980’s and 1990s.
This period of increased exit coincided with the introduction of the new client/server architecture. The technology management literature classifies technological changes based upon the magnitude of the change along economic, technical, and organizational dimensions (see Figure 5.2.3 for a summary). Economic differences focus upon whether the new product is a substitute for old technologies (Arrow, 1962); technical differences distinguish between changes in components or the architecture of the product being sold (Henderson & Clark, 1990); organizational differences consider whether the producers of the technology have the relevant capabilities (Tushman & Anderson, 1986). Some technological changes are judged to be incremental (or sustaining), others radical, and still others architectural. Based on this classification scheme, the introduction of client/server was of the architectural and radical variety. To build client/server applications required developing new capabilities in new component technologies like relational databases and learning how to restructure the relationship between these components. Based on this classification of client/server as radical/architectural technological change, the technology literature argues that incumbent firms are more likely to fail (Arrow, 1962; Burns & Stalker, 1966; Henderson & Clark, 1990; Tushman & Anderson, 1986). Incumbents are likely to fail because they either lack the incentives to invest properly to make the necessary technological changes (Reinganum, 1983) or they are incompetent in their ability to develop the new capabilities or routines necessary to create the new technology (Henderson & Clark, 1990; Nelson & Winter, 1982; Tushman & Anderson, 1986).
In fact, the transition to client/server was disruptive to incumbent firms – 31% of the pre-existing firms left the market between 1987 (introduction of client/server applications) and 1996. However, making the technological change did not seem to be the significant competitive issue. Across the whole sample, most firms released a client/server application by 1994. In fact, many of the firms that exited, such as ASK Group and Dun & Bradstreet, did so after they had released their client/server application.

Christensen and colleagues offer an alternative explanation that focuses on market structural changes (Adner, 2002; Christensen, 1997; Christensen & Bower, 1996; Christensen & Rosenbloom, 1995). A new technology may attract a new customer segment not currently served by the incumbents. The new technology actually is inferior to the existing technology along the dimensions that the mainstream values, but meets the needs of the new segment. Over time, the technology improves along the dimensions the main market values and comes to replace the old technology. Because incumbent firms allocate their resources to the main market, they miss seeing these disruptive forces building until it is too late. For client/server to be disruptive as Christensen describes, a new customer segment would have to emerge that uses the technology in a different way than the main market. In addition, the main market does not change its preferences. If they did, the technology would lose its disruptive power, given the producers close attention to the needs of the current market.

At the time of the commercial introduction of client/server, penetration of the manufacturing market was quite low, especially for smaller manufacturers (IDC, 1987a). Based on market size, the large manufacturer segment represented the main market. Based
on this low penetration, some vendors thought of client/server as a potentially disruptive technology. SAP, for example, initially saw the new technology as more attractive for smaller manufacturers, a market segment which they did not serve (Ambrosio, 1992). SAP thought that client/server would lower the cost of ownership and make these solutions more attractive to smaller firms. They believed that their core market, large manufacturers, would not be as interested initially because the new technology did not have as much functionality as the old.

However, the new use of client/server did not come from a new customer segment. Rather, it was the main market – large manufacturers – that changed its use. Large manufacturers were influenced by BPR and started to use software to configure new business processes across multiple functions. Paradoxically, the market that SAP originally thought was so attractive, the smaller manufacturers, continued to use client/server as they did host-based solutions. Such a change is not predicted by Christensen’s model because it assumes that each customer segment’s preferences remain relatively static. If we were to relax this assumption, Christensen’s model would then predict that incumbents that targeted large manufacturers should successfully transition through the disruption. Because these firms allocate their resources to this market, they should be able to recognize this change and make the appropriate adjustments. In fact, just the opposite occurred. Figure 5.2.4 compares the exit patterns of those vendors who targeted large manufacturers versus those who targeted smaller manufacturers. More vendors who targeted the large manufacturers exited (17 or 59% of all firms) between 1987

---

19 It is not clear that Christensen can do this and still preserve internal consistency.
and 1996 than those that targeted small manufacturers (14 or 20% of all firms) even though there were more than three times as many of these vendors.

This difference (represented in Figure 5.2.4) suggests that firm exit depended more on whether the customer’s usage pattern changed than on whether the software firms made the technological transition or allocated resources to the right customer segment.

However, there could be other differences between the two software vendor segments that help explain the difference in exit rates (see Appendix 5A for a discussion of significant differences between these two sub samples). Research has shown that younger and smaller firms have higher mortality rates than larger and older firms (Acs & Audretsch, 1987). The intuition is that smaller firms have fewer resources to respond to environmental
changes and competitive dynamics. This approach would predict that the vendors who targeted larger manufacturers were younger and smaller than those that targeted smaller manufacturers. However, the opposite is the case. In 1987, the average incumbent large manufacturer software vendor was 11 years old, compared to 7 years for those vendors who targeted smaller manufacturers.

The different mortality rates between the two segments could also be explained in terms of the differences in the customer segments. It is entirely possible that smaller manufacturers were later adopters of the new technology – in essence delaying the technological transition for those software vendors. This segment could have experienced less firm exit because the transition occurred at a later date. However, there is evidence that small manufacturers were also early adopters of new technology. Some of the earliest adopters of MRP, for instance, were small manufacturers (see Table 4.6.1 for a list). In addition, as previously mentioned, many thought that client/server would lower the cost of computing and make these solutions more attractive to smaller firms (Ambrosio, 1992). Surveys at the time also indicate a strong interest in client/server technology among smaller manufacturers (Gantz, 1992; Van Kirk, 1992). Therefore, there is no strong evidence that smaller manufacturers were later adopters of the technology.

Further analysis of the software firms reveals that their issue was not in producing the technological change, but in understanding how customers wanted to use the software. Further analysis of the software firms' actions reveals that they had trouble identifying the new usage patterns. Essentially, software firms who targeted large manufactures believed that the existing customer evaluation processes and uses would persist through the
technological change (Ambrosio, 1992). Advertising during this time period emphasized the traditional manufacturing integration found in MRP and MRP II systems as opposed to the new horizontal functional integration. In fact, a third party industry group, Gartner Group, not the software firms, coined the term, enterprise resource planning or ERP in 1992. Interestingly, in the same October issue in which Gartner advertises ERP as a new way of doing things, software vendors advertised their solutions from the perspective of the old use. Only after some additional time vendors began calling the new client/server software ERP and emphasizing the new use. In contrast, those firms that targeted smaller manufacturers did not face this issue. Because small manufacturers did not significantly change their use of the new technology, the transition was less disruptive for these vendors. Established firms are less likely to fail when customers use a new technology in similar ways to how they used the old technology.

However, those established firms that were able to identify the large manufacturer’s use pattern were able to successfully transition to client/server technology. Table 5.2.4 compares SAP, the eventual market leader, with Dun & Bradstreet software which exited the market in 1996. Table 5.2.4 shows that SAP and Dun & Bradstreet were similar in many respects – there were the same size and age, and they released their products in a similar time frame. The main difference is in their service strategy. SAP partnered with consultants, especially the big accounting firms like Andersen Consulting (now Accenture). It was these consultants that became heavily engaged in the BPR work and asserted themselves in the manufacturer’s purchase decision. SAP could leverage these

---

20 In some respects, SAP’s success is ironic, given that earlier they misread how the market would respond to the new technology. This suggests that SAP’s early success may have been more luck than strategy. But, SAP certainly has adapted since then to continue to grow its market share.
Table 5.2.4: Comparison of SAP with Dun & Bradstreet

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Successful Firm SAP</th>
<th>Unsuccessful Firm Dun &amp; Bradstreet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firm outcome</td>
<td>Current market leader in ERP</td>
<td>Exited Market in 1996</td>
</tr>
<tr>
<td>Founding</td>
<td>1972</td>
<td>1983 – through acquisition of McCormack &amp; Dodge which started in 1969</td>
</tr>
<tr>
<td>Size (1987)</td>
<td>Rev about 100M DM</td>
<td>Rev about $100M</td>
</tr>
<tr>
<td>Client/Server Release Date</td>
<td>1993/1994</td>
<td>1994</td>
</tr>
<tr>
<td>Strategic Action</td>
<td>Partnered with leading consulting firms in US</td>
<td>Acquired MSA (large mainframe software vendor) in 1990</td>
</tr>
<tr>
<td>Service Strategy</td>
<td>Traditional implementation support</td>
<td></td>
</tr>
</tbody>
</table>

Source: 10-Ks and analyst reports

partnerships to learn about the reengineering trend and help win the accounts. In contrast, Dun & Bradstreet was less accepting about the potential change in use. During the technological transition, they acquired one of the largest mainframe software vendors, MSA. From a technology perspective it does not make strategic sense that Dun & Bradstreet would purchase a solution that is architected on a technology that currently is being replaced unless they believed they were purchasing MSA’s large customer base. This rationale, however, eventually proved fatal. Part of the message of BPR was to get rid of the old systems and replace them with new ones that were cross-functional in nature. As a result, customers did not have the same allegiance to the installed software vendor as Dun & Bradstreet anticipated. In its service strategy, Dun & Bradstreet continued to focus
on its traditional implementation and educational support. They did not develop as strong a relationship with third party consultants as SAP. As a result, they were less aware of the changes in how manufacturers were changing their use of software. In many respects, successful migration through the technological transition depended more on the ability to anticipate and respond to the changes in customer use and less on whether the firm was established or new.

5.3 Radical incrementalism

Insights from the manufacturing software case can help build a more robust framework that better explains the competitive implications of technological change. The main contribution of this case is that how customers understand and use a technology is distinct, and at times asymmetrical from the production of the technology. Therefore, a fuller characterization of technological change should factor how customers use a technology as well as the production of the technology.

As such, this new framework builds upon Abernathy and Clark’s (1985) transilience map. The categorization schema introduces the commercial market as a distinct dimension from the technical characteristics of a technological innovation. Abernathy and Clark identify multiple characteristics of the commercial market – customer knowledge, channels of distribution, applications, modes of communication, and relationships with customer base – but they treat them as a whole and do not specify which dimension matters. In addition, their classification scheme is static. They place
technological innovations into one of these categories post hoc and then to derive the competitive implications from this categorization. But, as the history of the computer case in the insurance market suggests, at least initially both radical and incremental understandings of the computer and its adoption co-existed. The 1952 Actuarial Report noted that the committee had considered multiple uses, both more radical and more incremental, and recommended a use that had both incremental and radical aspects, but initially the incremental view dominated insurance firms' understanding, adoption, and use patterns. Only gradually, during the 1960s, did the market begin to shift toward the
recommended, relatively radical pattern. Similarly, on the production side, IBM and others made radically new computers like the UNIVAC and IBM’s 700 series, but they also made more incremental products that shared computer and tabulating equipment features, like IBM’s 650. By focusing on the end point of the change, current classification schemes miss the dynamic element of technological change. Change is not simply what changes, but also how it changes (Barnett & Carroll, 1995).

In contrast, identifying how customers actually use the technology as the salient feature of the commercial market introduces a dynamic element to this model. As the manufacturing case shows, over time, shifts may occur in how the customer understands and uses the technology. A more robust framework to characterize technological change captures both the nuances of the customer use perspective and its dynamic nature, as shown in Figure 5.3.1. A product technology change can be either radical or incremental based upon the magnitude of the change along economic, technical, and organizational dimensions currently identified in the literature (see Figure 5.2.3 for a summary). As we have seen from the cases, a radical change in use is when customers develop a new understanding, justification, and deployment for the technology, e.g., the introduction of MRP uses of planning software. An incremental change in use is when customers maintain the same understanding and justification but alter how they deploy the technology, e.g., manufacturers change how they schedule within the MRP application. There arrows on either side depict that a technology or use is dynamic – what starts out as incremental could be become radical (or vice versa).
When most customers at a particular time also interpret these technologies significantly differently than pre-existing technology and use them in new ways, the technological change may be classified as radical/radical. Examples of such technological changes include the introduction of the Model T, which enabled driving on rural roads (Abernathy, 1978; Abernathy & Clark, 1985), and of 8 inch disk drives, which allowed for more portable means of transferring data (Christensen & Rosenbloom, 1995). In contrast, incremental/incremental technological changes are those that from the producer perspective generate products which substitute for existing products and/or leverage existing technology, and that the customers interpret and use based upon the constraints of the pre-existing technologies. An example of incremental/incremental technological change is the evolution from 78 RPM to 33 RPM in the record industry.

The off-diagonal quadrants of Figure 5.3.1 are perhaps the more interesting categories. Incremental/radical technological changes are those that are incremental from the producer’s perspective but that users interpret and/or use in a radically different way. Developing the Sony Walkman was incremental in the sense that it combined pre-existing components that did not require substantial new architectural knowledge. Users, however, developed entirely new mobile uses of listening and interacting with music. In contrast, radical/incremental changes are those that are radical for the producers of the technology but that customers interpret and use in an incremental, not radical, way. Bresnahan and Greenstein’s (1996) example of the transition from mainframes to client/server technology represents such a radical/incremental combination. From the producer perspective, developing client/server hardware was a radical change in the sense that it required new
components, especially new software, as well as new skills in networking and
asynchronous process management. However, Bresnahan and Greenstein observed that the
customers deployed these new solutions incrementally by integrating with the existing
mainframe hardware and slowly migrating existing applications.

This framework also exposes some of the biases in the current literature. Typical
categorizations of the competitive significance of technological change only identify one
of the dimensions. Those that are production oriented characterize technological change
along the technology dimension (again see Figure 5.2.3); whereas, those that are market
oriented, for example, Christensen and Adner, focus on the use dimension. Consequently,
each only sees the radical/radical and incremental/incremental changes. But, what we have
seen from the historical cases is that the more interesting and perhaps more challenging
types of changes are when the changes are asymmetrical.

To make this more apparent, Figure 5.3.2 maps the changes from the
manufacturing case and the industry dynamic outcome to this classification scheme. The
radical shift in customer use from the practical approach to MRP in the 1960s and 1970s
without corresponding radical technological changes induced significant new firm entry.
Entry continued as incremental changes in both the technology and use continued in the
1980s. Different customer segments had different reactions to the radically new
client/server technology. Small manufacturers did not significantly change their use, but
large manufacturers did. Consequently, more incumbents that served the large
manufacturers exited.
Figure 5.3.2: Different changes and outcomes in the U.S. manufacturing planning market

<table>
<thead>
<tr>
<th>Product Technology Change</th>
<th>Customer Use Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radical</td>
<td>Incremental</td>
</tr>
<tr>
<td>Introduced decision making software in the 1960/70s</td>
<td>Expanded use in 1980s</td>
</tr>
<tr>
<td>Outcome: New firm entry</td>
<td>Outcome: Continued entry</td>
</tr>
<tr>
<td>1990s: Large manufacturers changed use with new technology</td>
<td>1990s: Small manufacturers maintained pre-existing use despite client/server</td>
</tr>
<tr>
<td>Outcome: Low incumbent firm survival</td>
<td>Outcome: Higher incumbent firm survival</td>
</tr>
</tbody>
</table>

More generally, we may assert that a radical change in customer usage patterns induces new firm entry because the change in use represents new needs and requirements. In addition, when faced with radically new technologies, customers may adopt conservative uses that replicate older technologies, as insurance companies did with computers. In these circumstances, to the extent that incumbent firms can adopt the new technology, we would expect them to have a greater chance of survival because they best understand existing use patterns. We may assert that established firms are less likely to fail when customers use a new technology in similar ways to how they used the old
technology. These insights help explain the initial anomaly – why established firms may persist through significant technological changes. In some cases, the issue seems to be due to a misspecification of the technological change. The change was not radical, but radical and incremental. It was radical for the producers but incremental for the users. And, creative destruction can occur without a radical technological change. Customers could change how they use an existing technology.

However, I must caution from extrapolating too much from a historical case. Further hypothesis generation and testing of this framework with different kinds of technologies and time period is needed.
Appendix A of Chapter 5: Coding of entry/exit patterns

To supplement the historical analysis, I generated the entry/exit model for planning software from 1954 – 2003, while focusing on the period between 1954 and 1996. Properly identifying the software vendors was challenging as there is not single source of information identifying these firms and in many cases, for example, planning software was a division within a broader firm. I used several sources to identify a complete list as possible. Starting in 1967, ICP began publishing software directories that included manufacturing software. Other sources of data include: IDC reports about the manufacturing software industry from 1984-1994, AMR reports on the industry from 1994-present, and The Software 500 list from Software Magazine in the years 1989 – 2002. In order to increase the number of manufacturing oriented firms, I also identified enterprise software firms from the Manufacturing Systems 50 from 1991-1997 from the magazine Manufacturing Systems. This generated a list of 269 firms. I eliminated some firms because through cross comparison of the list, these firms did not participate within the specific market of interest. Because many of these firms were small and private, I was unable to locate many of the founding and exit dates. Eliminating these firms, the total number of firms consists of 137 of the original 269 firms.

From the historical work, it is clear that this data set does not capture the entire market. Given the low cost of entry, the market includes many small private firms that are not included on the list. We know from market size data from IDC that individually these firms did not comprise any more than %0.01 of the market. Given their size, these firms
tended to target smaller manufacturers, so there is a concern if these omitted firms acted in a systematic way that is somehow different from those firms who targeted small manufacturers included within the sample. To address this issue, I have tried to identify some firms not included in my list by using local directories to see if the entry/exit activities appear to be any different. Thus far, there does not appear to be a significant difference, however, the sample is small. In fact, many of these small local firms appear to persist for long periods of time surviving off of providing services to local clients. In addition, the entry/exit pattern is consistent with contemporary industry analyst’s assessments of industry changes. For instance, when IDC identifies a slow down market entry corresponds to a slowdown in market entry in entry/exit model.

Like any industry study, the coding of entry and exit also has its own set of issues. I coded entry and exit along the guidelines as discussed in Carroll and Hannan (2000). The primary form of exit in the planning software industry is through merger. Acquisitions make it difficult to determine if the firm exited because of failure or success. For example, Oracle’s recent acquisition of PeopleSoft appears to have occurred against the wishes of PeopleSoft. In addition, some firms like IBM and Accenture exited the market to concentrate on other core business activities. Another more recent trend is private equity firms buying software companies to create a portfolio of solutions. In many cases, these acquisitions simply reflected a change in ownership and operationally the software firms remained the same. Therefore, I coded an exit when a firm left the market either through bankruptcy or through acquisition by another firm. Typically, after an acquisition the management team leaves and the products get integrated. I did not code privatization or
buy outs as an exit because typically this represents a change in ownership. If a firm was acquired by another firm not on the list, I added the acquiring firm as an entrant. Based on these criteria, 37% of all the firms exited the sample before 1997.

I further divided the sample by target customer segment, coding each firm as targeting either large or small manufacturers. To code this variable, I used information about target customers from the software vendors’ 10-Ks and annual reports as well as market segmentation from industry analysts such as IDC and AMR. In most cases, firms targeted either large or small manufacturers; however, some firms, such as SAP and PeopleSoft, targeted both large and small firms. In these cases, I used additional information about number of customers and revenue as well as classification from industry analysts to determine the appropriate target customer segment. For example, SAP generates most of its revenue from larger customers so it is classified as targeting large manufacturers, even though it has a sizable client list of smaller manufacturers. During this time period, no firm switched target customer segment. Of the 137 firms in the sample, 43 target large manufacturers and 94 target small manufacturers.

There are some interesting and important differences between these two groups of firms. Those firms that targeted larger manufacturers entered the market earlier, the average entry year was 1976 compared to 1982 for smaller manufacturers. In addition, entry into the manufacturing software market tended to represent a diversification for the vendors who targeted large manufacturers. 42% were diversified entrants compared to 14% of those firms who targeted smaller firms. For example, IBM expanded from hardware; Accenture, from consulting, and Oracle, from database management software.
As a result, those firms who targeted larger manufacturers also tended to be larger
(Because many of these firms were diversified, there is not consistent firm level revenue
data). Lastly, as discussed in more detail in the chapter, the mortality of the different
groups is different. 70% of those that targeted large manufacturers exited the market
compared to 56% in the small manufacturing segment.
Chapter Six

Conclusion

6.1 Implications for theory and research

The intent of this research was to more explicitly consider what is often implicit in models of technology and industry evolution – the customer’s understanding, experiences, and uses of the technology. Recognizing that customers evolve in their understanding and use of a technology in ways that are related to but independent of technological change has theoretical implications for broader topics in technology strategy: the relationship between industry-level selection and firm-level adoption, diffusion, and technology standardization. I shall address each in turn.

6.1.1 Industry-level selection and firm-level adaptation

Historically, explanations of firm evolution in the organizational literature and the sources of competitive advantage in the strategy literature have been divided between two explanations: those that argue that heterogeneous firms with different levels of capabilities must adapt to changing environment conditions (Peteraf, 1993; Scott, 1995; Teece, Pisano,
and those that argue that environmental conditions *select* new types of firms (Hannan & Freeman, 1977). By showing how one environmental participant, the customer, evolves in ways that are interrelated to firms innovating, this research argues that the source of competitive advantage lies in the *interplay* between firms adapting and the environment selecting. A firm’s innovative activities affect how customers evaluate and use a technology, but customers also change expectations based upon continued use of the technology, affecting the firm’s activities. Environmental selection and firm-level adaptation co-evolve.

Considering endogenous changes in the environment suggests that the pattern of industry change may not be evolutionary. A core feature of evolutionary industry models is that they essentially are two stage, where the first stage focuses on production innovation and the second, process, implying that the shift to cost-based competition is irrevocable (Agarwal, Sarkar, & Echambadi, 2002). The technical standard is retained because firms focus on process innovation and on knowledge and resources around the technical standard. Product innovation does continue, but firms only make incremental improvements to the standard (Utterback & Abernathy, 1975). To make this claim, these models assume that the convergence in use is also irreversible, but this is only the case under certain conditions. There is an inherent propensity toward variation in customer use through local differences in deployment and experiential learning, even after use has converged. Only in cases where this additional learning reinforces existing dominant use will the shift to process innovation remain irreversible. But, in many cases, customer learning does not reinforce the standard. For example, Zbaracki (1998) observed the de-
stabilization of TQM when customers learned that the standardized offering's reality did not match its proponents' rhetoric. In these cases where standardization breaks down as customers learn new uses, the industry's competitive dynamics shift back to product differentiation and innovation to meet the new customer needs. Therefore, unlike the evolutionary explanation, this model is not restricted to two stages of industry evolution, but instead allows for multiple periods of standardization and de-standardization (McGahan, 2004).

Lastly, much of the dissertation has focused on this effect at the industry-level, but this insight can be applied to the analysis of the development of firm-level capabilities. The capabilities view of the firm recognizes the need for firms to develop new capabilities in ever changing environments (Teece et al., 1997). This literature has presumed that the locus of this renewal lies inside the firm and that in technology-intensive industries product innovation is a primary source of this renewal (Danneels, 2002; Dougherty, 1992; Helfat & Raubitschek, 2000). This research suggests that the locus of organizational renewal need not lie inside the firm. Customers, just like firms, have capabilities, which develop and change in relation to a firm's capabilities. Firms can leverage these capabilities in an effort to renew their own capabilities. Understanding the rate of learning within a customer population and the broad incentive structure to reveal current uses can help us understand the conditions under which not dynamically renewing product innovation capabilities within a firm makes strategic sense.
6.1.2 Diffusion

Traditional diffusion models assume that the customers’ understanding of a technology does not change during the adoption process. Instead, these models focus on access to information about the new product, the influence of other customers, and economic incentives to help explain the rate and extent of adoption of a new technology (see (Geroski, 2000) for a review). In essence, they characterize diffusion as the purchase decision the customer makes, while ignoring that customers typically buy a technology to use it. Consequently, the process of implementing the technology and its effects do not factor into the explanation (Rogers, 2003).

This work recognizes that customers are not just purchasers, but also users. And, as users, they learn new things about a technology that changes their understanding and evaluation of a technology. Manufacturers shift their understanding of planning software from “a record keeping tool to reduce costs” to “a decision-making tool to improve profits”. It is not safe to assume that the next customer understands the technology in the same way as the first customer. Knowledge is not just information about the product, but an interpretation of what the technology is and should do. Consequently, the word, “diffusion” is misleading to describe the commercialization process of a new technology. Customers consume and use a technology which in turns changes what that product is.

Explaining the rate and direction of the commercialization process requires paying more attention to the patterns of use. Understanding how a dominant use is determined can help explain the typical S-curve pattern. It seems that the setting of a dominant use will
lead to higher rates of adoption. But, a dominant use is not eternal; customers can learn new things that destabilize the dominant use. Factoring in use, in turn, may help explain why customers sometimes abandon a technology, leading to a more robust theory of adoption.

6.1.3 Technological standardization

Technology standardization is an important event in technological development and industry evolution, but the mechanisms that explain when standardization occurs are not well specified. Currently in the literature, there are essentially two different views. One view focuses on the social-political processes that establish standards (Nelson, 1995; Rosenkopf & Tushman, 1998; Van de Ven & Garud, 1989); whereas, the other describes standardization in terms of the spillover effects from increasing returns to adoption (David, 1985; Silverberg, Dosi, & Orsenigo, 1988). Considering how customers use a technology adds a new dimension, suggesting that standardization is not just about the technology, but may also include customers converging to a specific use.

Factoring in customer use can help integrate these two previous views. Because customers learn different things about the technology from different uses, adoption of the technology in general does not generate the spillover. Rather, it is the adoption of similar uses. The participation of institutions in shaping how customers categorize is one way that
customers begin to use the technology in similar ways to help generate these spillover effects.

Market mediators, such as APICS and industry analysts, played a significant role in developing a collective understanding of manufacturing software that helped standardize use. Prior to the introduction of MRP, many manufacturing firms had ad-hoc and poorly documented production and control procedures. APICS' "MRP Crusade" and professional testing around MRP principles helped establish MRP as the standard way manufacturers should manage their production processes. Standardizing on the function helped form a collective understanding of what this software was supposed to do, which was reinforced by establishing certification standards for professionals and other third party educational groups incorporating the MPR standard to evaluate products.

Just as institutions help standardize the verification and performance criteria to develop the new technology (Van de Ven & Garud, 1993), a related group of institutions influence what comparisons and performance criteria customers converge upon to form a collective understanding of the product. These institutions differ from the kinds of organizations involved in setting the technical standards and governmental lobbying. They include organizations that represent customer interests such as trade and professional associations and customer-facing mediators such as consultants and industry analysts. The collective understanding formed through the interactions of these groups helps legitimatize a particular variant of use, in this case MRP.

Some have described this process by appealing to social movement theory – an inherently political process in which entrepreneurs compete with other views to mobilize
resources (Rao, Morrill, & Zald, 2000). There certainly were elements of social movement in the manufacturing case, especially APICS’ “MRP crusade”. This research highlights the opportunity to integrate social movement theory with the technology management literature to provide a more complete explanation of the emergence and diffusion of innovations (Hargrave & Van de Ven, 2006)

6.2 Managerial implications

To identify customers as users is to recognize that, like producers, customers have capabilities in relation to deploying and using a technology. Also like producers, customers' capabilities are dynamic— they learn about the technology and develop new uses for it. This implies that firms do not have to internally develop innovation capabilities, but can instead leverage customer capabilities by encouraging them to develop new uses for their technologies. Research in user-led innovation has shown that a special group of users, called “lead users,” are a valuable source of product innovation for firms (Von Hippel, 2005). In this study, IBM used elements of this strategy by hiring several “lead users” of early MRP applications to educate other customers and to develop its own product. Developing strategies and design elements that encourage users to develop new uses and then feeding that information back into the firm may be a way for firms to renew their own capabilities. For example, opting to send information back to Microsoft when a program crashes gives Microsoft information about how customers are using its software. Further integration of the user-based innovation literature should provide some additional
insight for managers. For instance, it is still not clear how beyond lead users more ordinary customers develop capabilities through using the technology (Von Hippel, 1986).

In addition, functional areas outside of product development and design, such as marketing, finance, and services, influence how customers use technology. How a customer must pay for a product over the lifetime of its contract and how a customer receives service influence how she uses the technology (Rosenberg, 1983). These kinds of customer-facing capabilities have traditionally been viewed as complements to product development and design. There is an implicit primacy given to products as these customer-facing capabilities are supposed to support product innovation (Teece, 1986). This work shows that this primacy may be misplaced. Firms can differentiate themselves without directly engaging in product innovation. Innovating in services and finance can help generate new customer uses which in turn feeds back into product innovation. Thus, there are other sources of innovation beyond the development and production of new technologies that can renew firm capabilities and affect industry evolution (Cusumano, Suarez, & Kahl, 2007).

To illustrate this effect consider how Salesforce.Com leveraged a different pricing model to gain entry into the well established customer relationship management market (CRM – software that targets sales and marketing functions within a firm). When Salesforce.Com entered this market in 1999, the CRM market was well established with clear market leaders like Siebel. Many of the established firms were well on their way incorporating new Internet technology into their solutions. Salesforce.Com did not enter the market with a new technological innovation or even new functionality. Rather, it
entered the market by pricing the software in a new way. At the time, CRM was typically sold at an enterprise level using a convention term license agreement. Customers would pay for the right to use the software upfront based upon some metric of size and number of users. The customer would pay for the software whether they used to it or not (see Siebel’s 10-Ks for a description of this model). In contrast, Salesforce.Com used a transaction-based license model in which users paid for the software based upon how many transactions they ran through the system. Salesforce.com also delivered the software in a new way – it hosted the application such that the user accessed it remotely. This reduced the amount of technical integration required to get the software to work at the user’s site. Based on this pricing model and delivery option, Salesforce.com targeted individual sales people as opposed to the executives who typically made the decision to buy CRM software. This software was used differently – at the individual level – than traditional CRM applications. Today, Salesforce.Com remains one of the few growing independent CRM vendors in the market.

Competitive differentiation can be achieved through innovation in other areas of the firm outside of product and process innovation. But, continuous investment into these activities is necessary.

6.3 Conclusion

At is core, this research is about change – how it happens and what its competitive implications are. By analyzing the history of how customers used a particular technology, this research has tried to make more explicit how the market in which organizations
compete can change. The history of an industry from a customer perspective shows that customers can endogenously change their needs and evaluation criteria for a technology simply through continued experience using the technology. Moreover, they can change in complex ways and across multiple dimensions. Cognitively, they may generate new interpretations of what the technology is, or they may change how they justify its use. Or, the market structure through which customers consume the technology may change. New trusted advisors and educators may generate different relationships with customers than previous intermediaries. A lot of subtle, but substantial changes, can take place in the “consumption junction”; consequently, we should not model the environment as a relatively stable place that only changes through exogenous events.

By invoking the historical method, this research has also tried to bring the process element of technological change to the forefront. Change is not just difference in content as most of the technology literature implicitly assumes. It is also a process (Barnett & Carroll, 1995). There are many interesting features of change as a process that have not been adequately explored, such as How long does change take? And, what is it shape (e.g., linear, cyclical, branching)? Focusing on the process of technological change also brings forth some unexplained issues. This view emphasizes the paradoxical combination of continuity and difference during technological change, which raises questions about how firms should identify themselves throughout the change process.²¹ The strategic dilemma

²¹ Incidentally, this paradox forms the heart of philosophical inquiry into the nature of change since at least the pre-Socratics. For philosophers, what makes change so perplexing is trying to explain how the same object can change over time and still remain the same thing. Unfortunately, I am getting more rotund as I grow older, but it is still me. For philosophers, a core issue in change is maintaining identity through change. Management scholars could benefit from a more comprehensive understanding of how philosophers think about and resolve these metaphysical and epistemological issues surrounding change.
is how to manage the contradictory notions of maintaining continuity and achieving
discontinuity. This issue seems particularly appropriate for modern times. In some sense,
technical innovation is not the hard part today, especially as the locus of innovation shifts
from the firm to a more open community (Chesbrough, 2003; Von Hippel, 2005). But,
what remains challenging is understanding how customers want to use the new technology
and providing a coherent identity to customers throughout the change process.
References


-224-


