

The Assimilation and Diffusion of Software Process Innovations

by

Robert G. Fichman

B.S.E, Industrial and Operations Engineering,
University of Michigan, 1982

M.S.E., Industrial and Operations Engineering,
University of Michigan, 1983

Submitted to the Alfred P. Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Management
at the
Massachusetts Institute of Technology
January 1995

© 1995 Massachusetts Institute of Technology. All rights reserved.

Signature of Author _____
Department of Management Science
January 1995

Certified by _____
Chris F. Kemerer
Douglas Drane Associate Professor of Information Technology and
Management
Dissertation Committee Chair

Accepted by _____
Birger Wernerfelt
Chairman, Ph.D. Committee
Sloan School of Management

ARCHIVES
MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

The Assimilation and Diffusion of Software Process Innovations

By

Robert G. Fichman

Submitted to the Alfred P. Sloan School of Management
in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in Management

ABSTRACT

This dissertation examines the factors influencing the assimilation and diffusion of software process innovations (SPIs). It is argued that SPIs possess two distinctive characteristics—*increasing returns* to adoption and *knowledge barriers* impeding adoption—that hold important theoretical implications. A disk-based survey was administered to over 600 IT managers to gather data on four SPIs: object-oriented programming languages (OOPLs), relational database management systems (RDBs), fourth generation languages (4GLs), and Computer Aided Software Engineering (CASE) tools. Gathered data were used to support three studies.

The first study tested a model of OOPL assimilation derived from Attewell's theory of the diffusion of complex organizational technologies. It was hypothesized that organizations with a higher propensity to innovate with SPIs would be distinguished by: 1) greater "learning-related scale," 2) more extensive pre-existing knowledge in areas related to OOPLs, and 3) greater diversity of knowledge and activities in general. A regression analysis incorporating the independent constructs and six control variables strongly supported all three hypotheses.

The second study provided an analysis of six alternative measures of the organizational innovativeness concept: Time of Adoption, Dichotomous Adoption, Aggregated Adoption, Diffusion, Infusion, and Assimilation Stage. Adoption data for OOPLs, RDBs and CASE were used to explore several empirical issues. Major results included: 1) richer measures do not necessarily lead to more strongly predictive models; 2) positive associations exist between the propensity to acquire many innovations, to acquire any given innovation early, and to implement innovations in depth; 3) models using Aggregated Adoption have the greatest variance explained; and 4) Diffusion and Infusion are especially difficult to predict.

The third study introduced the "assimilation gap" concept, defining it as the difference between the cumulative adoption curves associated with two alternative definitions of "adoption": adoption-as-acquisition and adoption-

as-deployment. An analysis of the assimilation gaps was performed for RDBs, 4GLs and CASE. As expected, moderate assimilation gaps were found for RDBs and 4GLs, and a very pronounced gap was found for CASE. Further analysis based on survival analysis techniques showed that the size of the gaps were significantly different.

Thesis Committee: Professor Chris F. Kemerer (Chair)
Professor John Rockart
Professor Stephan Schrader
Professor N. Venkatraman

ACKNOWLEDGMENTS

A junior professor once told me that the hard part about finishing a Ph.D. is not deciding which dissertation you are going to do, but figuring out all the ones you aren't. I think he meant doctoral students usually have no shortage of research ideas that appear quite promising at first glance; the challenge lies in sorting out the ones that actually are at least slightly promising. I can hardly imagine anyone being more skilled in assisting in this sorting process than my advisor, Chris Kemerer, has been. Only in retrospect do I realize how many times I teetered on the edge of a tar-pit, only to be reeled back in by few pointed questions and comments from Chris. Once settled into a viable topic, Chris continued to provide the kind of invaluable support he's given throughout my time at MIT—by helping me to sharpen my focus and position my research, and in general, by serving as a role model for the highest standards of scholarship.

My other committee members also provided able guidance and encouragement. I want to thank Jack Rockart for always pressing me to relate my work—and indeed to do the kind of work that can be directly related—to the actual practice of management. I'm grateful to Stephan Schrader for his willingness and ability to contend with a sea of unfamiliar IT terminology and concepts, and still come up with just the right suggestions at just the right times. Finally, I want to thank Venkat Venkatraman for his vast knowledge of pertinent research methodology; for his ability to clearly articulate the current standards for research; and perhaps most importantly, for his detailed suggestions on how such standards could feasibly be incorporated into my dissertation. On more than one occasion, a half hour with Venkat spared me a week of wild-goose chasing in the library.

Many others at MIT also helped me get through the long march of doing a dissertation. The Center for Information Systems Research provided generous funding throughout my tenure at MIT, and much more besides. Judith Quillard and Debra Hofman, in addition to providing thoughtful feedback on my work on many occasions, were always a joy just to talk to, whether over lunch, at CISR summer session, or at the yearly ICIS conferences. Professors Erik Brynjolfsson and Wanda Orlikowski taught doctoral seminars that helped make sense of two large swaths of the IT field, and were always available to discuss ideas or work in progress. My fellow students—especially Shyam Chidamber, Liz Davidson, Mike Gallivan and Henry Kon—contributed immeasurably to making my MIT years not only bearable but even, for the most part, enjoyable. I hope that my future academic endeavors are accompanied by the same spirit of friendship and camaraderie.

Most of all, I want to thank my wife, Susan Harrington, for her love, friendship, and unfailing support, and for making our last four years together the best ever, even in spite of the demands and austerity of student life.

This dissertation is dedicated to my wife Susan Harrington, and our beautiful baby daughter, Claire Noelle Fichman.

THE ASSIMILATION AND DIFFUSION OF SOFTWARE PROCESS INNOVATIONS

Abstract	2
Acknowledgments.....	4
Author's Introduction.....	9
Chapter 1. Distinctive Characteristics of Software Process Innovations and Implications for Theory	15
1.1 Introduction.....	15
1.2 The Classical Diffusion Model	17
1.3 Increasing Returns	21
1.4 Knowledge Barriers.....	24
1.5 Toward a Theory of the Assimilation and Diffusion of SPIs	26
1.6 Summary	30
Chapter 2. An Organizational Learning-Based Model of Software Process Technology Assimilation	33
2.1 Related Theory.....	35
2.2 Proposed Model.....	37
2.2.1 Learning-Related Scale.....	38
2.2.2 Related Knowledge	39
2.2.3 Diversity.....	41
2.3 Discussion.....	42
Chapter 3. An Empirical Study of the Assimilation of Object-Oriented Programming Languages.....	46
3.1 Overview of Object-Oriented Programming Languages.....	47
3.2 Potential Validity Threats	48
3.3 Data Collection.....	49
3.3.1 Disk-Based Surveys (DBSs)	50
3.3.2 Survey Instrument Development Procedure	54
3.3.3 Creation of Survey Sample	56
3.3.4 Survey Administration Procedure	60
3.3.5 Characteristics of Responding Sites and Informants	61
3.3.6 Analysis of Response Bias.....	66
3.3.7 Results of Telephone Survey of Non-Respondents	69
3.3.8 Method Bias Follow-up Survey	71
3.4 Measurement.....	72
3.4.1 Assimilation Stage	72
3.4.2 Learning-Related Scale.....	76
3.4.3 Related Knowledge	78
3.4.4 Diversity of Knowledge and Activities	81
3.4.5 Control Variables.....	83
3.4.6 Quantitative Analysis of Construct Validity	84
3.5 Results.....	92
3.5.1 Descriptive Data on OOPL Assimilation	92
3.5.2 Antecedents of OOPL Assimilation.....	98

3.6	Discussion.....	106
3.6.1	Summary of Results	106
3.6.2	Potential Limitations	109
3.6.3	Generalizability of Results.....	110
3.6.4	Implications and Contributions.....	111
3.6.5	Future Work	114
	Exhibits	116
Chapter 4.	Alternative Measures of Organizational Innovativeness: A Conceptual Analysis	128
4.1	Time of Adoption	129
4.2	Dichotomous Adoption.....	132
4.3	Aggregated Adoption	133
4.4	Extent of Implementation	137
4.4.1	Diffusion	138
4.4.2	Infusion.....	141
4.5	Assimilation Stage	144
4.6	Summary	148
Chapter 5.	Alternative Measures of Organizational Innovativeness: An Empirical Analysis	151
5.1	Predictor Variables	152
5.2	Empirical Analysis.....	153
5.3	Conclusions.....	163
Chapter 6.	The Illusory Diffusion of Innovation: An Exploratory Examination of Assimilation Gaps.....	165
6.1	Introduction.....	165
6.2	Prior Research	169
6.2	The Assimilation Gap Concept	171
6.3	An Empirical Study of Assimilation Gaps For RDBs, 4GLs and CASE	173
6.3.1	Estimating the Assimilation Gap.....	175
6.3.2	Study Methods.....	180
6.3.3	Results.....	183
6.4	Organizational Level Outcomes Contributing to the Assimilation Gap.....	193
6.5	Discussion and Conclusions.....	198
	Appendix 6A: Assimilation Gap Estimation Based on Diffusion Modeling Techniques.....	201
	Appendix 6B: Sensitivity Analysis for SPI Survivor Functions.....	205
Chapter 7.	Summary and Conclusions.....	207
7.1	Summary of the Research.....	207
7.2	Implications and Contributions.....	214
7.3	Future Work	219
7.3.1	Potential Studies Building Directly on Dissertation Results.....	220
7.3.2	The Role of Supply Side Institutions	222
7.3.3	Comparative Case Studies of SPI Deployment.....	224

Appendix A. Mini Case Studies of Early Adopters of Object Technologies	227
A1. Research Methods	228
A2. Synopsis of Case Sites	230
A2.1 EnergyCo E&P	230
A2.2 BankCo UK	233
A2.3 FinancialCo RMS	235
A2.4 BrokerCo UK	238
A2.5 Summary	241
A3. Case Study Analysis: Major Themes	242
A3.1 Nature of the Knowledge Burden Imposed by OOPL Assimilation	242
A3.2 Assimilation Strategies	253
A4. Summary and Conclusions	259
 Appendix B. OOPL DBS Screens	 261
 References	 314

AUTHOR'S INTRODUCTION

Over the past two decades, an impressive array of software process technologies have been invented. Relational databases, fourth-generation languages, CASE tools, and object technologies are just a few of the major innovations in software process technology to be commercialized since 1980 (see Table 1). These technologies are termed *software process innovations* (SPIs), because when acquired and deployed, they produce a significant change to an information technology (IT) group's process for developing software applications. While this thesis concentrates on SPIs embodied in software, purely procedural innovations—such as joint applications design or structured methodologies—also qualify as SPIs.

Table 1: Prominent Software Process Innovations

- Object technologies
- CASE tools
- Expert system shells
- Model-based development (templates)
- Rapid prototyping
- Fourth generation languages
- Large scale component reuse
- Joint applications design (JAD)
- Code generators
- Formal methods
- Relational database management systems
- Information engineering

While many SPIs have become available, current evidence suggests that IT departments face substantial challenges in adopting and deploying them. The risk of adoption failure is high, and the possibility of ending up committed to a "stranded" technology creates an additional peril [Stamps 1989]. Relatedly, SPIs typically diffuse across the IT industry more slowly than proponents originally expected—or might have been desirable given the potential these technologies hold to improve software development [Yourdon 1986; Bulkeley

1990; Howard and Rai 1993; Slitz and Mintz 1993]. This incongruity between invention and diffusion stands in contrast to other domains, such as hardware and packaged software, where frequent invention has been matched by robust diffusion.

This thesis seeks to foster an improved understanding of the factors affecting SPI assimilation and diffusion. Following Meyer and Goes, assimilation is defined here as the process spanning from an organization's first awareness of innovation to, potentially, acquisition and widespread deployment [1988]. Adoption, by contrast, is a single event in the assimilation process, typically acquisition or a commitment to use the innovation. Diffusion is the macro process by which innovations spread across a population of potential adopters. The goal of this research is to help both potential adopters and suppliers become more effective innovators. Potential adopters need guidance in the complex process of evaluating SPIs and in formulating an appropriate assimilation strategy. Suppliers need tools to assist them in identifying likely early adopters, understanding the special challenges these adopters face, and constructing deployment strategies that sustain diffusion of technologies over the mass market of later adopters.

It has become increasingly clear that no single theory of innovation diffusion suitable to all innovations and contexts is likely to emerge [Downs and Mohr 1976]. The approach taken here is to narrow the focus to more specific innovations and contexts, and to develop theory centered around the distinctive characteristics of those innovations and contexts. More specifically, this thesis will argue that SPIs are distinguished by two characteristics—strongly *increasing returns* to adoption [Arthur 1988] and substantial *knowledge barriers* impeding adoption [Attewell 1992]. These characteristics imply a need to reconceptualize both the micro-level determinants of adoption and the macro-level patterns of diffusion. Recent research in the economics of standardization [Katz and Shapiro 1986; Farrell and Saloner 1987; Arthur 1988] and organizational learning [Cohen and

Levinthal 1990; Attewell 1992] provide the base upon which such a reconceptualization can be built.

The primary research questions addressed by this thesis are:

- 1) What factors predispose IT departments to be early and sustained assimilators of SPIs? (Addressed in Chapters 2 and 3.)
- 1) How do alternative measures of the organizational innovativeness concept compare, and how and where are they best applied? (Addressed in Chapters 4 and 5.)
- 3) What patterns of diffusion do SPIs typically follow, and do these patterns conform to those suggested by their distinctive characteristics? (Addressed in Chapter 6.)

Technological innovation is the primary driver of improvements in industrial productivity [Nasbeth and Ray 1974]. Yet, if promising inventions cannot be widely deployed, then clearly the benefits resulting from their invention will be curtailed. Many have commented on the disappointingly slow rate of progress of software practice in the United States [Brooks 1987; Abdel-Hamid and Madnick 1989]. While much of the so-called software crisis is likely attributable to undisciplined application of existing technologies and inattention to measurement and continuous improvement [Kemerer 1989], incremental improvement can go only so far; at some point, more radical changes are needed to provide the base upon which further incremental improvements can be built [Sahal 1985; Foster 1986; Tushman and Anderson 1986].

Document Overview

Chapter 1 provides a brief review of the classical diffusion model and associated criticisms, and then develops a macro-level model of the diffusion of SPIs centered on their distinctive characteristics (i.e., increasing returns and knowledge barriers).

Chapter 2 presents a detailed examination of the relationship (first introduced in Chapter 1) between knowledge barriers and innovation, and based on this examination, develops an organizational learning-based model explaining differences in the propensity of organizations to initiate and sustain innovations in software process technology. Specifically, it is proposed that organizations with a higher propensity to innovate with these technologies are distinguished by three characteristics: 1) greater scale of activities over which learning costs can be amortized ("learning-related scale"), 2) more extensive pre-existing knowledge in areas related to the innovation, and 3) greater diversity of knowledge and activities in general. Such organizations, it is argued, should be better able to amortize learning costs, should be better able to assimilate any given amount of information related to SPIs, and should have a lesser burden of organizational learning in the first place.

Chapter 3 describes an empirical test of the organizational learning-based model (presented in Chapter 2), using data collected from over 600 IT departments on the assimilation of object-oriented programming languages (OOPLs). Data were collected using a disk-based survey mode, a novel technique whereby respondents insert a disk into their PCs and are then lead through questionnaire items automatically. The chapter begins with a brief description of object-oriented programming, followed by a presentation of the research design and data collection procedures, data analysis, and interpretation of results.

Chapter 4 provides a conceptual analysis of six prominent measures of the organizational innovativeness concept, including: 1) Time of Adoption, 2) Dichotomous Adoption, 3) Aggregated Adoption, 4) Diffusion, 5) Infusion, and 6) Assimilation Stage. For each measure, the analysis includes a description of the origins of the measure, potential advantages and limitations, suggested operationalizations for the case of software process

innovations, and guidance on when and how the measure is likely best applied.

Chapter 5 provides an empirical complement to the conceptual analysis of alternative innovativeness measures presented in Chapter 4. Specifically, data on three software process innovations—relational database management systems (RDBs), computer-aided software engineering (CASE), and object-oriented programming languages (OOPLs)—are used to support an empirical analysis of four issues researchers should consider when choosing among the alternative innovativeness measures.

Chapter 6 presents an empirical analysis of one of the propositions developed in Chapter 1, namely, that SPIs will be prone to an "assimilation gap". It develops the assimilation gap concept, demonstrates that such gaps can be sensibly measured, and identifies statistical techniques—based largely on event history analysis—to assess assimilation gap estimates and to draw conclusions about the size of one assimilation gap in relation to others.

Chapter 7 summarizes major contributions of this research, and suggests directions for future research.

Appendix A summarizes pertinent results from a set of "mini" case studies of early adopters of object technologies. Case study data are used to develop two interrelated themes: the nature of the knowledge burden imposed by OOPL assimilation, and the efficacy of different strategies employed by organizations to manage this burden. A central assumption of this dissertation is that software process innovations in general (and OOPLs in particular) impose a substantial knowledge burden on would-be adopters. It is this assumption that motivated the use of Attewell's macro level theory of technology diffusion as a guiding framework in Chapter 1, and that motivated the decision to narrow the focus, in the variance model developed and tested in Chapters 2 and 3, to organizational characteristics that should be associated with a greater propensity to innovate in the presence of knowledge barriers.

The cases provide a rich array of support for the contention that OOPL assimilation does, in fact, impose a substantial burden of organizational learning. The strategies employed by organizations to cope with, and to overcome the knowledge burden imposed by OOPLs provide another window into the burden of organizational learning associated with SPI assimilation, when one considers the cost of these strategies, and the difficulties organizations encountered in employing them.

CHAPTER 1. DISTINCTIVE CHARACTERISTICS OF SOFTWARE PROCESS INNOVATIONS AND IMPLICATIONS FOR THEORY

1.1 INTRODUCTION

The study of innovation diffusion—the process by which an innovation spreads through a population of potential adopters—has a long history as a multi-disciplinary field. Rogers, in his classic synthesis, identifies over a dozen separate research traditions, ranging from rural sociology to geography to economics [Rogers 1983, p. 53].

These diverse research traditions are unified by their concern with three basic questions:

- 1) What factors explain differences in adopter innovativeness (e.g., earliness of adoption)?
- 2) What factors explain differences in innovation adoptability (e.g., rate of diffusion)?
- 3) What typical patterns do innovations follow as they spread through a community of adopters?

Nevertheless, no single, strongly predictive theory addressing these questions is likely to emerge. The variations in innovations (e.g., product versus process; administrative versus technical; incremental versus radical) and the adoption contexts in which they may be applied (e.g., individual versus organizational; voluntary versus forced; competitive versus non-competitive) are simply too great [Downs and Mohr 1976].

Consider, for example, the contrast between what it takes for an individual to adopt a phone answering machine, compared to what it takes for an organization to adopt a new software process innovation (SPI). Most consumers can, based on a 30 second television commercial, grasp the purported benefits of an answering machine. An SPI, on the other hand, typically requires many weeks of analysis by highly skilled professionals to

perform even a rudimentary technical assessment. A trial of an answering machine requires a small, typically reversible, investment to buy the machine (most electronics stores allow the return of unsatisfactory merchandise), and a few days of using the machine. A meaningful trial of a new SPI requires many thousands of dollars and at least several months. Comparisons between technologies are complicated by differences in product maturity, project team make up, project size, problem domain, funding, scheduling, end user support, and other factors. Consumers are free to adopt answering machines regardless of whether most other people also adopt, because only a small portion of the market is required to make such devices economically viable for producers, and even if they were eventually pulled from the market, it is costless for a consumer to revert to non-use of answering machines when it breaks or wears out. With a new SPI, by contrast, adoption by many others is essential for adoption to be worthwhile in the long run for most organizations, and investments in technology assessment and assimilation are largely irreversible [Fichman and Kemerer 1993]. Finally, use of an answering machine can be inserted into consumer's daily routine with no other corresponding changes. Assimilation of an SPI, by contrast, typically requires a host of related organizational changes—to staffing and training procedures, technical methods, project management procedures, team structure, and incentives. It would be very surprising then if the factors that best explain the propensity to adopt new consumer product, such as an answering machine, were the same as those that predict the propensity to adopt a complex organizational technology, such as an SPI.

As a result, the approach taken in this thesis is to concentrate on particular kinds of innovations and contexts, and to develop theory centered around the distinctive characteristics of those innovations and contexts. Specifically, it is argued that SPIs are distinguished by two characteristics—strongly *increasing returns* to adoption [Arthur 1988] and substantial *knowledge barriers* impeding adoption [Attewell 1992]. The combination of these two factors

suggests that the study of the adoption and diffusion of SPIs will require new explanatory variables and knowledge of new patterns of diffusion.

The remainder of this chapter provides a brief summary and critique of the so-called "classical model" of diffusion developed by Rogers (Section 1.2) followed by a proposed model of diffusion specific to software process innovations (Sections 1.3 through 1.6).

1.2 THE CLASSICAL DIFFUSION MODEL

Everett Rogers, in a widely cited work, provides a synthesis of over 3000 previous studies of adoption and diffusion [1962; 1983]. The results of this synthesis include numerous generalizations about innovation diffusion, which Rogers defines as: "The process by which an innovation is communicated through certain channels over time among the members of a social system" [1983, p. 5]. Among the more well-established generalizations are:

- 1) Innovations possess certain characteristics (relative advantage, compatibility, complexity, trialability, observability) which, as perceived by adopters, determine the ultimate rate and pattern of adoption;
- 2) Some potential adopters are more innovative than others, and can be identified as such by their personal characteristics ("cosmopolitanism," level of education, etc.);
- 3) The adoption decision unfolds as a series of stages (flowing from knowledge of the innovation through persuasion, decision, implementation and confirmation) and adopters are predisposed towards different kinds of influence (e.g., mass market communication versus word-of-mouth) at different stages;
- 4) The actions of certain kinds of individuals (opinion leaders and change agents) can accelerate adoption, especially when potential adopters view such individuals as being similar to themselves; and
- 5) The diffusion process usually starts out slowly among pioneering adopters, reaches "take-off" as a growing community of adopters is established and the effects of peer influence arise, and levels-off as the

population of potential adopters becomes exhausted, thus leading to an "S-shaped" cumulative adoption curve.

Limitations of the Classical Model

The magnitude of Rogers' achievement in providing a coherent synthesis of innovation studies spanning five decades can hardly be overestimated. When employed in appropriate adoption contexts—e.g., individual adoption of innovations for personal use—the classical model can have strong explanatory power [Fichman 1992]. Nevertheless, the appropriateness of the classical model as a general model of diffusion for all adoption contexts has been widely criticized by innovation theorists—including Rogers himself [Kelly and Kranzberg 1978; Rogers and Eveland 1978; Eveland and Tornatzky 1990; Attewell 1992].

The classical model has been criticized along a number of lines. First, it has been criticized for failing to consider the role of several classes of plausible antecedents of organizational innovativeness/innovation adoptability, including:

- 1) *Structural characteristics* of organizational adopters, such as centralization, formalization, and functional differentiation [Zaltman *et al.* 1973];
- 2) Characteristics of the adopting unit's *industrial sector*, including competitive intensity and industry heterogeneity [Robertson and Gatignon 1986], and market size and concentration [Eveland and Tornatzky 1990];
- 3) Influences arising from *supply side institutions* and *propagation mechanisms*, such as marketing allocation, R&D support, technology standardization, and vertical coordination with customers [Robertson and Gatignon 1986].

The classical model has also been criticized for a lack of attention to *intra-organizational* processes of adoption [Rogers 1983, ch. 10] and attendant

explanatory factors, such as expressed managerial preferences, reward systems and infrastructure support [Leonard-Barton and Deschamps 1988].

Finally, the classical model has been criticized for its reliance on simple measures of innovativeness—such as earliness of adoption and adoption/non-adoption—which are unable to capture potentially marked differences in post-adoption innovativeness [Downs and Mohr 1976]. (The topic of measuring organizational innovativeness is addressed in detail in Chapters 4 and 5.)

Most of these criticisms arise from the classical model's disproportional focus on adoption by private individuals, and its resulting inattention to factors that affect adoption by individuals in organizations, and especially, organizations as a whole. The classical model reflects the biases of the research base from which it was synthesized. Rogers notes that fewer than one sixth of the diffusion studies conducted as of 1978 examined adoption by organizations, and that many of those that did nevertheless retained an individualistic focus [1978].

Alternative Models and Frameworks

Many alternatives to the classical diffusion model have been proposed, each of which—like the classical model—reflects the distinctive elements of the contexts for which it was developed. Several of these models, including ones that should be of particular interest to IT researchers, are summarized in Table 1.2.1. Collectively, these models address many of the limitations of the classical model described above. The first four models are the most general; the next two are specific to the IT context; and the last two are even more targeted, focusing on technologies subject to increasing returns, and to knowledge barriers, respectively.

Table 1.2.1: Alternative Diffusion Models

Author	Context		Main Areas of Contrast with Classical Model
	Innovations	Adopters	
Mansfield 1968	Industrial process technologies	Manufacturing firms	Inclusion of economically oriented organiza- tional characteristics (e.g., growth rate, profit level, liquidity) and economically oriented innovation characteristics (e.g., economic advantage, uncertainty, commitment to trial)
Zaltman <i>et</i> <i>al.</i> 1973	Technological and adminis- trative innovations	Organizations	Inclusion of organizational structural characteristics (e.g., centralization, form alization)
Robertson & Gatignon 1986	Industrial process technologies	Private sector firms	Inclusion of influences such as adopter indus- try competitive environment (e.g. industry heterogeneity, competitive intensity, demand uncertainty) and supply-side institution characteristics (e.g., supplier reputation, standardization, marketing support)
Tornatzky <i>et</i> <i>al.</i> 1990	Industrial process technologies	Private sector firms	Inclusion of adopter industry characteristics (e.g., size, concentration); technology deployer characteristics (e.g., resources, size, mission); and boundaries between users and deployers (e.g., structural, cultural)
Kwon & Zmud 1987	Information technologies	Organizations	Inclusion of influences from the external environment (e.g., heterogeneity, uncertainty, competition); organizational structure (e.g., centralization, specialization); and the task environment (e.g., variety, autonomy)
Swanson 1990	Information technologies	Organizations and IT Units	Inclusion of IS unit characteristics (e.g., size, diversity, age of applications portfolio, professional orientation)
Arthur 1988	Technologies subject to in- creasing returns	Organizations and Indivi- duals	Inclusion of influences related to bandwagons and critical mass (e.g., size of prior installed base, sponsorship, technological expectations)
Attewell 1992	Complex orga- nizational technologies	Organizations	Inclusion of influences arising from institu- tions for lowering knowledge barriers (e.g., consulting firms, adoption as a service, technology simplification, special buyer- supplier relationships)

The models developed by Arthur and by Attewell provide a particularly appropriate starting point for a theory of the diffusion of SPIs. SPIs, and

technologies embodied in commercial software more generally, are strongly subject to increasing returns, that is, they become more valuable to any individual adopter to the extent that others also adopt [Brynjolfsson and Kemerer 1993; Fichman and Kemerer 1993]. SPIs also typically require an arduous organizational learning process to acquire the skills and knowledge needed to use them effectively, and this, combined with other costs associated with an immature technological network, can become a barrier to adoption in many organizations (see Appendix A).

The focus here is on increasing returns and knowledge barriers because both characteristics are strongly at odds with the implicit assumptions of classical diffusion [Fichman 1992], and, as explained in the subsections below, both have an emerging literature that develops compelling implications for diffusion theory. While it is true that SPIs possess other potentially distinctive characteristics—they are process innovations (rather than product innovations); they are typically deployed on a project by project basis (rather than by the organization all at once); they are adopted by internal IT departments (rather than other departments or the organization as a whole)—these other characteristics, while important to bear in mind when designing empirical research, do not appear to have comparably resounding implications for theory.

1.3 INCREASING RETURNS

Although it has been known for many years that increasing producer experience and economies of scale in production result in improved price/performance for many technologies over time, only recently have researchers attended to situations where the more general phenomenon of increasing returns to adoption may be a primary driver of technology diffusion dynamics.

Arthur provides a classification of five sources of increasing returns: 1) learning-by-using, 2) scale economies in production/learning-by-doing,

3) technological inter-relatedness, 4) informational increasing returns, and 5) network externalities [1988]. SPIs are typically subject to all five sources of increasing returns (see Table 1.3.1 below).

Table 1.3.1: Sources of Increasing Returns

Source	Explanation	SPI Example
1. Learning-by-using	By employing new technologies in a variety of production settings, users engage in a process of learning about how the technology can be best applied and improved, and this learning gets returned to producers and mediating institutions as an important source of product improvement ideas.	Early adopters of Information Engineering have learned that it is not feasible to develop a detailed, global enterprise model as a discrete project, and that more selective and/or incremental approaches are more likely to be successful.
2. Economies of Scale/ Learning-by-doing	Wider adoption provides vendors with benefits of scale and experience that get translated into better price/performance (thus promoting more pervasive adoption by users) and/or higher unit margins (thus promoting more pervasive adoption by producers).	Later vintages of popular fourth generation languages have had markedly improved machine performance compared with early vintages, owing to the ongoing producer investments in these languages enabled by a larger installed base.
3. Technological Inter relatedness	Widespread adoption of a core technology frequently triggers the emergence of infrastructure technologies that substantially improve the utility of the core technology.	Widespread adoption of relational databases led to the availability of compatible database design methods and tools, a variety of language interfaces, and even dedicated database hardware for high performance applications.
4. Informational Increasing Returns	As a technology becomes more widely adopted, it becomes better known and understood (e.g., uncertainty is lowered).	The burgeoning popularity of object-oriented programming has lead to an explosion of materials (articles, conferences, etc.) explaining the technology and its potential benefits.
5. Network Externalities	Simply belonging to a larger network of users can offer advantages, independent of the other sources of increasing returns.	Adopters of popular CASE tools benefit from a wider variety of application templates, greater availability of trained staff, larger user groups, etc.

Increasing returns matter because when they are strongly present, "critical mass" dynamics and associated factors which inhibit, promote or sustain

bandwagons can become dominating considerations in understanding the overall rate, level, and pattern of innovation diffusion. "Excess inertia" can develop around an existing technology because of reluctance to leave a mature technological network and join an immature one, particularly when investments in the new technology are irreversible [Farrell and Saloner 1987]. In extreme cases, an industry may become "locked-in" to an inferior technology, such as the QWERTY keyboard [David 1985]. In addition, the degree of sponsorship and standardization, and the use of subsidies for early adopters can become critical in tipping the balance in favor of a particular technology [Katz and Shapiro 1986]. For example, France's PTT established "critical mass" for the Minitel videotext service by literally giving away equipment to hundreds of thousands of heavy telephone users during introduction of the technology [Rogers 1991]. "Small events" such as widely publicized successes (or failures), might also tip the balance toward (or away from) a particular technology [Arthur 1988]. It has also been argued that managerial expectations about the future course of technological change of an innovation, its complements, and its substitutes can strongly affect the timing and outcome of the adoption decision [Rosenberg 1976].

In addition, a high degree of heterogeneity of interests and/or resources across the potential adopter population can make it possible for some technologies to bootstrap to "critical mass," even when the value of the innovation, in the absence of many adopters, is quite low for most candidate adopters [Granovetter 1978; Oliver *et al.* 1985; Markus 1987]. In situations where an adopter population has a high degree of heterogeneity, it is more likely that there will be at least some candidate adopters that require only a few, if any, prior adoptions for the innovation to be worthwhile for their own use. Such organizations are said to have a low "adoption threshold"[Granovetter 1978]. After organizations with low thresholds adopt the innovation, it then becomes viable to a new set of candidates with slightly higher thresholds, who adopt, and so on, until the entire population has adopted. A different population, with the same *average* need and resources, but little variation, is

more likely to be stymied by the absence of those first few adopters with low thresholds. Christensen describes an interesting example of this phenomenon from the hard disk drive market, where the demand for smaller, less expensive drives for use in smaller computers provided the production volumes required for the 5.25 inch, and later the 3.5 inch diameter formats to sufficiently improve recording density so as to displace 8 and 14 inch formats even for mainframe computers [Christensen 1994]. If small computers had never been invented and the needs had been uniformly those associated with large computers (e.g., high recording density) then small diameter disks would have had no place to develop the volumes necessary to become viable for mainframe use.

1.4 KNOWLEDGE BARRIERS

All organizational innovations require some measure of organizational learning if they are to be adopted. However, some innovations—what Attewell has termed "complex organizational technologies"—fall on the extreme end of the spectrum in terms of the burdens they place on would-be adopters in obtaining the knowledge needed to understand and assimilate them [1992].

Attewell argues this is true of technologies that, when first introduced, 1) have an abstract and demanding scientific base, 2) are "fragile" in the sense that they don't always operate as expected, 3) are difficult to trial in a meaningful way, and 4) are "unpackaged" in the sense that adopters can not treat the technology as a "black box," but must acquire broad tacit knowledge and procedural know-how to use it effectively [1992].¹ All of this describes most SPIs; in fact, SPIs appear to be exemplars of the kinds of "complex organizational technologies" Attewell had in mind, although his study focused on "business computing" in general (see Table 1.4.1 below).

¹These four technology attributes are based on a classification originally developed by Eveland and Tornatzky [1990].

Table 1.4.1 Technology Attributes Associated with High Knowledge Barriers

Technology Attribute	Explanation	SPI Example
1. Scientific base	Technologies with an abstract or demanding scientific base, or that are not physically observable, are more difficult to explain and require a more active and prolonged learning period on the part of users in order to grasp and deploy them.	Early adopters of structured design were confronted with a daunting array of new design principles and techniques, the use and benefits of which could not be physically observed.
2. Fragility	Technologies with core features that must be replicated exactly to get expected results, or where performance in the laboratory is a poor predictor of performance in the field, create uncertainty for users and require more resources and "hand holding" during deployment.	Getting adequate performance from early relational databases required subtle tradeoffs between normalization and denormalization during logical design, and sophisticated selection and tuning of indexes during physical design.
3. Trialability	Technologies that can not be easily installed in stages and still obtain benefits effectively require that organizations compress all learning about the technology into a pre-implementation phase. Also, large scale implementations require inherently greater implementation expertise than small ones.	According to proponents, to obtain expected benefits from object-oriented programming organizations must simultaneously adopt object-oriented analysis and design, and then build up a library of reusable components; this is expensive, takes time, and impacts most IT job categories.
4. Packaging	When the subcomponents of a technology can not be tightly bundled into a "turnkey" product that can be introduced into organizations unchanged, users are confronted with learning the operational details of each component and their potential interactions.	Successful adoption of integrated CASE tools involves developing expertise in many related components, including: underlying analysis and design methodologies; technical use of the tool; procedures governing how and when the tool is used; and design of project teams.

Attewell argues that when innovations initially impose a heavy knowledge burden, diffusion is better conceptualized as a process driven by lowering knowledge barriers over time than as a process of communication and social influence (as per classical diffusion). Mediating institutions that specialize in creating and accumulating technical know-how, such as consulting and service firms, will come into existence and will effectively capture "economies of scale in learning." That is, by applying the new technology in

multiple client settings, they will be able to amortize substantial learning related costs. Relationships between suppliers and users will go beyond selling, and will become structured around the task of reducing knowledge hurdles. Many organizations, rather than adopting the innovation directly, will obtain the benefits of the technology indirectly as a service, with a transition to self-service occurring more gradually over time.

Attewell's notion of economies of scale in learning can also be extended to certain kinds of end-user organizations—those that are also better positioned to amortize learning costs, owing to greater "learning-related scale." When knowledge barriers are high, such organizations should be more likely to be on the vanguard of adoption, because they can learn more cost effectively. In addition, characteristics that expand an organization's "absorptive capacity"—defined as an organization's ability to appreciate an innovation, assimilate it, and apply to new ends—may be an especially important determinant of innovative behaviors in the presence of knowledge barriers [Cohen and Levinthal 1990]. These characteristics include the extent of knowledge the organization possesses in areas closely related to the focal innovation, as well as the diversity of knowledge contained in the organization in general.

1.5 TOWARD A THEORY OF THE ASSIMILATION AND DIFFUSION OF SPIs

As discussed above, increasing returns and knowledge barriers, each viewed separately, may have considerable implications for the study of SPI adoption and diffusion. It is also important to consider how these two characteristics might *interact*, and the additional implications this interaction might hold for theories related specifically to innovations like SPIs where both characteristics are strongly present. Six propositions leading towards development of such a theory are presented below.

First, when a technology is strongly subject to increasing returns, then it necessarily follows that a wide discrepancy will exist between its *initial performance* (defined as the performance an average adopter is likely to

achieve with a technology during its first few years of commercial availability, including not only productivity or quality improvements, but also amortized conversion, adaptation, learning, and disruption costs) and its *network potential* (defined as the hypothetical future performance a technology would provide if it were to become universally adopted by the network of users, suppliers, and mediating institutions).

Second, because of knowledge barriers and other costs associated with joining and immature technological network, most organizations should find the initial performance of most SPIs to be lower than pre-existing best practices. In other words, increasing returns are not simply a bonus for most adopting firms, but *must* occur for most SPIs to become worthwhile innovations over the long term.

Past studies of innovation diffusion have frequently defined "adoption" as physical acquisition of technical artifacts. Yet, in the case of SPIs, the more telling question may be whether the innovation ever becomes widely deployed within those organizations that have acquired it. This suggests the use of two alternative definitions of adoption, *adoption-as-acquisition* and *adoption-as-deployment*, and inspires the third proposition: an SPI should be especially prone to an *assimilation gap* (a large gap between the pattern of adoption-as-acquisition and the pattern of adoption-as-deployment across the same population of potential organizational adopters) during its *early adoption cycle* (i.e., the first few years following commercial introduction, where uncertainty about the future of the technology is high). (See Figure 1.5.1.) During the early adoption cycle, stakeholders typically communicate a vision of an innovation's main features and benefits based on what it will be like to use the technology at its highest potential, rather than its likely initial level. This, together with managerial expectations of widespread future adoption, can spark rapid rates of acquisition. Yet when knowledge barriers are present, many of the organizations that adopt based on this positive vision may be unable to widely deploy the SPI. This is because the

organizational knowledge needed to generalize, scale up and institutionalize a technology—especially an immature one—typically dwarfs the knowledge needed to evaluate and acquire it. In addition, while organizations can partially offset knowledge barriers during initial projects by using small teams of talented people, selecting favorable applications and users, hiring external consultants, etc., eventually adopters must reach the point where *average* employees can use the technology effectively on larger projects for widespread deployment to be achieved.

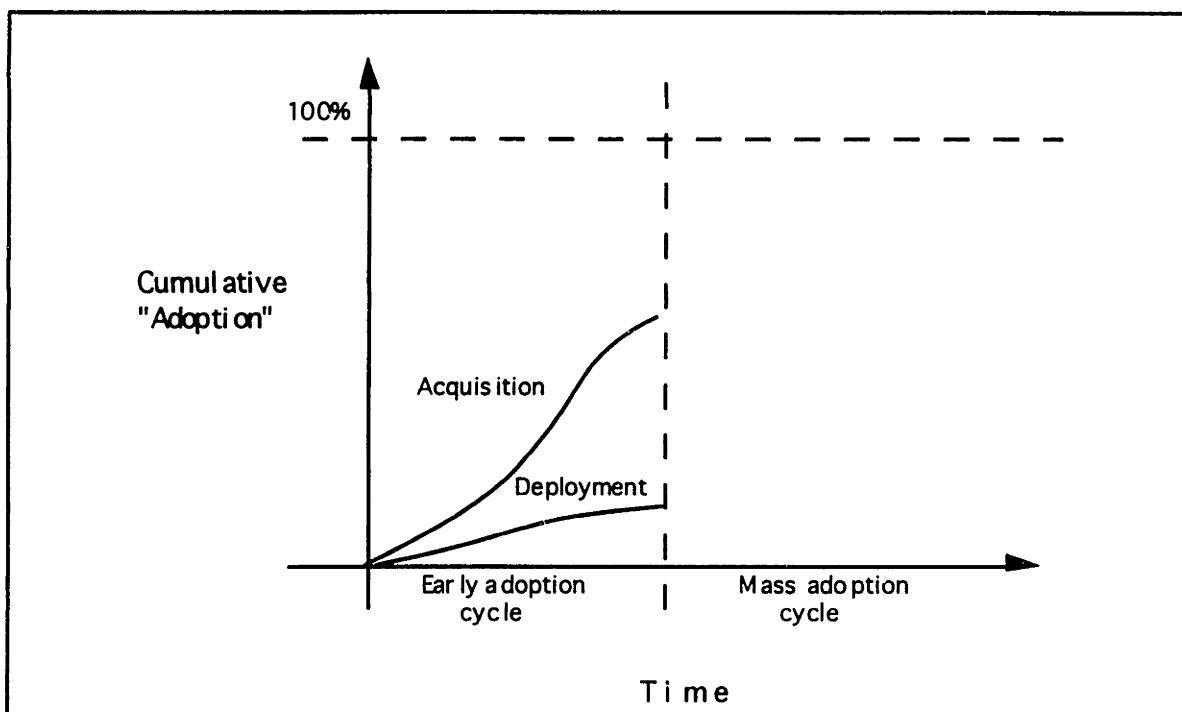


Figure 1.5.1: Assimilation Gap

Fourth, increasing returns should be driven more by cumulative *deployment* than by cumulative *acquisition*. Obviously, adopters of SPIs must deploy them if they are to get benefits themselves. But, beyond this, organizations that acquire but do not widely deploy an SPI will not be engaging in a vigorous process of learning-by-using; they are unlikely to buy the additional licenses, upgrades or add-on products and services that sustain economies of scale and learning-by-doing for producers; they are unlikely to buy

complementary products that support the innovation's infrastructure; they are unlikely to have a useful story to tell other adopters about how to make the innovation work; and they will not be contributing to the emerging network of trained employees, user groups or other network-related resources. Previous experience with SPIs suggests that widespread deployment need not follow acquisition [Yourdon 1986; Howard and Rai 1993].

Fifth, when a large assimilation gap occurs during the early adoption cycle, a *stalled bandwagon* (defined as a situation where "critical mass" is not achieved, and both acquisition and deployment plateau at levels far short of what was originally expected or would have been "optimal" based on the network potential of the innovation) should usually follow, because slow or failed deployment among early adopters delays the learning-by-using and other forms of increasing returns needed to make the SPI attractive to a mass market of potential adopters (see Figure 1.5.2).

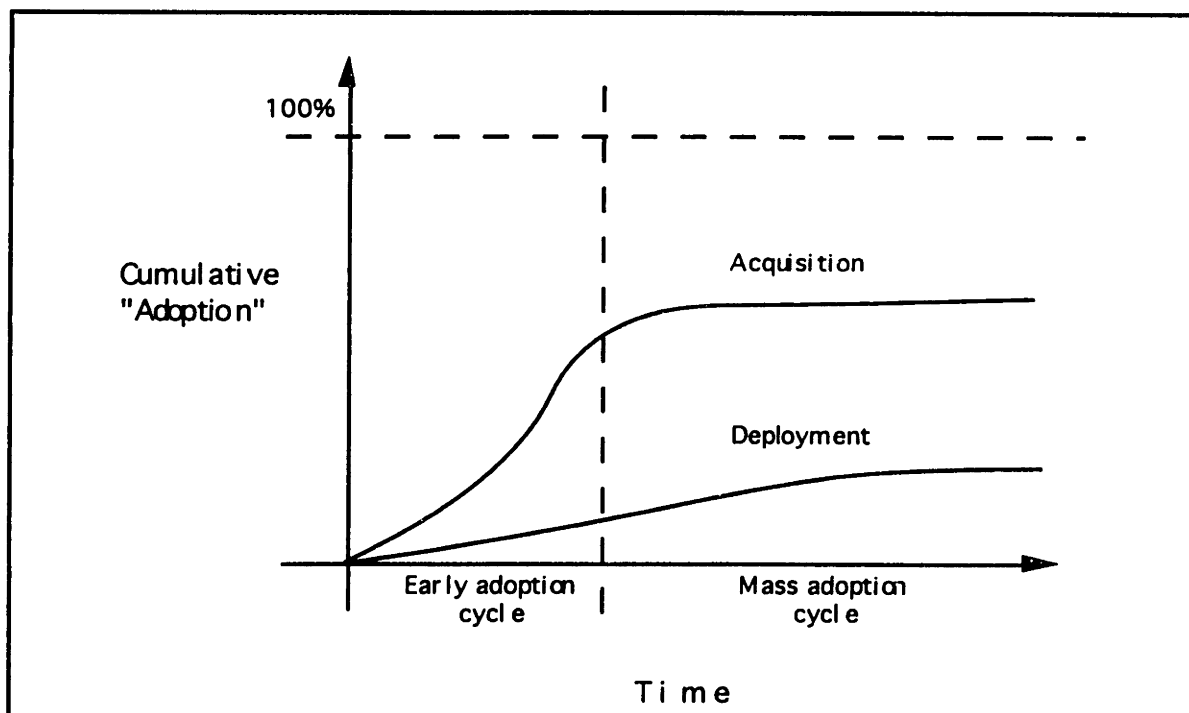


Figure 1.5.2: Stalled Bandwagon

Sixth, there might be instances where robust and innovative institutions for lowering knowledge barriers will overcome a large assimilation gap, resulting in a *sustained bandwagon* of acquisition and deployment. That is, heavy speculative investments in learning-by-doing and learning-by-using among vendors, sponsors, and mediating institutions, together with technology standardization and sharing arrangements, might sometimes lower knowledge barriers sufficiently, even in face of initial deployment problems, to provoke a sustained bandwagon (see Figure 1.5.3).

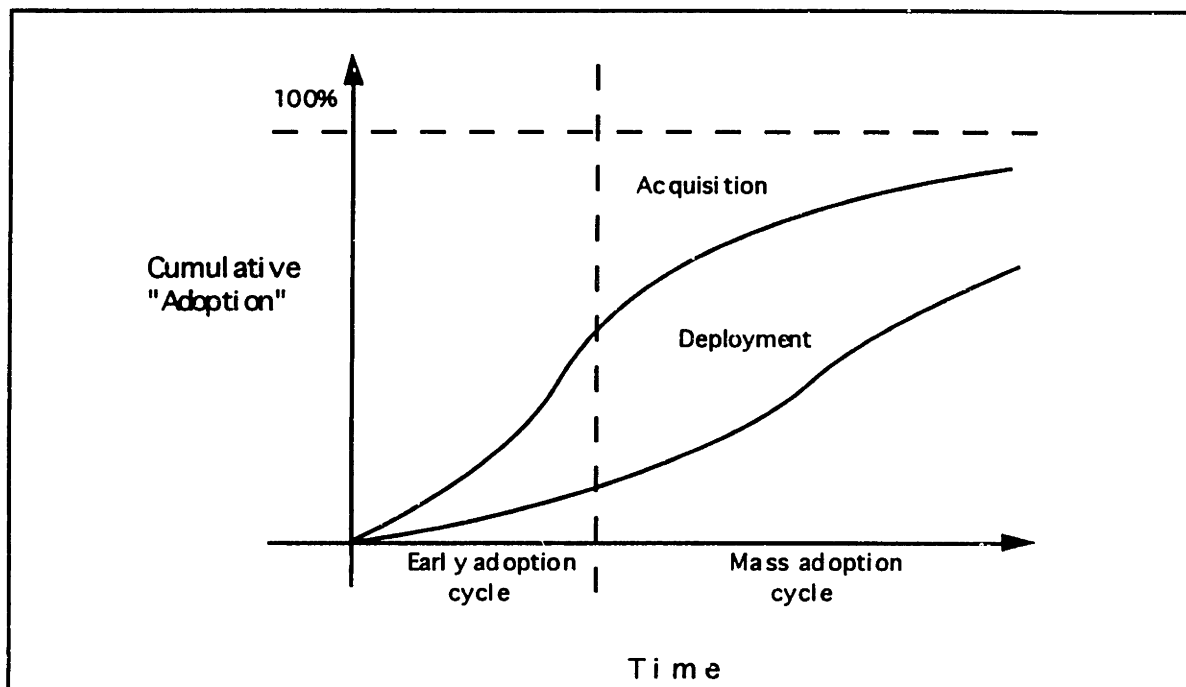


Figure 1.5.3: Sustained Bandwagon

1.6 SUMMARY

To summarize, this chapter has presented the following basic arguments:

- 1) SPIs are strongly subject to all five sources of increasing returns to adoption, meaning there will be a wide disparity between initial performance and the network potential of such innovations.
- 2) SPIs, when first introduced, also have attributes which impede deployment: they are highly abstract and scientific, fragile, difficult to

trial, and unpackaged. This means they impose a substantial knowledge burden on adopters.

- 3) The combination of these two characteristics produces a situation where SPIs have high network potential, but when first introduced, high knowledge barriers and low performance relative to current best practices.

In such a situation, it is argued, a revised conception of diffusion dynamics is called for where new explanatory variables (e.g., sponsorship, standardization, expectations, adopter heterogeneity, institutions for lowering knowledge barriers, economies of scale in learning, related knowledge), and new patterns of diffusion (e.g., assimilation gaps, stalled bandwagons) become central to understanding and modeling adoption and diffusion.

The empirical components of this thesis build on the above theory in two ways. First, the implications of knowledge barriers for SPI assimilation are more fully developed (Chapter 2) and then tested using data from a large scale survey of the assimilation of object-oriented programming languages (Chapter 3). Second, in Chapter 6, data on the diffusion of three other SPIs (RDBs, 4GLs and CASE) are used to explore proposition four, which posits the existence of assimilation gaps.

This page intentionally left blank.

CHAPTER 2. AN ORGANIZATIONAL LEARNING-BASED MODEL OF SOFTWARE PROCESS TECHNOLOGY ASSIMILATION

The assimilation of software process innovations (SPIs) typically requires a lengthy and arduous process of organizational learning wherein individuals and the organization as a whole acquire the knowledge and skills necessary to effectively apply the technology. Assimilating an SPI requires learning on a number of fronts, including grasping the abstract principles on which the technology is based; understanding the nature of the benefits attributable to the technology; grasping specific technical features of different commercially available instances of the technology; discerning the kinds of problems to which the technology is best applied; acquiring individual skills and knowledge needed produce a sound technical product on particular development projects; and designing appropriate organizational changes in terms of the team structure, hiring, training, and incentives.

It is hard to overstate how difficult and expensive this learning process can be. Huff estimates that the 5-year cost of CASE tool adoption is \$35,000 per user, with the majority of this cost attributable to such learning-related activities as technology evaluation, consultants, installing a methods support group, and training [1992]. And, these estimates do not include the potentially substantial cost of experiential learning required to reach the point on the learning curve where performance comparable to pre-existing technologies has been attained [Kemerer 1992].

The case studies of early adopters of object technology provide additional examples of the knowledge burden imposed by SPI assimilation (see Appendix A). At one site, a large energy company spent millions of dollars on learning-related activities and developing infrastructure, all for a staff of 15 developers. Project team members were put through a six week "indoctrination period," consisting of two weeks of off-site introductory training, followed by four weeks of advanced on-site training. Two OO "mentors" were hired and kept on-site for over a year, with their primary

charter being to provide direction and facilitate learning, rather than to do development work themselves. Several team members were assigned to spend the better part of two years developing "infrastructure" tools needed to facilitate use of Smalltalk by the team as whole. These included a CASE-like modeling tool, a interface building tool, and a relational database interface tool. It was noted by one developer that participation in the infrastructure projects served as an important training ground, and that those who did not have this opportunity never reached the level of technical proficiency of those who did. As Attewell observes: "studies have shown that while one can readily buy the machinery that embodies an innovation, the knowledge needed to use modern production innovations is acquired much more slowly, and with considerably more difficulty" [Attewell 1992, p. 5].

This chapter presents a theoretical model explaining differences in the propensity of organizations to initiate and sustain innovations in software process technology. Following Downs and Mohr's [1976] position that, in the absence of a unitary theory of innovation, researchers should develop theories of the "middle-range" focusing on the distinctive qualities of particular kinds of technologies, this model places factors that promote *organizational learning* at its core.¹ Specifically, this model proposes that organizations with a higher propensity to innovate with software process technologies are distinguished by three characteristics: 1) greater "learning-related scale," 2) more extensive pre-existing knowledge in areas related to the innovation, and 3) greater diversity of knowledge and activities in general. Following a presentation of the general theoretical underpinnings of this model in Section 2.1, the constructs, and the rationales linking them to

¹Organizational learning has been receiving increasing attention not only in the study of technology adoption and diffusion, but also in related innovation sub-fields such as research and development [Cohen and Levinthal 1990], implementation of "home grown" technologies [Leonard-Barton 1989], and the transfer of technologies across divisions [Argote *et al.* 1990]. For more general reviews of organization learning see [Fiol and Lyles 1985; Levitt and March 1988; Huber 1991; Dodgson 1993].

technology assimilation are presented in Section 2.2 Finally, Section 2.3 provides a discussion of important issues and concerns raised by the model.

2.1 RELATED THEORY

The primary theory base for this model lies in Attewell's recent work linking organizational learning and innovation diffusion [1992]. In this work, Attewell provides a compelling reconceptualization of innovation diffusion theory for what he calls "complex organizational technologies," i.e., technologies that—like SPIs—impose a substantial knowledge burden on would-be adopters. As described in Chapter 1, these technologies typically have an abstract and demanding scientific base; tend to be "fragile" in the sense that they don't always operate as expected; are difficult to trial in a meaningful way; and are "unpackaged" in the sense that adopters cannot treat the technology as a "black box" [Eveland and Tornatzky 1990].

Any theory of innovation incorporates (in some fashion) communication of new information about innovations, and hence, learning by potential adopters. What distinguishes Attewell's approach is his explication of the *kinds* of information involved, and the *mechanisms* by which information is acquired and propagated. Attewell draws a central distinction between the communication of "signalling" information about the existence and potential gains of the innovation, versus know-how and technical knowledge. Classical diffusion theories implicitly focus on signalling information, and assume that the prominent factor explaining differences in innovativeness is differences in the time it takes this information to reach potential adopters. Attewell argues, however, that acquiring technical knowledge and know-how places far greater demands on potential adopters, and as a result ". . . it plays a more important role in patterning the diffusion process of complex technologies than does signalling . . . [and] should move to center stage in any theory of complex technology diffusion" [1992, p. 5].

Attewell argues that know-how and technical knowledge are tacit, relatively immobile, and often have to be recreated by users via the processes of learning-by-doing [Arrow 1962] and learning-by-using [Rosenberg 1982]. Others have made similar distinctions between classes of information. Von Hippel asserts that the kind of information used in technical problem solving is "sticky"—that is, it is costly to acquire, transfer and use in a new location [1994]. Badaracco distinguishes "migratory knowledge," which can be fully and clearly articulated, and can be transferred via books, formulas and machines, from "embedded knowledge," such as individual craftsmanship and know-how, and team-based knowledge [1991]. This distinction between classes of information is crucial to the understanding of innovation diffusion: if the knowledge needed to use complex technologies weren't "tacit," "sticky," and "embedded," then it could be readily bundled with the artifacts embodying the innovation and incorporated into the purchase price. Then the ability of organizations to learn—and the cost of such learning—would be a less important concern.

Attewell concludes that, for complex organizational technologies, the immobility of know-how and technical knowledge become a barrier to diffusion, and that supply side institutions have to innovate to lower these barriers, both in the design of products and in the development of novel institutional mechanisms. These mechanisms include: mediating institutions that specialize in acquiring and propagating knowledge, such as consulting and service firms; special buyer-supplier relationships structured around learning (such as user groups); adoption by end-users as a service; and technology simplification. Attewell thus places organizational learning at the center of his theory by focusing on mechanisms that, over time, lower the burden of organizational learning surrounding adoption.

However, while Attewell's model suggests that many organizations will defer adoption until knowledge barriers are sufficiently lowered, it does not explicitly address the question as to which end-user organizations should be

among the early adopters, even in face of high knowledge barriers. It is proposed here that organizations with a higher propensity to adopt innovations in the face of high knowledge barriers will be those for which the burden of organizational learning is effectively lower, either because they already possess much of the know-how and technical knowledge necessary to innovate, or because they can acquire it more easily or more economically. The model described below builds on Attewell's work by identifying the characteristics of such organizations.

2.2 PROPOSED MODEL

What kinds of organizations should have a higher propensity to initiate and sustain innovations in software process technology? Attewell's description of consulting and service firms provides a first clue. He notes that these organizations are able to capture "economies of scale in learning" because they can apply what they learn about new technologies in multiple client settings. This suggests a closely related idea, that of *Learning-Related Scale*. Organizations with greater Learning-Related Scale have a greater *opportunity* to capture "economies of scale in learning," because they have a greater scale of subsequent activities related to the focal innovation. This opportunity may or may not be exercised in any given instance.

A second clue lies in the work of Cohen and Levinthal, who assert that a firm's ability to appreciate an innovation, assimilate it, and apply it to new ends—what they term its "absorptive capacity"—is largely a result of the firm's pre-existing knowledge in areas related to the focal innovation, and its diversity of knowledge in general [1990]. These three explanatory factors—Learning-Related Scale, Related Knowledge, and Diversity—are developed in detail in the sub-sections below.

2.2.1 Learning-Related Scale

Learning-Related Scale is defined here *as the scale of activities over which learning costs can be spread*. Just as consulting firms typically have greater Learning-Related Scale because of their base of potential clients traditional IT departments possess similar opportunities to the extent that they engage in a steady stream of development of new applications. The users of these applications can be viewed as "clients," and different functional areas within the host organization who are the recipients of delivered systems serve as different "client settings." Learning-Related Scale should have a more pronounced effect on the assimilation of technologies that have high knowledge barriers because, as argued above, learning costs (evaluations, trials, pilot projects, training, learning-by-doing, developing infrastructure) are likely to swamp the out-of-pocket costs of purchased products.

This research hypothesizes that organizations that possess greater Learning-Related Scale should be more likely, others things being equal, to initiate and sustain innovative activities, because decision makers are more likely to recognize, either explicitly or implicitly, the opportunity for quicker and more assured pay-back of the substantial required investments in learning. Such organizations should also be more likely to engage in learning tactics that improve chances of success—but may entail high fixed costs—because of the opportunity to recover those costs over a broader scale of subsequent activities.

The mini-case studies of OOPL adopters provide two good examples of such tactics (See Appendix A). Example 1: a multinational financial services firm pays a six-figure salary to hire an experienced OO "architect", and leverages him by: 1) having him be responsible for recruiting experienced OO developers, 2) having him establish procedures to be followed on all object-oriented projects, and 3) having him act as a global architect who, through participation in design reviews of every development project, insures designs conform to object-oriented principles. If not for the potential high volume of

subsequent object-oriented development activities (i.e., high Learning-Related Scale), the cost of hiring top talent would have been difficult to justify. Example 2: a multinational energy company spends two years and millions of dollars on the in-house development of "infrastructure" tools to facilitate the development of an expected stream of OOPL applications. One of the reasons these costs were viewed as justified is that they were attached to an even larger development project. In both cases, high Learning-Related Scale enabled the use of a high fixed cost assimilation tactic deemed to be essential to successful OOPL assimilation.

2.2.2 Related Knowledge

It is an intuitively appealing notion that it is easier to acquire new knowledge in the vicinity of prior related knowledge. One need only consider the difference in effort needed to understand and appreciate a new journal article fully within one's field of specialization, compared with an equally lengthy and complex article in a distant field, where the novel contribution of the article may be shrouded in unfamiliar theories, methods and prior empirical results. This research hypothesizes that a similar relationship holds for the case of organizations acquiring the knowledge needed to assimilate SPIs. Specifically, it is hypothesized here that organizations with more extensive prior knowledge in domains related to a software process technology are more likely to initiate and sustain innovation with that technology, because the burden of organizational learning is correspondingly lowered. Unlike Learning-Related Scale and Diversity, the measures employed for Related Knowledge will always be innovation-specific.

Related Knowledge is defined here as *the extent of abstract knowledge, know-how and skills possessed by the organization in areas related to the focal innovation*. Before explaining the linkage between Related Knowledge and innovation, it seems appropriate to clarify what is encompassed by Related Knowledge. Successful assimilation of an SPI requires that an organization reach the point where it possesses an associated bundle of knowledge and

skills. This bundle includes at least three components. The first component is knowledge of the abstract principles on which the innovation is based and the skills necessary to apply those principles. Examples include knowledge of set theory operations and normalization in the case of relational databases, knowledge of functional decomposition in case of structured analysis, and knowledge of encapsulation, inheritance and message passing in the case of object-oriented programming languages. The second component is expertise in specific technology products embodying the focal innovation, i.e., a particular OOP language, CASE tool, etc. These technologies are typically extraordinarily complex; software professionals can spend months or years acquiring the expertise necessary to become proficient with any particular product, be it a database, methodology, language or other sort of process tool. The third component is expertise in technologies that typically form a "cluster" around the focal innovation. With the notable exception of early artificial intelligence languages—which were implemented on standalone, propriety systems—SPIs tend to be integrated with, and implemented on top of, existing technologies. Pre-existing knowledge and skills in these technologies can be seen as being part of the knowledge base for the focal innovation. Related Knowledge, then, includes knowledge of associated abstract principles, expertise in the technologies directly embodying the innovation, and expertise in the technologies that cluster around the focal innovation.

Why should a linkage exist between prior related knowledge and ease of organizational learning surrounding new technologies? Two complementary arguments can be made. First, drawing on work of cognitive psychologists, Cohen and Levinthal [1990] develop the argument that prior related knowledge makes it easier for individuals to acquire and retain new knowledge because it gives individuals rich, well-organized, mental schemas into which new knowledge can be placed, and allows the associative connections needed for insights related to the new knowledge. Ease of organizational learning follows from ease of individual learning, because while it has been argued that individual learning is not always sufficient for

organizational learning,² it is certainly necessary. Second, prior related knowledge effectively diminishes the "distance" a firm must travel to get from its current bundle of knowledge and skills, to one that encompasses the intended innovation [Pennings and Harianto 1992b, p. 362]. Since some of the burden of knowledge acquisition is eliminated, innovation is less costly, and less likely to fail.

To summarize, Related Knowledge facilitates innovation by increasing the ability of the organization to acquire any given quantity of new knowledge, and by reducing the actual quantity of new knowledge that must be acquired.

2.2.3 Diversity

This research defines Diversity as *the degree of heterogeneity of organizational knowledge and activities (technical, functional, and business) in areas related to applications development*. Diversity thus encompasses both diversity of knowledge and diversity of activities. Cohen and Levinthal (1990) argue that diversity of knowledge facilitates learning because when there is uncertainty about the domains from which potentially useful information may emerge, a diverse knowledge base increases the likelihood that new information will be related to what is already known. They also argue that diversity of knowledge enables individuals to make novel associations and linkages involving the focal innovation; this is especially important because technologies typically require "reinvention" [Rice and Rogers 1980] or adaptation [Leonard-Barton 1988a] as part of the assimilation process.

Rationales linking diversity of activities to innovation have also been proposed. Swanson argues that diversity of activities increases the likelihood that an organization will have at least one domain that is sufficiently "innovation

²Attewell argues, for example, that individual learning becomes organizational learning only insofar as it becomes embodied in organizational routines and practices [1992].

ready"—due to resources of the receiving area, or compatibility of the innovation with the receiving area—for the innovation to be introduced to the organization [1994]. In addition, in the presence of learning-by-using or other forms of intra-company increasing returns to adoption, diversity of activities contributes to an environment where innovation can more easily be sustained, essentially because the organization can bootstrap from areas with a high fit to those with lower ones [Markus 1987; Christensen 1994].³

It is therefore hypothesized that Diversity of organizational knowledge and activities in areas related to applications development, other things equal, will be positively related to the assimilation of SPIs. It is worth noting that, in actual practice, diversity of knowledge and diversity of activities are likely to be inextricably linked, because the knowledge contained within an organization is largely a result of its past history of activities [Nelson and Winter 1982].

2.3 DISCUSSION

This chapter has presented the argument that SPIs typically impose a substantial knowledge burden on would be adopters; that when such a burden exists, organizational learning becomes central to the understanding of the innovation process; and that, in particular, organizations possessing greater Learning-Related Scale, more extensive Related Knowledge, or broader Diversity of knowledge and activities are more likely to initiate and sustain innovation with software process technologies. Such organizations can better amortize learning costs, should be better able to assimilate any

³It can be expected that for some organizations Diversity will be confined only to areas that are unrelated or have a poor fit with the focal innovation, and should therefore have no positive effect (and in fact might pick up a negative compatibility effect). However, all other things being equal, such organizations can be expected to be balanced by other organizations where, also by chance, Diversity is confined only to high fit areas, and as a result, is an even stronger predictor than it ordinarily would be.

given amount of new knowledge, and have a lesser burden of organizational learning in the first place.⁴

It is appropriate to pause here to consider a number of issues and concerns raised by the proposed model. First is the question of how broadly the model can be generalized. The model was specifically developed with SPIs in mind, and examples related to these technologies are used to make the discussion more concrete. However, the model can be expected to generalize to other complex process technologies that impose a similar knowledge burden when first introduced.

Second, the proposed model rests on the assumption that organizations have a higher propensity to innovate if they can do so more economically or have higher probability of success. This is primarily a rational choice argument. It assumes that managers—either explicitly or implicitly—recognize when their organization is a more appropriate candidate for innovation. Of course an organization may undertake innovation for non-instrumental reasons, such as for increased prestige, even when it is not otherwise a good candidate. Furthermore, managers may inappropriately conclude that their organization is a good candidate for successful and economical innovation, and initiate assimilation prematurely, before knowledge barriers have been sufficiently lowered. It should be noted, however, that to the extent that organizations make decisions straying too far from the normative ideal, they should be less likely to be able to sustain innovative activities. As a result, the argument does not rest entirely on assumptions of *a priori* rational choice.

Third, it should be noted that the factors contained in the proposed model represent only the *opportunity* for more economical and successful

⁴While the proposed model is centered around the distinctive characteristics of SPIs, some linkages can still be made to the classical diffusion model described in Chapter 1. What has been labeled Learning-Related Scale here might be viewed as a determinant of "perceived relative advantage" in the Rogers model. Likewise, greater Related Knowledge might be viewed as a determinant of lower "perceived complexity".

innovation. Some managers may not recognize such an opportunity exists, or may not know how to exploit it. Successful assimilation of SPI requires substantial investments in process change and developing infrastructure. Greater Learning-Related Scale gives an organization the opportunity to amortize these learning costs, but does not guarantee that managers will be willing to commit necessary resources in the first place, or that they will be aware of the best ways to allocate those resources. As result, a natural complement to this research would be an examination of variables that capture the extent to which appropriate implementation tactics are actually invoked by adopters. Tactics implicit in the above discussion include organizational investments in "rich" mechanisms for learning and knowledge transfer (e.g., using mentors or introducing extended periods of on-site "practice" with the technology), a conscious strategy of innovating, where possible, in the vicinity of existing organizational knowledge, and exploitation of diversity by identifying a "safe haven" for the technology (an area where the technology is a particularly good fit and the learning process can begin).

Finally, it should be noted that a vast array of other variables have been previously identified as potentially influencing organizational adoption, including: leader characteristics and attitudes [Kimberley and Evanisko 1981; Meyer and Goes 1988]; communication channel use [Zmud 1983; Nilakanta and Scamell 1990]; top management support and technology champions [Roberts and Fufeld 1981; Beatty 1992; Howard and Rai 1993]; training [Raho *et al.* 1987; Leonard-Barton and Deschamps 1988]; structural characteristics such as centralization and formalization [Zaltman *et al.* 1973; Zmud 1982]; and environment characteristics like competitiveness and concentration [Robertson and Gatignon 1986; Eveland and Tornatzky 1990]. These variables (with the exception of training, top management support and technology champions) are more generally associated with an organizational willingness or need to innovate rather than an organizational ability to innovate, and while it might be hypothesized that such variables will play a less

pronounced role in the assimilation of SPIs, there is no reason to assume their influence will vanish. Empirical studies must therefore include some reasonable number of these variables to eliminate noise, and to control for likely confounds.

CHAPTER 3. AN EMPIRICAL STUDY OF THE ASSIMILATION OF OBJECT-ORIENTED PROGRAMMING LANGUAGES

This research employs a naturally occurring quasi-experiment to test the model presented in Chapter 2. The form of the experiment is a post-test only, non-equivalent groups design predicting the propensity of organizations to initiate and sustain SPI assimilation. To support this analysis, detailed data about a single innovation type, object-oriented programming languages (C++, Smalltalk, etc.) were collected cross-sectionally from single key informants in IT departments at over 600 business sites.

The primary goal of this chapter is to test the expected role of organizational learning-related factors in a way that promotes generalizability of results to the universe of internal applications development sites in medium to large US enterprises. This research is also intended to generalize to other typical SPIs besides object-oriented programming languages (OOPLs) and to other general measures of organizational innovativeness besides Assimilation Stage. One of the reasons for the primary focus on OOPLs is that they, like SPIs more generally, exhibit the characteristics associated with high knowledge barriers (abstract/demanding scientific base, "fragility," etc.) (see Chapter 1 and Appendix A).

The remainder of this chapter is organized as follows: Section 3.1 provides a brief overview of OOPLs. Section 3.2 summarizes potential validity threats. This is followed by a description of data collection procedures in Section 3.3. Section 3.4 provides a detailed description of the measures used to operationalize theoretical constructs, and a statistical analysis of the validity of those measures. Sections 3.5 and 3.6 present the results of data analysis, and a discussion of those results, respectively.

3.1 POTENTIAL VALIDITY THREATS

The primary objective of research design is to manage potential threats to the validity of expected results. This research relies on three primary sources for guidance on identifying and managing potential threats to validity: Cook and Campbell [1979], Bagozzi [1980] and Lee [1989].

Cook and Campbell, in their definitive text, identify four categories of threats: 1) statistical conclusion validity, 2) internal validity, 3) construct validity, and 4) external validity. Bagozzi provides a useful complement to Cook and Campbell by giving a more thorough and up-to-date treatment of construct validity. Issues related to internal, external, and statistical conclusion validity are addressed throughout this chapter, and then summarized in Section 3.6. An analysis of construct validity is provided in Section 3.4.6.

Drawing on prior work by other researchers, Lee describes some additional validity threats specific to studies that, like this research, rely on key informants to answer questions about quantifiable organizational properties. These threats include: 1) perceptual and cognitive limitations, 2) lack of information, and 3) motivational barriers to participation [Seidler 1974; Silk and Kalwani 1982; Huber and Power 1985; Lee 1989]. The negative effects of perceptual and cognitive limitations were managed by the use of the disk-based survey mode and by careful attention to survey item construction (see Sections 3.3.1 and 3.3.2). Lack of information was managed through the selection of informants likely to be well informed and through evaluation of the characteristics of actual informants (See Sections 3.3.3 and 3.3.5). Finally, motivational barriers were lowered through survey administration procedures, such as provision of incentives and assurances of confidentiality (See Section 3.4.4).

A summary of salient validity threats and the procedures used to address them is provided in Section 3.6.1. The analysis of validity threats uncovered no unmanageable problems.

3.2 OVERVIEW OF OBJECT-ORIENTED PROGRAMMING LANGUAGES

OOPs are a family of programming languages designed to facilitate the development of software systems conforming to object-oriented principles [Pascoe 1986; Peterson 1987; Thomas 1989]. These principles—most notably encapsulation and inheritance—are thought to promote such long standing software engineering goals as abstraction, modularity, reuse, maintainability, extensibility, and interoperability [Meyer 1987; Taylor 1990; Wirfs-Brock *et al.* 1990; Booch 1994]. Table 3.2.1 below provides definitions of some key object-oriented terms.

Table 3.2.1: Object-oriented Terms

Term	Definition
Class	An abstract definition or "template" for a collection of objects that share identical structure and behavior.
Encapsulation	The practice of enclosing a data structure and all operations which access the structure inside a metaphorical "capsule" (i.e., an object), and disallowing direct access to the data by means other than those operations.
Information Hiding	A design principle which states that the internal structure of a given module should be a "black box" that stays hidden from other modules. Information hiding insulates other modules from changes that affect a given module's internal representation but not its external interface.
Inheritance	A mechanism which allows objects from different (but related) classes to share common characteristics (variable and method definitions). This is achieved by placing common characteristics in higher-level classes, and creating links to those classes from lower-level classes.
Message Passing	The practice of using explicit messages sent from one object to another as the only form of object communication. A message consists of an object identifier, a method name, and a set of arguments.
Method	A small program associated with an object that performs an atomic and cohesive operation on that object's data.
Object	An abstraction of a real-world object which encapsulates a set of variables and methods corresponding to the real-world object's attributes and behaviors.
Polymorphism	A programming mechanism that allows the same message protocol to be sent to objects from distinct yet similar classes to invoke a similar action. For example, polymorphism allows the same PRINT message protocol to be sent to a document or spreadsheet, rather than having two separate message protocols for each.
Variable	An item of data associated with an object.

Cook and Campbell, in their definitive text, identify four categories of threats: 1) statistical conclusion validity, 2) internal validity, 3) construct validity, and 4) external validity. Bagozzi provides a useful complement to Cook and Campbell by giving a more thorough and up-to-date treatment of construct validity. Issues related to internal, external, and statistical conclusion validity are addressed throughout this chapter, and then summarized in Section 3.6. An analysis of construct validity is provided in Section 3.4.6.

Drawing on prior work by other researchers, Lee describes some additional validity threats specific to studies that, like this research, rely on key informants to answer questions about quantifiable organizational properties. These threats include: 1) perceptual and cognitive limitations, 2) lack of information, and 3) motivational barriers to participation [Seidler 1974; Silk and Kalwani 1982; Huber and Power 1985; Lee 1989]. The negative effects of perceptual and cognitive limitations were managed by the use of the disk-based survey mode and by careful attention to survey item construction (see Sections 3.3.1 and 3.3.2). Lack of information was managed through the selection of informants likely to be well informed and through evaluation of the characteristics of actual informants (See Sections 3.3.3 and 3.3.5). Finally, motivational barriers were lowered through survey administration procedures, such as provision of incentives and assurances of confidentiality (See Section 3.4.4).

A summary of salient validity threats and the procedures used to address them is provided in Section 3.6.1. The analysis of validity threats uncovered no unmanageable problems.

3.3 DATA COLLECTION

A disk-based survey mailed to IT managers was the primary mode of data collection for this thesis, although two specialized follow-up surveys (one via telephone and the other paper-based) were also administered to gather

additional data. This subsection describes the procedures used to collect data, and covers additional related topics as follows:

- 1) Overview of disk-based surveys;
- 2) Description of survey instrument development procedures;
- 3) Description of the creation of survey sample;
- 4) Description of the disk-based survey administration procedures;
- 5) Summary of the characteristics of responding sites and informants;
- 6) Analysis of potential for response bias;
- 7) Summary of the results of telephone survey of non-respondents;
- 8) Description of a follow-up paper-based survey.

3.3.1 Disk-Based Surveys (DBSs)

This research employed a disk-based survey (DBS) as the primary data collection technique [Pilon and Norris 1988; Horton 1990; Saltzman 1993]. This novel technique is especially well suited to the target informants—IT managers—the vast majority of which have IBM compatible PC's on their desks at work.

Overview of Disk-based Surveys

DBSs are outwardly similar to their paper-based counterparts, except that informants receive a computer disk instead of a hard copy questionnaire. They insert the disk into their PC's disk drive and initiate the survey session. A program then automatically leads them through the questionnaire items.

For this survey the Ci3 authoring system, marketed by Sawtooth Software, was used [Anonymous 1992]. Capabilities provided by this system include: programmable item validation; context sensitive help; sophisticated list

processing capabilities; the ability to store prior responses as variables that can be displayed as part of the text of subsequent questions; the ability to perform transparent branching based on the responses to one or more of the previous questions; and randomization, if desired, of the order of items within questions, questions within blocks, and blocks within questionnaires. All of these features (except for the last two forms of randomization) were, in fact, employed in constructing the DBS used for this research.

DBS Advantages and Disadvantages

For this application, the DBS mode combines the most important advantages of both paper-based and telephone survey modes (see Table 3.3.1.1 below). DBSs also incorporate the advantages of computer-based telephone interviewing, including elimination of most kinds of informant coding errors, and complete elimination of transcription errors and delay. In addition, current evidence suggests that DBSs have higher response rates than comparable paper-based surveys [Pilon and Norris 1988; Horton 1990; Saltzman 1993]. Although no studies have been done to confirm why this might be, it has been suggested that higher response rates occur because: 1) respondents cannot easily determine survey length and complexity [Pilon and Norris 1988; Saltzman 1993]; 2) respondents simply prefer working at PC's to paper and pencil [Horton 1990; Mitchell 1993]; and 3) the novelty of the mode piques the interest of respondents [Saltzman 1993].

Table 3.3.1.1: Comparison of Survey Modes

Advantage	Paper-based	Telephone	Disk-based
Allows complex, multi-item questions	✓		✓
Eliminates interviewer effects	✓		✓
Lowers social desirability response due to interviewer presence	✓		✓
Allows informants to take survey at their own pace and convenience	✓		✓
Allows informants to interrupt survey sessions and resume later	✓		✓
Provides interactive feedback to correct informant errors or to eliminate confusion		✓	✓
Allows researcher to retain control over question sequencing		✓	✓
Minimizes item non-response, especially inadvertent		✓	✓

DBSs are not without disadvantages, however. DBSs provide no opportunity to do on-the-spot screening of respondents—a major advantage of telephone interviews. Also, although at least one researcher has found administration costs to be comparable to paper-based surveys on a per response basis [Pilon and Norris 1988] there is the additional cost to program and test the DBS itself. For a long and complex questionnaire, this can be considerable. For the DBS employed in this research, over 100 hours of programming and an equal amount of quality assurance testing were required. Finally, unless the overwhelming majority of intended respondents have compatible computer hardware readily available, the use of the DBS mode could serve to lower response rates and introduce response bias.

Potential for Mode Bias

The use of any novel mode of survey administration raises the specter of mode bias. Two distinct kinds of bias are possible. First, it might be that the kinds of informants who choose to respond to DBSs are systematically different than those who respond to other modes. Second, it might be that informants who do respond are prone to respond differently to DBS questions

compared with questions administered in other ways. Keisler and Sproull found, for example, that people taking surveys administered electronically (via E-mail) had more extreme responses and less socially desirable responses than those taking equivalent paper-based surveys [1986].

The first kind of bias is unlikely with this research. In follow-up telephone interviews with 34 non-respondents, 29 reported having DOS-based PC's on their desks and the remaining 5 had PC's readily available (see Section 3.3.7 below). As a result, the lack of access to compatible hardware does not appear to be common enough to introduce a sizable bias. In addition, only one non-respondent reported concerns with the media itself as the primary reason for non-response (specifically a bad experience with a previous DBS that took two hours to complete). Other researchers who have performed several DBS surveys have similarly found no bias on who responds compared with paper-based modes [Bernstein 1993; Smith 1993].

The second kind of bias—in how informants respond—also is unlikely in this research. Mitchell [1993] found no mode bias in a classroom-based experiment of DBS versus paper-based modes. Likewise, researchers who have performed several DBS surveys have found no mode bias [Bernstein 1993; Smith 1993].

This then raises the question of why a mode bias might exist for E-mail surveys but not for DBSs. Keisler and Sproull argue that the mode bias found in their study was due a lack of social cues, including official envelopes and cover letters on official letterhead [1986]. With the DBS mode, however, such cues are not lacking. Informants still receive a mailed package containing a cover letter and survey; they still respond using a provided mailer; and they still receive hard copy reminder and letters. In fact, the DBS mode only changes how informants physically take the survey, not how they interact with the sponsoring organization. Therefore, mode effects are not believed to be a significant source of bias in the research.

3.3.2 Survey Instrument Development Procedure

The survey instrument was developed over a four month period extending from mid-September, 1993 to mid-January, 1994. The main source of guidance on survey construction and refinement techniques was Dillman's "Total Design Method" [1983, ch. 2-3]. The resulting instrument was quite lengthy and complex, with a total of 104 questions, 106 computer screens, and 35 branch points. (A listing of DBS screens is provided in Appendix B.) Item validations were included to ensure that numeric values fell within allowable ranges, to insure percentages summed to 100, and to ensure that data provided later in the questionnaire were consistent with data provided earlier in the questionnaire. The shortest path through the questionnaire was 64 questions; the longest path was 88 questions.

The instrument was developed in four phases. A mock-up of the questionnaire was drafted in the first phase. In addition to the items needed to measure theoretical variables, many additional items were devoted to capturing descriptive and contextual data about the sites and their OOP-related activities. A few additional related items of interest to a research sponsor (e.g., what kinds of material most influence buying behavior) were also included.

In the second phase, the mock-up of the survey was successively refined, based on comments from a wide range of academics and market researchers, including:

- 1) academic colleagues,
- 2) diffusion researchers at other universities,
- 3) market research analysts at International Data Corporation, and
- 4) market researchers at the three initial study sponsors (all leading computer companies).

In the third phase, the refined mock-up was sent to a contractor, Para Technologies, for programming. The contractor developed the disk-based survey (DBS) using Sawtooth Software's Ci3 authoring system. A review of the "live" version of the instrument generated substantial stylistic changes to improve ease of use. Thorough tests of every version of the disk-based survey were performed, over half a dozen versions in total.

In the fourth and final phase, the disk-based survey was pretested at five Boston area firms: a bank, a financial services firm, a utility company, an insurance company, and a manufacturer. The goals of the pretests were to gauge survey length, to obtain feedback on the user interface, to obtain feedback on question wording and technical accuracy, and to identify instances where respondents had difficulty responding due to cognitive limitations or a lack of knowledge. These pretests proved to be an essential part of the DBS refinement process, as described below.

The pretests were conducted in two waves. First, three pretests were conducted and the survey was modified based on informant feedback. Then the last two pretests were held. Similar procedures were followed at all pretest sites. The respondents were told broadly about the research and its goals. They were then given the survey disks and asked to take the survey at their own pace, as they would normally work. The author was positioned behind the respondents so actual responses could be read as they were typed. When apparent hesitation or confusion was observed, or a response that appeared questionable was entered, a notation was made for later follow up. In addition, the respondents themselves were encouraged to speak out as they proceeded when they encountered especially troublesome items.

After completing the questionnaires, the respondents were asked to provide any feedback that occurred to them. They were then prompted for further information on items previously noted as potential trouble spots. Finally, the respondents were engaged in a general discussion covering such topics as what they thought of the survey, would they be likely to respond to such a

survey, what they thought of the survey length, and what elements of the survey design would make their participation more likely.

The pretests—especially the first three—provided invaluable feedback. The most common suggestions related to improvements in the user interface (e.g., conventions for moving around screens). Next most common were suggestions on question wording. Very few instances of technical inaccuracy or inability to answer surfaced, thus providing confidence that the prior reviews of the paper-based mockup were effective in identifying these sorts of shortcomings. Following the pretests, a new version of the DBS was produced, subjected to a final round of testing, and then "frozen".

3.3.3 Creation of Survey Sample

The sampling unit for this research is the information technology (IT) department at individual sites. The theoretical universe consists of all US-based sites of medium to large enterprises (over 500 total employees) that currently have on-site development of custom software applications for internal use. For each site, a manager knowledgeable about applications development at that site served as the key informant.

The sampling unit was defined to be the business site, rather than the company/division or parent company, because most of the variables and postulated relationships in this research have their most appropriate theoretical interpretation at the site level. Adoption decisions are typically made at the site level, and such concepts as Learning-Related Scale, Diversity, and Related Knowledge can vary markedly across sites, and hence, have a imprecise interpretation at any but the site level. Using a site level approach has an important additional benefit: informants are likely to be best informed about characteristics of their own sites.

The theoretical universe was defined to exclude organizations with fewer than 500 total employees because a high proportion of such sites have

minimal custom software development activities, and hence, are not very appropriate candidates for adoption of most software process innovations.²

Source List for Sampling Frame

The sampling frame was extracted from a list maintained by International Data Group on behalf of Computerworld magazine advertisers. Herein, this list will be referred to as the "CW Database." At the time the sampling frame was extracted, the CW Database had a total of 40,000 cases.

The goal of the CW Database is to provide Computerworld advertisers with good prospects for their sales efforts. The target informants are information technology professionals with buying influence or authority for hardware, software, or telecommunications products. As a result, the CW Database fully encompasses, but is not limited to, the kinds of informants desired for this research.

The CW Database was initially created in 1990, by sending an eight page questionnaire to a list of 400,000 people, at sites with at least 100 employees, who were either subscribers to one of nine IDG publications (e.g., Computerworld, Infoworld, PC World) or were on a pre-existing IDC database of sites with computers installed. The CW Database is updated on a nine month rolling basis. Every month, approximately 25,000 surveys are mailed to: 1) existing CW Database informants who haven't been contacted in nine months, and 2) new prospects, which come primarily from referrals from existing CW Database informants and from new subscribers to Computerworld. The typical response rate is 20-25% for new prospects, and 40-50% for existing informants.

²On the CW Database, more than 75% of sites of organizations with less than 500 total employment report fewer than 10 total IS employees.

Although no formal analyses have been done to establish how well the CW Database matches up with the universe of business sites in general, it is believed by IDG to reflect a proportional representation of all industrial sectors, with three exceptions:

- 1) The federal government is intentionally under represented, because government sites are less promising prospects for sale efforts (government purchases are regulated, and hence, government employees have less buying authority);
- 2) The educational sector is over represented because educators are more likely to participate in surveys in general;
- 3) Computer industry organizations are under represented, because of the difficulty in distinguishing employees that have authority over IT for internal use, from those involved with activities related to the development and marketing of the company's own products.

The CW database is well suited as a source for the sampling frame for this research because: 1) the criteria for including sites is compatible with the theoretical universe for the current research, 2) the database is regularly updated with "fresh" informants and current information, 3) the database contains informants that are likely to be knowledgeable about their site's applications development activities, 4) the database contains variables needed to create the sampling frame (see below) and to analyze response bias (see Section 3.3.6), and 5) the known disproportional representation of some sectors is not a significant threat to the research design.

Extraction Procedure to Create Sampling Frame

Cases had to meet four criteria to be extracted into the sampling frame:

- 1) The site was part of an organization with 500 or more total employees;
- 2) The site had at least one category of software development tools (e.g., programming languages, CASE tools) in place;

- 3) The informant had an information technology management related title;
- 4) The site had DOS or Windows-based computers in place.

The resulting sampling frame contained about 4000 sites. The first two criteria ensured that candidate sites matched the theoretical universe. The third criterion helped to ensure that candidate informants were those most likely to be well informed about the topics addressed in the survey—one of the main concerns when using key informant analysis. The last criterion, which eliminated only an incremental 5% of the sample, was included solely to exclude sites that were most likely to lack the means to take the disk-based survey, i.e., a DOS-based computer.

Fifteen hundred cases were selected from the sampling frame. Duplicates (multiple informants from the same site) were removed and replaced. The selection was random, except that sites that had previously reported "Object DBMS/tools" in place (in the survey used to construct the CW Database) were two times over-sampled. This was done to ensure a large enough N for planned analyses involving OOPL adopters only. To correct for this over sampling, case weighting was used in all statistical analyses. Specifically, cases were weighted so that the incidence of cases reporting "Object DBMS/tools" in place in the set of usable responses was the same as the incidence in the original sampling frame.

Several potential threats to the validity of this research were rendered less likely by the way in which the survey sample was constructed. The use of a large sample helps to ensure adequate statistical power, thus lowering the chance of a Type II error, one of the more prominent threats to statistical conclusion validity.³ The use of a probability sample eliminates a prominent

³The other main threat to statistical conclusion validity, violation of the assumptions of statistical tests, is addressed in Section 3.5.2.

internal validity threat, namely selection.⁴ The use of a source list that is consistent with the theoretical universe helps to manage threats to external validity.

3.3.4 Survey Administration Procedure

As with survey construction, the primary source for guidance on survey administration procedures was Dillman [1983]. On January 21, 1994, a survey packet containing a cover letter, a postage paid return mailer, and two alternative sized DOS-compatible disks (3.5 and 5.25 inch formats) were sent to each of the 1500 sample informants (one per site). The initial mailing was followed by a reminder postcard sent to all informants one week later. Two weeks later a follow-up letter reiterating the importance of participation was sent to all informants that had not yet responded. Data collection was cut off after six weeks.

Numerous strategies, including most of those recommended by Dillman, were employed to boost response rates and eliminate bias. These included:

- 1) Use of an individually addressed, hand-signed cover letter that emphasized the survey's university (MIT) affiliation, assured confidentiality, explained the social value of participation, and explained the importance of a high response rate;
- 2) Provision of generous incentives, including a chance to receive one of five \$100 lottery prizes (for those responding within two weeks) plus a copy of the research results and some additional technical reports (for all respondents);
- 3) Use of multiple follow up contacts, i.e., the aforementioned reminder card and follow up letter;
- 4) Use of other techniques designed to communicate the sponsoring organization's commitment and investment in the survey, including provision of an 800 number for questions, use of first class

⁴Other potential internal validity threats having to do with the direction of causation and the possibility of omitted common causes are addressed in Section 3.5.2.

postage on the out-going package, and provision of a postage paid, pre-addressed mailer for returning the disks.

The main elements of Dillman's Total Design method *not* employed were: 1) hand addressing of outgoing packages, and 2) a third follow up contact providing a duplicate copy of the DBS. Hand addressing was thought to be of little value for the receiving audience; a duplicate mailing of the DBS was prohibitively expensive.

The combination of techniques employed appear to have been quite effective. A total of 679 disks were returned out of 1500 for a raw response rate of 45%. This is an outstanding response rate for a survey as long and complex as this one, sent to (typically) busy managers with no affiliation to the sponsoring organizations. Although, as Dillman points out, survey recipients react to the "total package" rather than individual elements when deciding whether to participate in a survey, it seems likely that three elements in particular contributed to the high response: the mix of incentives, the MIT affiliation, and the DBS mode of administration.

3.3.5 Characteristics of Responding Sites and Informants

An analysis of the descriptive characteristics of responding informants and their sites was performed. This analysis was based on 608 usable responses out of a total of 679 returned disks (see breakdown of unusable responses in Table 3.3.5.1 below). As described below, the results were very favorable, suggesting a highly qualified pool of informants, and good coverage of the theoretical universe.

Table 3.3.5.1: Excluded Cases

Reason Not Usable	Number of Cases
Diskettes arrived after the data cut-off	20
Site not qualified (no custom development of software applications)	18
Informant not qualified (not a manager, or reported no knowledge of applications development).	21
Informant appeared to be unconscientious (multiple response sets with changing answers)	2
Survey incomplete	5
Informant apparently confused about how to respond to a crucial question ⁵	5
Total unusable cases	71

Characteristics of Responding Informants

Three questions were included in the survey for the express purpose of supporting a post-hoc evaluation of the qualifications of informants to respond to the survey. The ideal informant was set out in advance to be a middle level IT manager, who was well informed about the applications development technologies and activities at their site. The responding informants overwhelmingly matched these criteria (see Tables 3.3.5.2 through 3.3.5.4 below). Nine of ten reported being managers or directors. Eighty-four percent reported being "very" or "extremely" knowledgeable about applications development. Nine of ten informants also reported holding responsibility for at least 10% of applications development activities at their

⁵These five informants, when presented with a list of six OOPLs, selected all six as having been acquired, yet later reported that none of the languages had ever been approved for production use. Four of five of these respondents also reported being only "slightly familiar" with OOP. This is an extremely implausible joint set of results, especially considering the very low overall level of penetration (less than 1 %) for two of the languages (Eiffel and CLOS). The most likely explanation for this apparent anomaly is that these respondents thought they were merely cursoring down the list of OOPLs, when in fact they were selecting each OOPL. Even though selected items get highlighted and unselected items do not, this convention, though explained earlier in the DBS and a common feature of modern user interfaces, might not have been salient to these respondents since all items ended up the same color (i.e., highlighted).

location; 74% reported responsibility for at least half of applications development activities.

Table 3.3.5.2: Informant Managerial Level

Category	Count	Percent
VP or above	8	1.2
Director	232	38.2
Manager	318	52.3
Supervisor	17	2.8
Other Managerial	17	2.8
Project Leader	16	2.6

Table 3.3.5.3: Knowledge of Applications Development

Category	Count	Percent
Slightly	11	1.8
Moderately	86	14.1
Very	213	35.0
Extremely	298	49.1

Table 3.3.5.4: Scope of Responsibility for Applications Development

Category	Count	Percent
None	24	4.0
Less than 10%	42	6.9
10% to 24%	38	6.3
25% to 49%	54	8.8
50% to 75%	79	13.0
75% to 100%	371	61.0

Informant attitudes towards the DBS itself were quite positive, as evidenced by the fact that nearly 40% reported being "much more willing" to take future disk-based surveys, while only 4.5% were less willing (see Table 3.3.5.6).

Table 3.3.5.6: Willingness to Answer Future Disk-based Surveys

Category	Count	Percent
Much more willing	236	38.9
Somewhat more willing	136	22.4
About the same	208	34.1
Somewhat less willing	15	2.4
Much less willing	13	2.1

Characteristics of Responding Sites

Responding sites overwhelmingly fit the theoretical universe of this study. As mentioned above, only 20 responding sites were unqualified because of a complete absence of applications development activities and were therefore dropped from further analysis. Of qualified sites, 97% reported that custom applications had been developed at their site for at least 3 years; 69% reported custom software development for at least 10 years (see Table 3.3.5.7).

Table 3.3.5.7: Years of Development

Category	Count	Percent
Less than one year	4	.7
1 to 2 years	11	1.8
3 to 4 years	43	7.2
5 to 10 years	130	21.3
Greater than 10 years	420	69.1

The responding sites span a wide range of sizes and types (see Tables 3.2.5.8 through 3.2.5.10 below). The range of reported total host organization employment (at all sites) is 1 to 200,000, with median of 1,200. Eighteen percent are sites within large companies that have a total employment over 5,000. A wide range of industrial sectors are represented, including not-for-profit sectors such as government (14%), education (9%), and medical (6%); The largest sectoral representation is in manufacturing with 39%. The size of the total on-site IT staff (including development, technical support and

operations) ranged from 1 to 3,000, with a median of 16. Smaller departments are included (16% reported fewer than 5 total staff members) and well as very large (13% reported over 100 staff members). Thirty-eight percent of sites reported an applications development staff at more than one location.

Table 3.3.5.8: Total Employment

Category	Count	Percent
Less than 250	30	5.0
250 to 499	30	5.0
500 to 999	189	31.1
1000 to 2499	180	29.6
2500 to 5000	72	11.8
5000 to 9999	53	8.8
10,000 or more	54	8.8

Table 3.3.5.9: Industrial Sectors

Category	Count	Percent
Manufacturing	237	39.0
Banking/finance/insurance	30	5.0
Transportation/communications/utilities	37	6.1
Wholesale/retail	37	6.1
Health/medical	56	9.1
Education	52	8.5
Government	87	14.4
Other	72	11.8

Table 3.3.5.10: Total IT Staff at Site

Category	Count	Percent
Less than 5	96	15.8
5 to 9	96	15.7
10 to 24	170	28.0
25 to 49	89	14.7
50 to 99	76	12.5
100 to 249	54	8.8
250 or more	28	13.3

The vast majority of responding sites appear to be typical MIS organizations. Eighty-one percent reported mainframes or midrange computers as their primary host environment for applications development, whether accessed via terminals (40%) or terminal emulation (41%). Ninety-two percent of sites reported that online MIS/transaction processing comprises at least 5% of their development activity.

3.3.6 Analysis of Response Bias

One of the potential limitations of large scale surveys are their typically low response rates, and resulting higher exposure to response bias, i.e., systematic differences between respondents and non-respondents. Response bias is a problem only to the extent that it limits the researcher's ability to generalize findings to the survey sample as a whole, and by extension, the survey's theoretical universe. With a 45% response rate, the potential exposure of this research to a response bias is less than most large scale surveys, although it is still worthy of attention.

Fortunately, this research had a unique resource available to help determine the extent and substantive importance of response bias, namely, the variables in the original CW Database from which the sample was drawn. Most surveys, lacking an *a priori* database, must resort to more indirect methods for assessing bias, such as comparing early respondents to late respondents using variables collected by the survey itself.

Four variables were selected from the CW Database to support an analysis of response bias:

- 1) IS Employees: The size of the information technology function in the company at large, as measured by the number of IS employees;
- 2) Host Size: The size of the host organization, as measured by total employment;
- 3) Sector: The primary industrial sector;

- 4) OO Adoption: a rough indicator of the likelihood of adoption of object technology (a binary variable coded to “yes” for sites reporting “object-oriented DBMS/tools in use” on mainframe, mid-range, or desktop computers).

The first three variables were selected because they are among the most prominent demographic characteristics of respondents. The fourth variable, OO Adoption, serves as a rough indicator of adoption of OOPLs. A fifth variable was created, Response-Yes, and coded to 1 for respondents and 0 for non-respondents.

Biases Affecting Descriptive Inferences

The first step in the response bias analysis was to check for biases that affect the ability to generalize descriptive results, such as the proportion of respondents in later assimilation stages. A chi-square test revealed a significant negative relationship ($p=.01$) between IS Employees and Response-Yes. The response rate ranged from a high of 47% for sites with 10-24 IS employees in the company at large, to a low of 27% for sites with more than 500 IS employees in the company at large. This relationship was expected: managers in large organizations are more likely to have their mail screened by assistants, and because they are of greater interest to vendors, such managers are surveyed more frequently, which leads to a greater reluctance to participate in any particular survey.

Significant relationships were also found between Response-Yes and both Sector ($p=.02$) and OO Adoption ($p=.01$). Relationships with the latter variable disappeared after controlling for IS Employees, however, suggesting that the direct association is spurious for OO Adoption. No association was found between Response-Yes and Host Size ($p=.39$).

The main descriptive results of substantive interest are the distribution of object-oriented programming languages across assimilation stages, and the extent of adoption of three historical SPIs—relational DBMS, 4GLs and Upper

CASE tools. The likely bias related to OOPLs was corrected by adjusting the weights of the responding cases so that the incidence of OO Adoption in the respondent sample matched the incidence of OO Adoption in the sampling frame. Nevertheless, slight biases probably remain for the three historical SPIs due to the under representation of sites within companies with larger IS functions. The result of this is to potentially underestimate the true proportion of sites adopting these technologies (since sites with larger host organization IS functions are more likely to, other things equal, adopt these technologies.)

Biases Affecting Inferences About Associations

Of greater concern to this research are response biases that have the potential to affect inferences about associations between independent and dependent variables. For example, it might be that respondents are systematically different from non-respondents, in a way that leads to spurious conclusions about the relationship between these variables and OOPL assimilation.

To check for this kind of bias, a logistic regression analysis was performed. The dependent variable was OO Adoption (a binary measure of whether any object-oriented tools are in use), and the independent variables were IT Size (an index including the number of IS employees and the level of IS spending in the organization as a whole), Host Size (the number of employees in the host organization) and Government (a dummy variable). Of the available variables, these three map most directly to predictor variables included in the causal model tested later in this chapter. The equation was estimated first for the sample as a whole, and then just for respondents. Individual parameter estimates were quite similar, as shown in Table 3.3.6.1 below. As a result, it seems unlikely that there is a significant response bias affecting inferences about associations

Table 3.3.6.1: Logit Model Parameter Estimates

Variable	All Cases	Respondents
Host Size	.063	-.014
IT Size	.323*	.349*
Government	-.343	-.300

*Significant at $p=.05$

3.3.7 Results of Telephone Survey of Non-Respondents

A 14 item telephone questionnaire was developed to support analysis of the reasons for non-response. One hundred cases were randomly selected from among those who had not responded as of four weeks after the OOPL disk survey mailing date. Data collection was terminated after establishing contact with a total of 60 sites. This required almost three complete passes through the randomly ordered list of cases. For the other 40 cases where the informant was never reached, it was at least established that the informant did still work at the site.

Of the 60 contacted sites, no interview was conducted in 18, either because the intended informant had left the company (N=7), the survey had in fact recently been returned (N=8), or the respondent refused to be interviewed (N=3). (See Figure 3.3.7.1 below.) Of the 42 sites where interviews were held, 4 were only willing to stay on the phone long enough to give a broad reason for non-response, 9 did not remember receiving the disk, and 4 had forwarded it to another manager.

An analysis of the primary reason for non-response was performed for 29 respondents, those that remembered the disk and had not forwarded it (N=25), plus those that would only provide the reason for non-response (N=4) (see Table 3.3.7.1 below). It was found that neither questionnaire design itself or the disk-based method of administration was likely to have introduced bias into who chose to respond. Only 3 non-respondents

volunteered anything specific to the questionnaire as the primary reason for non-response (i.e., "survey looked like junk". "not interested in the questions", "last disk-based survey took two hours"). "Lack of time" was by far the most commonly stated reason for non-response, with sixty-five percent (N=19) giving this as their primary reason.

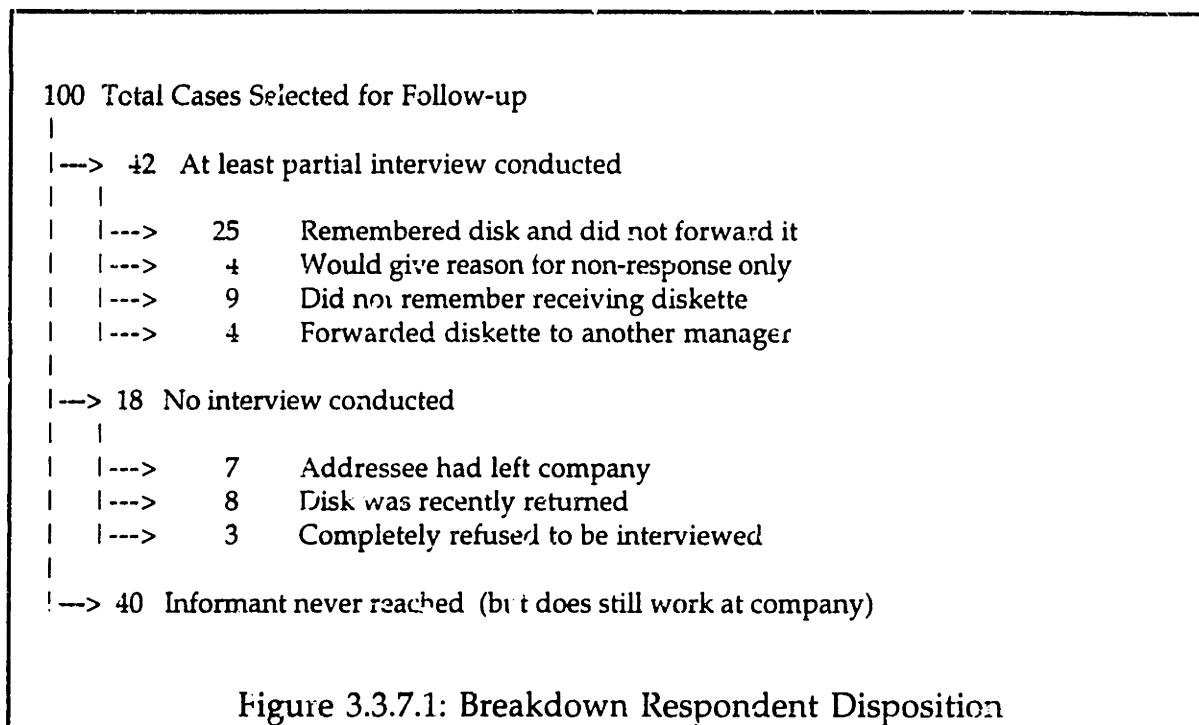


Table 3.3.7.1: Primary Reasons for Non-response (N=29)

Reason	Number Selecting
Lack of time to complete the survey/too busy	19
Don't generally participate in surveys	3
No applications development performed at site (not qualified)	2
Something about this survey in particular bothered me	3
Other —"been out of office" "not the right person"	2

The 27 respondents who indicated that something other than the characteristics of the survey itself prevented their response were further probed as to whether anything about the survey did, in fact, bother them at all. After probing, two of these respondents mentioned concerns with the

disk-based mode, one citing fears of viruses and the other citing an inability to gauge survey length.

Of the 34 that did not forward the diskette, all had IBM-compatible PCs on their desks at work (N=29) or readily available to them (N=5), suggesting that a lack of access to necessary equipment—a major potential concern with the DBS mode—was not a primary cause of non-response.

3.3.8 Method Bias Follow-up Survey

A method bias has occurred when statistical relationships between variables are due to the method used to measure the variables rather than relationships between the constructs they represent. A method bias is most likely to be a problem in studies that, like this study, use a single instrument administered to a single informant to contemporaneously measure both independent and dependent variables. (Having different informants provide data associated with the independent and dependent variables was not a feasible option for this research.) The specific concern is that respondents will infer the "expected" relationships between variables and then shade their answers to be more consistent with the expected relationships.

One test of whether a method bias might exist is to gather data on just the dependent variable, at a later time, from the same informant [Venkatraman 1993]. The idea is that at this later time the expected relationships would not be as salient to the informant, and hence, less, if any, shading would occur. The model is then estimated using the dependent variable at Time 2, and an analysis is performed to determine whether the expected theoretical relationships still hold.

A paper-based follow up survey was performed to support an analysis of method bias. A one-page, 12 item survey was constructed to measure Assimilation Stage. Eight weeks after the original disk-based survey had been mailed, the follow up surveys were mailed to the 608 informants who had

provided usable responses to the original survey. A total of 411 surveys were returned within 6 weeks, for a raw response rate of 68%. Of the 411, nineteen responses were not usable because of incomplete or inconsistent responses, and nine were classified as rejectors. This left a net of 384 cases for which Assimilation Stage at Time 2 could be computed.

A comparison of Assimilation Stage at Time 1 (Stage T1) and Time 2 (Stage T2) revealed a high degree of consistency between the two. The two measures were correlated at $r=.72$, and in 89% of the cases, the Stage T2 was either the same as Stage T1, or differed by only one position. Further analysis to test for the presence of a method bias is presented in Section 3.5.2.

3.4 MEASUREMENT

This section presents the measures used to operationalize the primary dependent construct (Assimilation Stage) and the three independent constructs (Learning-Related Scale, Related Knowledge, and Diversity). In addition, the measures for a set of six control variables, included to reduce noise and eliminate confounds, are described. Following a description of these measures, the quantitative components of construct validity are analyzed for the independent constructs.

3.4.1 Assimilation Stage

The primary dependent variable for this research is Assimilation Stage achieved by a certain date. This research follows Meyer and Goes in viewing the innovation process in organizations as proceeding through "stages of assimilation," and in viewing Assimilation Stage as a more robust measure of organizational innovativeness than traditional measures such as earliness of adoption [1988]. Assimilation Stage may be viewed as a combined measure of earliness of initiation of assimilation activities, speed of assimilation activities, and an absence of rejection, stalling or discontinuance. As will be argued in detail in Chapter 4, Assimilation Stage provides a unique

combination of both generality and richness, and avoids many of the particular limitations of alternative innovativeness measures. Furthermore, Assimilation Stage is especially appropriate when studying emerging technologies such as OOPLs, because it alone among commonly used innovativeness measures captures gradations of innovativeness among those that have yet to adopt.

A six stage model of SPI assimilation is proposed (see table 3.4.1.1 below). The proposed model departs from Meyer and Goes in both the number and exact definition of assimilation stages because of contextual differences between the medical innovations they studied and SPIs. Basically, SPI assimilation is much less of a formal and bureaucratic process, and SPI innovations are less "packaged".

Table 3.4.1.1: Definition of Assimilation Stages

Stage	Criteria to enter stage	Typical behaviors
1. Awareness	Key decision makers are aware of the SPI	Passive, accidental or opportunistic learning about the SPI through media, word of mouth, vendor promotions, etc.
2. Interest	The organization is committed to actively learning more about the SPI	Directed exploration of specific products and adoption issues Matching products to needs
3. Evaluation/trial	The organization has acquired a specific innovation-related products and has initiated evaluation or trial	Acquisition and installation of tools Hands-on evaluations Trial projects
4. Commitment	The organization has committed to use a specific SPI product in a significant way for one or more production projects	Initial development of production projects Training and infrastructure development
5. Limited deployment	The organization has established a program of regular but still limited use of the SPI product	Development and implementation of selected production systems using the SPI Continued development of infrastructure
6. General deployment	The organization has reached a state where the SPI is used on a substantial fraction of new development, including at least one large and one mission critical system	Ongoing use on a wide variety of projects, including large, mission critical ones Continued development of infrastructure

This proposed model is similar to the one employed by Ettlie [1980] in his analysis of the adequacy of stage models, but with two important differences. First, "trial" and "evaluation" are combined into a single stage (Evaluation/Trial) because of the finding by Ettlie that frequently only one of these stages occur, or they occur simultaneously. Second, Ettlie's "implementation" stage is divided into two stages (Limited Deployment and General Deployment) to distinguish the fact that in many cases an SPI may be deployed on a few projects but never make the transition to a regularly used technology.

Another interesting stage model was employed by Cooper and Zmud in a study of material requirements planning [1990]. Their model identifies six stages of IT implementation—initiation, adoption, adaptation, acceptance, routinization, infusion. The proposed model contrasts with this model in two main ways. First, the proposed model has more pre-adoption stages. Second, Cooper and Zmud's later three stages are viewed not as linear stages, but as ongoing post-adoption processes that occur (or don't) concurrently with wider deployment/diffusion of the technology. This seems especially true in the context of SPIs where there may not be a single, monolithic "implementation" effort. Table 3.4.1.2 provides a mapping between the proposed model and those used by Ettlie, Meyer and Goes, and Cooper and Zmud.

Table 3.4.1.2: Comparison of Stage Models

SPI Assimilation Stage (Fichman)	Adoption Stage (Ettlie)	Medical Innovation Assimilation Stage (Meyer and Goes)	IT Implementation Stage (Cooper and Zmud)
1. Awareness	1. Awareness	1. Apprehension	1. Initiation
2. Interest	2. Interest	2. Consideration 3. Discussion	
3. Evaluation/trial	3. Evaluation 4. Trial	4. Acquisition proposal 5. Medical fiscal evaluation 6. Political strategic evaluation 7. Trial	
4. Commitment	5. Adoption	8. Acceptance	2. Adoption
5. Limited Deployment	6. Implementation		3. Adaptation 4. Acceptance
6. General Deployment			9. Expansion

Table 3.4.1.3 below provides a description of the Guttman Scale used to operationalize Assimilation Stage for the case of OOPLs. Organizations were classified according to the highest stage achieved as of the time the survey was administered.

Table 3.4.1.3: Guttman Scale for Assimilation Stage

Stage	Criteria to enter stage	Items Used to Classify (See Exhibits 1 through 4)
1. Awareness	Key decision makers are aware of the SPI	Is informant familiar with OOPL concepts or products, or aware prior OOPL-related activities at site? (See OOPL Survey items E-02, E-03)
2. Interest	The organization is committed to actively learning more about the SPI	Is informant aware of plans to investigate any OOPL for possible production use within next 12 months? (See F-01, E-09)
3. Evaluation /trial	The organization has acquired a specific innovation-related products and has initiated evaluation or trial	Has site acquired any OOPL? Is the site evaluating or trialing any OOPL? (See E-04, F-01)
4. Commitment	The organization has committed to use a specific SPI product in a significant way for one or more production projects	Are any specific production projects planned, in progress, implemented or canceled that use an OOPL as primary language? (See E-08, E-10)
5. Limited deployment	The organization has established a program of regular but still limited use of the SPI product	Have at least three projects been initiated? (See E-10) Has at least one project been completed? (See E-10)
6. General deployment	The organization has reached a state where the SPI is used on a substantial fraction of new development, including at least one large and one mission critical system	Have at least three projects been completed? (See E-10) Has site has implemented at least one large OOPL project requiring at least 12 person months (See I-09) Has one or more core or mission critical applications been completed? (See I-11) Has there ever been a year where at least 25% of new application developments used OOPL? (See I-14)

This model assumes a linear progression through assimilation stages. However, organizations may diverge from this progression in two ways, either by evaluating and rejecting OOPLs, or by adopting, but later discontinuing the use of OOPLs. Additional items were included in the survey to identify rejectors and discontinuers (see Table 3.4.1.4). Since such organizations might well possess distinctive properties, a separate analyses is needed that focuses just on rejectors and discontinuers. In point of fact, no organizations in this study had discontinued use of OOPLs; however, several had rejected, and an analysis of the properties of these organizations is provided in Section 3.5.2.

Table 3.4.1.4: Rejection and Discontinuance

Category	Description	Items Used to Classify (See Exhibits 2 and 3)
Rejection	The organization has evaluated and rejected SPI for current use.	Has the site previously evaluated and rejected OOPLs? (See E-09)
Discontin- uance	The organization committed to using the SPI at some point in the past, but is not now using it, and does not foresee using it in the future.	Has the site ever approved an OOPL for production use? (See E-08) Are there no projects currently pending? (See E_10) Has the site discontinued use? (See E-12)

3.4.2 Learning-Related Scale

As argued in Chapter 2, an organization possesses greater Learning-Related Scale when it has a better opportunity to amortize the learning costs associated with technology assimilation, owing to the scale of subsequent activities. More formally, Learning-Related Scale is defined here *as the scale of activities over which learning costs can be spread.*

To operationalize this concept, three indicators were captured, as presented in Table 3.4.2.1. These indicators focus on new development because this is the domain in which SPIs (including OOPLs) are primarily applied. The first two

indicators are volume-related measures, and are most analogous to the conventional economic notion of scale. The last indicator is a "density" measure intended to capture the rate at which applications are being replaced. The logic here is that a greater density of activities might hold benefits because individuals and teams are applying the new technology more frequently, on average, and hence, have a better opportunity to more rapidly move down their own learning curves with the new technology. Research in other contexts has shown that the benefits of learning-by-doing can degrade if the activity is interrupted [Argote *et al.* 1990].

Table 3.4.2.1: Definition of Learning-Related Scale (LRS) Indicators

Indicator	Description	Specific calculation (See Exhibit 5)
LRS1	The volume of development of completely new applications	% of development for new systems (P-01.3) times total number of application developers (R-00b)
LRS2	The volume of development of new applications together with significant enhancements to existing systems	% of development for new systems and enhancements (P-01.2 plus P-01.3) times total number of application developers (R-00b)
LRS3	The rate at which the current applications portfolio being redeveloped or replaced	Average of % of applications replaced over last three years (P-03) and % of applications to be replaced over next three years (P-04)

While, in principle, many organizations could eventually amortize SPI-related learning costs, those with a greater volume and frequency of new applications development can amortize costs more quickly. This is particularly important because: 1) faster amortization improves return on investment; 2) managers often have little patience with investments that do not show a rapid pay-back; 3) intensity of effort is crucial to effective learning, yet little cumulative learning will occur if application of the new technology is infrequent; and 4) if a firm does not move along the internal learning curve quickly enough, then its process will be surpassed by the external learning curve (i.e., the firm could have established an efficient process faster by waiting for more mature technology vintages to become available).

3.4.3 Related Knowledge

Related Knowledge is defined here as *the extent of abstract knowledge, know-how and skills possessed by the organization in areas related to the focal innovation*. Unlike the other two independent constructs, the measure for Related Knowledge will always be SPI-specific. As argued in Chapter 2, Related Knowledge can take at least three forms, including: 1) knowledge of the abstract principles on which the SPI is based and the skill to apply them; 2) expertise in the use of the particular products embodying the SPI, or similar products; and 3) expertise in the technologies that typically "cluster" around the SPI. The approach to measurement employed by this study focuses primarily on the last category of Related Knowledge (See Table 3.4.3.1), although in doing so, some elements of the first category are also captured.

Specifically, this study operationalizes Related Knowledge as the extent of development staff experience in four areas: 1) C language programming, 2) client-server application development, 3) PC/workstation-based applications development, and 4) development of graphical user interfaces (GUIs). These four areas were identified through discussions with OOP experts. In addition, the mini-case studies of OOP adopters provide confirmation of the face validity of these areas.

Table 3.4.3.1: Definition of Related Knowledge (RK) Indicators

Indicator	Description	Specific measure (See Exhibit 7)
RK1	Knowledge of C language syntax and related tools	Percentage of development staff with experience programming using C (O-03.1)
RK2	Knowledge of client-server applications development	Percentage of development staff with experience developing client-server applications (O-03.2)
RK3	Knowledge of PC/workstation-based development	Percentage of development staff with experience developing PC or workstation applications (O-03.3)
RK4	Knowledge of GUI development	Percentage of development staff with experience developing graphical user interfaces (O-03.4)

Experience with the C Language

The C language is a central part of the cluster of technologies and skills surrounding commercially available OOPLs. C++, a C language hybrid, holds nearly 86% of the market for OOPLs.⁶ In the current study, C++ had been acquired by 94% of those that had acquired any OOPL.

An organization with more extensive prior experience with C who adopts C++ (or another hybrid such as Objective C) is already proficient in the sizable subset of the language that is common to C, and already possesses expertise in common supporting technologies, such as operating systems and utilities. It is no accident that C++ was developed in the first place, or that it has become dominant: the *raison d'être* for C++ is to allow individuals and organizations to leverage their existing knowledge of the popular C language and associated technologies.

A study conducted by IBM provides empirical confirmation of the link between C programming experience and proficiency in learning object-oriented programming [Liu *et al.* 1992]. This study found that upon completing a week long IBM course in C++ students with prior experience programming in C were judged (by instructors and lab assistants) as significantly more proficient in object-oriented programming compared to those who did not have C in their background. This effect was overwhelming compared with other student characteristics. Interestingly, C language experience was also a significant (albeit weaker) predictor of rated proficiency when the OOPL used in the course was Smalltalk, rather than C++. The authors of the IBM study believe that this is because C programmers are likely to be more "sophisticated" programmers on average. Another, not unrelated explanation is that C programmers are more likely to have acquired at least some knowledge of object-oriented programming in

⁶Personal communication, Steve McClure, International Data Corporation, December 1994.

general, because of a natural interest in the prominent coverage of given to C++, a C-based object-oriented hybrid, in journals and trade publications.

Experience with PC/Workstation Development

To date, commercially available OOPLs, including C++ and Smalltalk, have been implemented almost exclusively on PC and workstation platforms. Not unrelatedly, OOPL applications nearly always include a PC or workstation-based runtime component. Would-be adopters who have expertise in PC or workstation-based development already possess skills in the use of the operating systems, windowing shells, utilities and so forth comprising these environments, and so are spared the burden of acquiring this knowledge in order to adopt an OOPL.

Experience with Client-Server Development

Medium or large scale multi-user applications form the core applications in a typical IT department. The most common architecture for an OOPL-based multi-user application is client-server, with the OOPL resident on the PC or workstation-based client. This is because, currently, there are few (if any) mature OOPL implementations on larger computers (mini or mainframes), and no other commonly used architecture supports both the use of an OOPL and multi-user access to data. In point of fact this was the architecture employed by all three of the organizations in the previously mentioned case studies that developed multi-user systems with an OOPL. (In the fourth organization, only a stand-alone pilot system was built with the OOPL). Individuals and organization with prior experience in client-server are spared the burden of acquiring this expertise during the course of adopting an OOPL. Another advantage of client-server experience is that the message passing paradigm for inter-module communication in client-server applications is also used in object-oriented applications, thus providing potential adopters with skills related to this object-oriented principle.

Graphical User Interfaces

The linkage between OOP and development of graphical user interfaces (GUI) stretches back to the 1970's, where Xerox PARC pioneered the GUI concept in applications built using Smalltalk [Thomas 1989]. In the present day, OOPL applications, whether stand-alone or based on client-server, typically include a GUI. This was true for all four case studies of OOPL adopters conducted by the author. There is also an overlap in terms of abstract principles and associated skills, because like OOPL applications, GUIs and are event-driven and involve manipulation of objects (e.g., screen icons).

Summary

To summarize, the C language, client-server, PC/Workstation-based development, and GUIs are all part of the OOPL technology "cluster," and additionally, overlap in terms of some specific OOP concepts and skills.

3.4.4 Diversity of Knowledge and Activities

Diversity of organizational knowledge and activities contributes to organizational learning surrounding SPIs by making it easier for individuals to appreciate and absorb new information that is encountered; by increasing the likelihood that at least one domain will exist that represents a good fit with the SPI; and by providing an opportunity to bootstrap, as the organization become more proficient with the SPI, from higher fit areas to areas with a lower fit. More formally, this research defines Diversity as *the degree of heterogeneity of organizational knowledge and activities (technical, functional, and business) in areas related to applications development.*

This research employs four indicators for this construct: 1) diversity of programming languages, 2) diversity of runtime platforms, 3) diversity of application architectures, and 4) diversity of business sectors supported (See Table 3.4.4.1 below).

Table 3.4.4.1: Definition of Diversity (DIV) Indicators

Indicator	Description	Specific calculation (See Exhibit 6)
DIV1	Diversity of programming languages employed	The number of different programming languages used by at least 5% of the development staff in 1993. Categories presented: 1) assembly, 2) Cobol, 3) C, 4) another 3GL, and 5) traditional 4GL (Q-05a.1 thru Q-05a.5). ¹
DIV2	Diversity of runtime platforms employed	The number of different runtime platforms accounting for at least 5% of new development for over the past 3 years. Categories presented: 1) traditional mainframe, 2) traditional midrange, 3) client-server with mainframe host, 4) client-server with midrange host, 5) client-server with desktop host, 6) networked peer-to-peer workstations/PCs, 7) standalone workstations or PCs (Q-02.1 thru Q-02.7).
DIV3	Diversity of applications architectures employed	The number of different application architectures accounting for at least 5% of new development for over the past 3 years. Categories presented: 1) primarily batch MIS/transaction processing, 2) primarily online MIS/transaction processing, 3) information retrieval/reporting/query/DSS, 4) real-time or process control, 5) engineering/scientific/modeling 6) office automation/personal productivity/groupware (Q-03.1 thru Q-03.6).
DIV4	Diversity of business sectors supported with IT applications	The number of distinct industrial sectors supported by applications development by the applications development staff. Fifteen standard categories presented (Q-01a.1 thru Q-01a.15).

¹In the same question, data were also gathered on the percentage of developers using OOPs, but this was not included in the indicator because of obvious confounding problems with OOPs Assimilation Stage.

The basis for selection of these particular languages and runtime platforms was the experience of IDC market researchers with surveys of this kind, and the author's own professional experience. The five programming language categories were selected to provide a parsimonious set of the most prominent languages typically in use in IT departments. Likewise, the seven runtime platforms were intended to provide a parsimonious set of the most common platforms typically in use. The six application architectures were developed based on a framework previously developed by Rumbaugh *et al.* [1991]. The

business sector categories are based on the Standard Industrial Classification (SIC) codes.

3.4.5 Control Variables

Several variables commonly employed in prior studies of innovation diffusion were also captured to eliminate noise and to control for potential confounds (see Table 3.4.5.1). These include: host organization size [Rogers 1983], size of the IT function [Swanson 1994], specialization [Damanpour 1991], IT staff educational level [Zmud 1982], environmental complexity [Cooper and Zmud 1990], and sector [Bretschneider and Wittmer 1993]. These are viewed as control variables because they fall outside of the scope of the theory presented in Chapter 2. All of these variables are expected to have positive relationships to Assimilation Stage, with one exception: government organizations are expected to be less innovative with regard to SPIs, because of the finding by Bretschneider and Wittmer that government organizations were less innovative with respect to another information technology (microcomputers) [1993].

The primary potential confound to be controlled is related to size: since it is likely that both Learning-Related Scale and Diversity covary with organizational size—which itself been empirically linked to innovation—organizational size is of special concern. Tornatzky *et al.* [1983] have argued that size *per se* has no compelling rationale linking it to innovation, and more likely serves as a proxy for other variables, such as slack resources, education and professionalism, specialization, and, especially, scale.⁷ Included variables provide both direct (Host Size) and indirect (IT Size, Education,

⁷ Scale differs from size in that scale is defined in relation to some particular activity. While large size is usually necessary for large scale, it is not sufficient. A large machine tool manufacturer, for example, may have a large scale manufacturing operation, but a small scale marketing operation. Conversely, a large movie distribution company might have a large scale marketing operation, but a small scale manufacturing operation.

Specialization) control for this potential confound, and are expected to adequately address this validity threat.

Table 3.4.5.1: Definition of Control Variables

Variable	Description	Specific measure (see Exhibits 8 thru 10)
Host Size	Size of the host organization	Log of the number of employees in the host organization (R-04)
IT Size	Size of the IT function throughout the host organization	Mean of two categorical variables for all sites within the informant's span of influence, one capturing number of IT employees, and a second capturing level of external IT spending (CW-01, CW-02)
Specialization	Degree of specialization of the IT staff at the site	Sum of the number of specialties for which the site has at least one full time staff member. Categories were: technology evaluation, quality assurance, data administration, methods & tools, metrics & measurement, system testing (Q-04.1 thru Q-04.6)
Education	Level of education of the IT staff at the site	Mean of two variables, the percentage of IT staff at the site holding bachelor's degrees (O-01.1+O-01.2), and the percentage holding master's degrees or higher (O-01.2)
Environmental Complexity	Extent of environmental complexity	Mean importance of seven typical objectives for systems development (rapid, cost effective, keep to budgets & schedules, high performance, high reliability, easy to use, easy to change) (S-01.1 thru S-01.7)
Government	Public or private sector	Binary variable identifying sites where government is the primary industrial sector.

3.4.6 Quantitative Analysis of Construct Validity

Construct validity is the extent to which operational indicators faithfully map to their higher order constructs. Bagozzi proposes six components of construct validity [1980]:

- 1) **Theoretical Meaningfulness of Concepts:** The language used to represent the concept is internally consistent and adequate to describe the scope and range of the concept.
- 2) **Observational Meaningfulness of Concepts:** The operationalizations used to measure the concept have a strong semantic linkage to the concept itself: e.g., questions are clear, unambiguous, and actually tap

into the concept. (This is similar to the traditional notion of content validity.)

- 3) Internal Consistency of Operations: Indicators are reliable and correspond to a single (i.e., unidimensional) concept.
- 4) Convergent Validity: Two or more indicators of the same theoretical concept, measured with maximally dissimilar methods, are in agreement.
- 5) Discriminate Validity: Indicators of different concepts differ.
- 6) Nomological Validity: Predictions from a formal theoretical network containing the concepts are confirmed, i.e., they predict, and are predicted by, the expected theoretical concepts. (This similar to the traditional notion of predictive validity.)

The first two components of construct validity cannot be evaluated quantitatively, but rather, are ensured by building on existing theory and by careful attention to question construction. It is believed that the theoretical base established in Chapter 2, and the questionnaire development procedures described in Section 3.3.2, have ensured theoretical and observational meaningfulness of concepts.

The remainder of this section describes the procedures used to evaluate the quantifiable components of construct validity, namely, internal consistency, convergent validity, discriminant validity, and nomological validity, for the three independent constructs—Learning-Related Scale, Related Knowledge and Diversity. As a first step, several traditional approaches to evaluating construct validity were employed, including inspections of within-construct zero-order correlations, computation of Cronbach's alpha, and common factor analysis with oblique rotation. Next, a confirmatory factor analysis was performed using the structural equation modeling approach with Lisrel. Finally, a cursory check of nomological validity was performed by analyzing zero-order correlations of the three independent constructs with each other and with Assimilation Stage.

Inspection of Within-Construct Correlations

Within-construct correlation matrices were computed using standardized indicators of each of the three theoretical variables (see Tables 3.3.6.1 through 3.3.6.3 below.)

Table 3.4.6.1: Correlation Coefficients for Learning-Related Scale (LRS)

	LRS1	LRS2	LRS3
LRS1	1		
LRS2	.92*	1	
LRS3	-.04	-.07	1

Table 3.4.6.2: Correlation Coefficients for Related Knowledge (RK)

	RK1	RK2	RK3	RK4
RK1	1			
RK2	.33*	1		
RK3	.45*	.37*	1	
RK4	.44*	.43*	.50*	1

Table 3.4.6.3: Correlation Coefficients for Diversity (DIV)

	DIV1	DIV2	DIV3	DIV4
DIV1	1			
DIV2	.39*	1		
DIV3	.30*	.35*	1	
DIV4	.12*	.20*	.07	1

* significant at $p \leq .05$

Based on an examination of these matrices, two indicators, LRS3 and DIV4, were eliminated from the measurement model. Both indicators had very low average correlations with other indicators measuring their respective constructs (well under $r=.2$), suggesting they do not load on these constructs as had been expected. In addition, when these indicators were included as separate variables in preliminary regressions, they were non-significant, and had no effect on the pattern of significant coefficients for other variables, or on the variance explained.

Computation of Cronbach's Alpha

The Cronbach's alpha for the reduced Learning-Related Scale, Related Knowledge and Diversity scales were .96, .74 and .62, respectively. While the values for Learning-Related Scale and Related Knowledge are well within the standard rule-of-thumb (Cronbach's alpha above .7), the value for Diversity is a bit low, suggesting particular attention should be given to other techniques for analyzing convergent reliability provided below.

Common Factor Analysis

Although common factor analysis is traditionally used for exploratory analysis of underlying factors, it can also be used to gain confirmatory insights into convergent and discriminant validity if criteria for acceptance are set out *a priori*. For this research, the criteria were:

- 1) Do three interpretable factors emerge?
- 2) Do all indicators load most heavily on the expected factors?
- 3) Are within-construct factor loadings reasonable (i.e., above .5)?

As an inspection of Table 3.4.6.4 below reveals criteria 1 and 2 were met, with exactly three factors emerging, each corresponding to an *a priori* construct, and with no cross factor loading exceeding within factor loadings. More specifically, factors 1, 2 and 3 correspond to Learning-Related Scale, Related Knowledge, and Diversity, respectively. Criteria 3 was also met, with all within factor loadings above the cut off of .5. This analysis thus suggests good convergent and discriminant validity.

Table 3.4.6.4: Factor Loadings-Pattern Matrix

Indicator	Factor 1	Factor 2	Factor 3
LRS1	.93	.03	.01
LRS2	.97	-.05	.05
RK1	-.04	.61	.01
RK2	.12	.57	.01
RK3	-.18	.69	.00
RK4	.03	.73	-.01
DIV1	.11	.02	.53
DIV2	-.02	.07	.67
DIV3	-.04	-.06	.55

Confirmatory Factor Analysis Using Structural Equation Modeling

The next step in the analysis of convergent and discriminant validity was a confirmatory factor analysis using structural equation modeling with Lisrel [Bagozzi and Phillips 1982; Venkatraman and Ramanujam 1987]. The main advantages of Lisrel over the traditional methods described above are: 1) provision of explicit estimates of error variances for each indicator; 2) support for specific statistical tests of convergent and discriminant validity; and 3) the ability to impose and test just those constraints suggested by theory, rather than the ones imposed by the statistical procedure itself. As shown below, the Lisrel model analysis does indicate that there is considerable room for improvement in measuring these constructs; however, the contribution of this study lies not in developing the best possible construct measures, but rather, in testing relationships between constructs that are adequately measured.

Using the Lisrel program option in SPSS, the model in Figure 3.4.6.1 below was estimated. This is the best fit measurement model for these data. Each indicator was specified to load one factor and no others. The indicator correlations are provided in Table 3.4.6.5. The χ^2 for this model is χ^2 (df=29)=192 (P=.000). Covariation between Learning-Related Scale and Diversity, and between Learning-Related Scale and Related Knowledge were

Table 3.4.6.5: Indicator Correlations

	LRS1	LRS2	RK1	RK2	RK3	RK4	DIV1	DIV2	DIV3
1. LRS1	1								
2. LRS2	.92	1							
3. RK1	-.06	-.09	1						
4. RK2	.10	.06	.33	1					
5. RK3	-.18	-.24	.45	.37	1				
6. RK4	-.01	-.05	.44	.43	.50	1			
7. DIV1	.33	.37	.07	.09	-.05	.05	1		
8. DIV2	.28	.31	.03	.13	.08	.10	.39	1	
9. DIV3	.21	.23	.02	.00	-.04	-.00	.30	.35	1

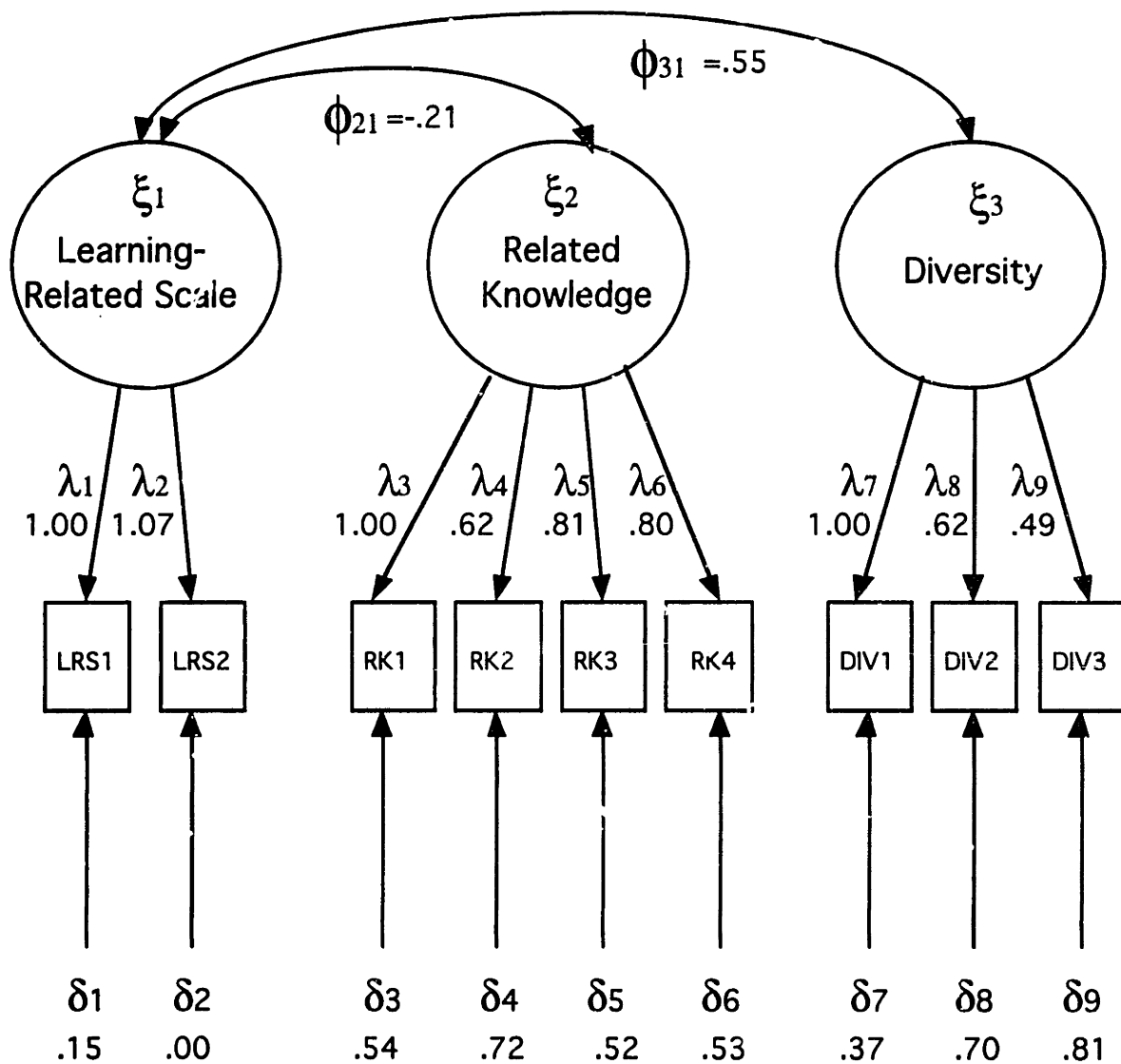


Figure 3.4.6.1: Lisrel Model Estimates

permitted, while covariation between Diversity and Related Knowledge were fixed to zero. The error variance for LRS2 was fixed to zero because in a preliminary run of the model, it was estimated to be slightly negative (-.05), a theoretically impossible result.

The *a priori* rationale for the positive link between Learning-Related Scale and Diversity, confirmed in the best-fit model, is that organizations with a larger scale or new development activities are more likely to have a longer history of substantial development activities, thus leading to a natural accumulation of diversity in applications-related technologies. Covariation between Learning-Related Scale and Related Knowledge was not originally expected since no compelling rationale links those variables. However, freeing up this link did lead to a significant improvement of the chi-square statistic $\chi^2(df=1) = 13, p = .000$. This weak relationship probably results from the fact that very small IT organizations, with a tendency towards lower Learning-Related Scale are more likely to have a large percentage-wise commitments to applications development on PCs and workstations—one of the four indicators of the Related Knowledge construct. There was no *a priori* compelling rationale for a strong relationship between Diversity and Related Knowledge, and the absence of this link was borne out, in that freeing this link did not lead to a significant further improvement in the Chi-square statistic.

An examination of the t-statistics for Lambda loadings supports convergent validity in that all are highly significant. However, an examination of error variances reveals some areas for concern. Specifically, three indicators DIV2, DIV3 and RK2—have error variances above .70, which violates the rule of thumb that about half the variance of indicators should be explained by their associated factors.

To examine discriminant validity, an alternative “constrained” model was estimated with the three correlations among the factors fixed to unity. (This is equivalent to a model where all indicators are hypothesized to load on a

single factor.) The model statistic for the constrained model is $\chi^2(df=30) = 1196$. The statistic for χ^2 differences between the original and constrained model is $\chi^2(df=1) = 1004, p=.000$. This satisfies the criterion for discriminant validity suggested by Jöreskog, namely, that the original model is a significantly better fit to the data than the constrained model [1971].

Nomological Validity

As a cursory check of nomological validity, zero-order correlations were computed for the three independent constructs and the dependent construct, Assimilation Stage (see Table 3.4.6.6). (The factor scores for the independent constructs were computed using the Lisrel Lambda loadings.)

Table 3.4.6.6: Correlation Matrix for Independent and Dependent Constructs

	Assimilation Stage	Learning-Related Scale	Related Knowledge	Diversity
Assimilation Stage	1			
Learning-Related Scale	.38**	1		
Related Knowledge	.29**	-.09*	1	
Diversity	.33**	.40**	.07	1

** $p \leq .01$ * $p \leq .05$

Expected relationships (all those involving Stage, and the Learning-Related Scale to Diversity link) are all confirmed, with $r = .29$ or higher. In addition, only one relationship that was not expected, i.e., between Learning-Related Scale and Related Knowledge, did have a significant correlation. As mentioned previously, this is probably due to the higher propensity of small IT groups to have a high percentage-wise commitment to PC/Workstation-based development. The full analysis of the theoretical model using multivariate regression techniques provides another, more rigorous test of nomological validity.

Summary

In summary, the quantitative components of construct validity appear to be quite adequate for the purpose of this study, which is to test relationships between novel constructs that are adequately measured.

3.5 RESULTS

This section presents the results of several statistical analyses, which collectively test the proposed model predicting OOPL Assimilation Stage. The three hypotheses to be tested are:

H1: Learning-Related Scale is positively related to OOPL Assimilation Stage.

H2: Related Knowledge is positively related to Assimilation Stage.

H3: Diversity is positively related to Assimilation Stage.

Descriptive data on the nature and extent of OOPL Assimilation are provided in Section 3.5.1, followed by a set of multi-variate regressions testing the causal model in Section 3.5.2. Although the proposed model is tested using data on assimilation of OOPLs, it is intended to generalize broadly to SPIs.

3.5.1 Descriptive Data on OOPL Assimilation

OOPLs are one of the more recent SPIs to become commercially available, and this fact is reflected in the break down of responding organizations provided in Figure 3.5.1.1. While nearly half of responding organizations (44%) have acquired at least one OOPL, only about 11% have committed to using it for production applications, and only 1% have reached the General Deployment Stage.

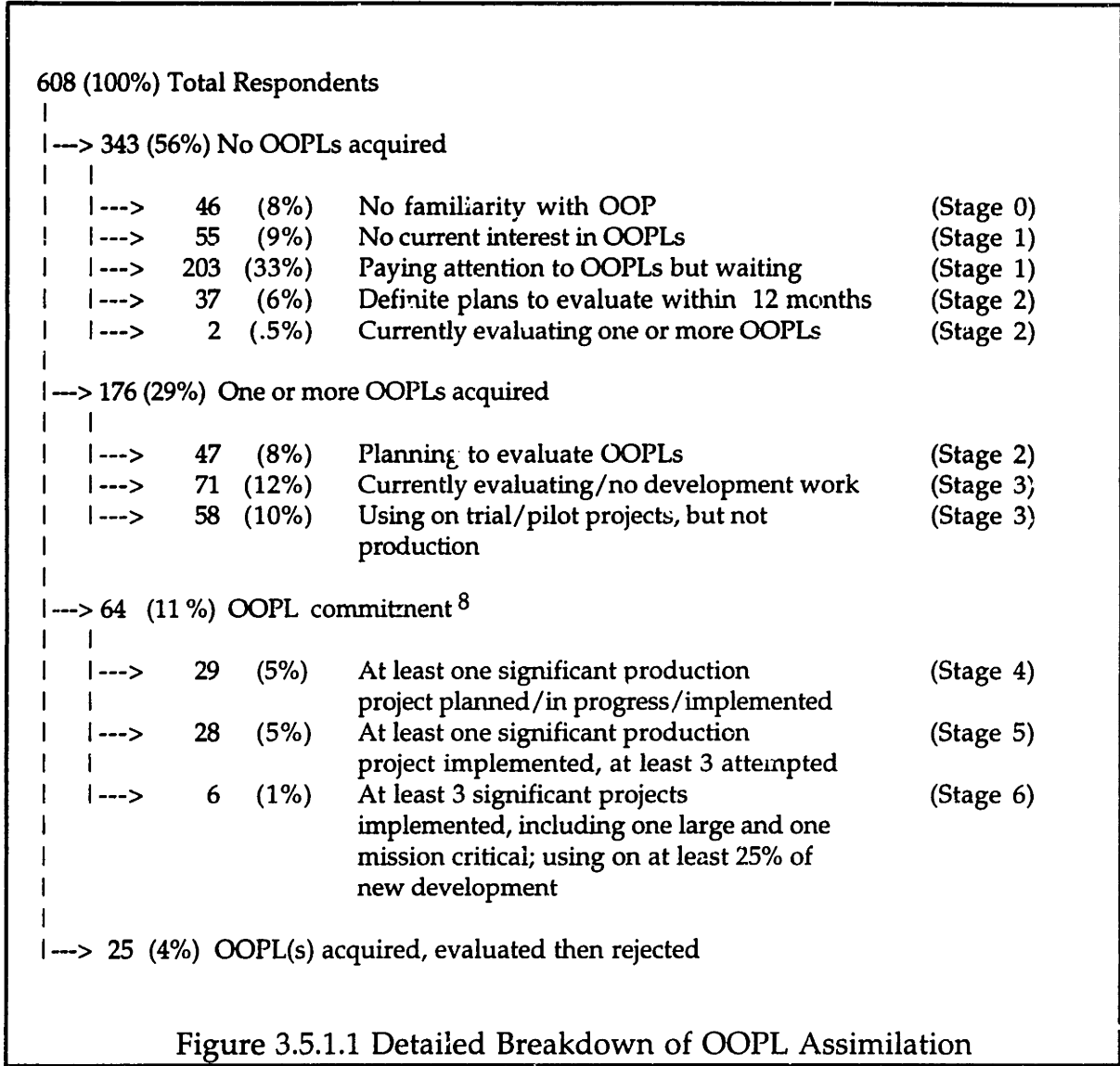


Figure 3.5.1.1 Detailed Breakdown of OOPL Assimilation

This figure illustrates the value of gathering detailed data in support of a rich measure like Assimilation Stage, as opposed to the traditional use of time of adoption or dichotomous adoption. Of the 123 organizations that report using an OOPL, almost one half (N=58) are only using it for trial projects, with the others (N=64) using it for production. Many organizations that have acquired an OOPL, and would therefore be classified as adopters in some studies, have actually yet to begin an evaluation (N=47), or have rejected the technology (N=25). The importance of being able to segregate rejectors will be

⁸This value sums to 64 rather than 63 due to rounding of fractional values to the nearest integer.

shown in the Section 3.5.2, where it is demonstrated that rejectors, rather than resembling other acquirers (or committed users) actually bear a closer resemblance to organizations in the Aware and Interested categories. Table 3.5.1.1 below provides a summary breakdown by Assimilation Stage.

Table 3.5.1.1 OOPL Assimilation Stage

Category	Count	Percent
0. Not aware	46	7.6
1. Aware	258	42.5
2. Interest	86	14.1
3. Evaluation/Trial	129	21.2
4. Commitment	29	4.8
5. Limited Deployment	28	4.6
6. General Deployment	6	1.0
TOTAL	583	100

This table confirms the particular value of Assimilation Stage as an innovativeness measure for SPIs that have emerged more recently. The inclusion of the first three categories (Not Aware through Interest) captures significant variance in innovativeness that would be excluded from consideration in any of the other measures (Time of Adoption, Dichotomous Adoption, Aggregated Adoption, Diffusion, Infusion) to be described in Chapter 4.

Reasons for Commitment, Rejection, Delay

Three groups of respondents—committed users (N=64), rejectors (N=24) and those that were aware of OOPLs but not interested (N=258)—were queried regarding the reasons for these behaviors.

Committed users were asked to select, from a randomly ordered list, all factors that were significant in triggering their organizations to approve the use of OOPLs on production projects (see Table 3.5.1.2). Instrumental reasons, including such traditional objectives as expectations of improved develop-

ment cycle times and reduced cost, were the most prominent, thus providing indirect support for the instrumentally oriented theoretical model. Less instrumental reasons, such as external or internal "encouragement," were the least prominent.

Table 3.5.1.2: Reasons for Commitment (N=64)

Reason	Percent Selecting
To help move to client-server applications development environment	66%
Expectations of reduced development cycle times	65%
Expectations of increased developer productivity/lower costs	60%
To help integrate production applications and desktops	59%
To support new applications that can't be done with current technologies	54%
To help achieve high levels of software reuse	47%
Expectations of more reliable applications	39%
Need OOPLs to use other purchased software, e.g., Windows	31%
An external organization strongly encouraged adoption	18%
Senior management strongly encouraged adoption	17%
Competitors already adopting OOPLs	11%
Merged with another group that was already using an OOPL	5%
Other	7%

Rejectors were also asked to select, from a randomly ordered list, the factors contributing to their decision to not use an OOPL that had been acquired and evaluated (see Table 3.5.1.3). These results are also consistent with the research model in the following sense. Rejectors are organizations that initiate assimilation and then come to find there is not, after all, a good fit between the technology and the organization, or that the technology simply does not perform up to expectations. The number one reason for rejection, "too difficult to integrate with existing tools and processes" can be viewed two ways, either as a sign of a problem with the technology, or as a sign of a lack of capability or commitment on the part of adopters. Organizations that are more skilled, or able to commit the resources necessary to overcome the technical limitations of a less mature technology, are less likely to conclude that integration is too "difficult". The prominence of reasons such as "prefer to

wait until standards have emerged" and "technology too immature" and "too early to invest" (56% selected at least one of these reasons) imply that these organizations may have found, after initiating assimilation, that knowledge barriers were in effect still too high, or in the view put forward by Attewell, they had not sufficiently delayed adoption.

Table 3.5.1.3: Reasons for Rejection (N=25)

Reason	Percent Selecting
Too difficult to integrate with existing tools and processes	61%
No pressing need for OOP in our environment	56%
Prefer to wait until standards have emerged	31%
Technology too immature	29%
Lack of management support	25%
Implementation cost could not be justified	23%
Too early to invest	20%
Other new technologies are more promising than OOPLs	15%
Claimed benefits could not be verified	12%
Other	12%

Organizations deferring evaluation were asked to select reasons accounting for their delay from a list similar to the one presented to rejectors (see Table 3.5.1.4). The most obvious reason—no perceived need for the technology—was by far the most frequent response, although here again, other reasons, such as "prefer to wait until standards have emerged" and "too early to invest" and "technology too immature" were prominent choices, with 53% selecting one or more of these reasons. This, too, is consistent with Attewell's thesis that organizations delay adoption until knowledge barriers are sufficiently lowered.

Table 3.5.1.4: Reasons for Deferral (N=258)

Reason	Percent Selecting
We have no pressing need for OOPLs in our environment	58%
Prefer to wait until standards have emerged	37%
Too early to invest	34%
OOPLs are too incompatible with our legacy systems	34%
Technology too immature	30%
Not convinced claimed benefits of OOPLs are genuine	27%
Lack of management support for OOPL evaluation	23%
OOPLs are too difficult to evaluate and trial	12%
Other new technologies are more promising than OOPLs	4%
Other	6%

OOPL Language Breakdown

C++ is by far the dominant language acquired. Of locations that have acquired any OOPL (N=2645) 94% have acquired C++, 88% consider it their primary OOPL, and 67% have acquired nothing but C++ (see Table 3.5.1.5 below). With Smalltalk, by comparison, 15% have acquired it, 5% consider it their primary OO language, and only 1% have acquired nothing but Smalltalk. The only surprise here is the relatively high proportion of acquirers of OO Pascal hybrids. Further investigation indicates that this figure appears trustworthy.⁹

⁹Of the 44 OO Pascal acquirers, 40% report at least some custom development for Macintoshes (and hence are candidate Object Pascal adopters) versus only 10% for non OO Pascal acquirers. 91% of OO Pascal acquirers report DOS/Windows development (and hence are candidate Quick or Turbo Pascal adopters) versus 73% for non-OO Pascal acquirers (see Q_04a.5 vs. Q_06.12; Q_04a.5 vs. Q_06.3). This suggests that about 1/3 of OO Pascal hybrid acquirers are Object Pascal, and the other two thirds are Quick Pascal and Turbo Pascal 5x.

Table 3.5.1.5: OOP Language Breakdown Among Acquirers (N=264)

Language	Acquired		Primary OOPL		Sole OOPL acquired	
	N	Percent	N	Percent	N	Percent
C++	249	94%	233	88%	177	68%
Smalltalk	40	15%	14	5%	3	1%
OO Pascal	44	17%	14	5%	8	3%
Objective C	19	7%	4	2%	3	1%
Eiffel	4	2%	0	0%	0	0%
CLOS	4	2%	0	0%	0	0%

3.5.2 Antecedents of OOPL Assimilation

This section describes the procedure used to estimate and assess the proposed theoretical model, stated formally as follows:

$$\text{Assimilation Stage} = \beta_0 + \beta_1 * \text{Learning-Related Scale} + \beta_2 * \text{Related Knowledge} + \beta_3 * \text{Diversity} + \beta_4 * \text{IT Size} + \beta_5 * \text{Host Size} + \beta_6 * \text{Specialization} + \beta_7 * \text{Education} + \beta_8 * \text{Environmental Complexity} + \beta_9 * \text{Government} + \varepsilon.$$

The first three variables correspond to the independent constructs; the last six are controls.

Descriptive data on the indicators used to calculate the independent and control variables are provided in Table 3.5.2.1. For the two categorical indicators (O and P) the median value is provided. For all other indicators, mean and standard deviation are provided. Values for all constructs are calculated as the mean of the standardized value of all associated indicators, with the exception of the three independent constructs, where the standardized indicators are multiplied by their associated Lisrel lambda weights and then summed.

Table 3.5.2.1: Descriptive Data on Indicators

Construct	Ind	Indicator descriptions	Mean	S.D.
N/A	A	Number of application developers at site (Used in calculation of Learning-Related Scale (LRS))	32.8	125.5
N/A	B	Percentage of applications-related effort attributable to new systems (Used in calculation of LRS)	25.1	14.3
N/A	C	Percentage of applications-related effort attributable to new significant enhancements (Used in calculation LRS indicators)	32.3	20.4
N/A	D	The total number of employees in the host organization (Used in calculation of Host Size)	4267	12671
Learning-Related Scale	E	$\text{LOG}(A*B)$	2.39	.77
	F	$\text{LOG}(A*(B+C))$	2.71	.70
Related Knowledge	G	Percentage of development staff with experience programming in C	20.6	26.2
	H	Percentage of development staff with experience developing PC or workstation applications	16.9	25.9
	I	Percentage of development staff with experience developing client-server applications	38.7	33.2
	J	Percentage of development staff with experience developing graphical user interfaces	16.5	24.6
Diversity	K	The number of different programming languages used by $\geq 5\%$ of the development staff in 1993	2.27	1.20
	L	The number of different runtime platforms accounting for $\geq 5\%$ of new development over last 3 years	2.77	1.26
	M	The number of different application architectures accounting for $\geq 5\%$ of new development over last 3 years	3.63	1.09
IT Size	O	Mean of categorical variable capturing level of external IT spending within respondents span of influence	\$500 to 999k	-
	P	Mean of categorical variable capturing number of IT employees within respondent's span of influence	10 to 24	-
Host Size	N	$\text{LOG}(D)$	3.17	.62
Specialization	Q	Sum of the number of six specialties for which the site has at least one full time staff member	1.88	1.78
Education	R	The percentage of IT staff at the site holding bachelor's degrees or higher	65.6	31.8
	S	The percentage of IT staff at the site holding master's degrees or higher	10.5	16.0
Environmental Complexity	T	Mean importance of seven typical objectives for systems development (7 point scale)	5.51	1.05
Government	U	Binary variable identifying sites where government is the primary industrial sector	.14	.35

The zero-order correlations among all the variables are provided in Table 3.5.2.2. An examination of the correlations involving Assimilation Stage shows that all have the expected signs. Government was expected to be negatively correlated with Assimilation Stage because of the prior finding that governmental agencies are less innovative than private sector organizations regarding information technology [Bretschneider and Wittmer 1993]. With the exception of Environmental Complexity, all of the correlations between predictor variables and Assimilation Stage are significant at $p \leq .05$. Of the significant correlations among predictor variables, all also have the expected signs (positive), with the exception of the small, negative correlation between Learning-Related Scale and Related Knowledge ($r = -.09$). (As explained in Section 3.4.6 above, this small correlation is likely due to the propensity of smaller scale operations to do more development work on PCs and workstations.)

Table 3.5.2.2: Correlations

	Assim Stage	1.	2.	3.	4.	5.	6.	7.	8.
1. Learn-Rel Scale	.38**								
2. Rel Knowledge	.29**	-.09*							
3. Diversity	.33**	.40**	.07						
4. IT Size	.38**	.62**	-.06	.34**					
5. Host Size	.15**	.39**	-.07	.19**	.41**				
6. Specialization	.28**	.37**	.08	.33**	.34**	.23**			
7. Education	.19**	.20*	.23**	.16**	.19**	.11	.11		
8. Env Complexity	.08	.02	.17**	.00	.00	-.01	.10*	.01	
9. Government	-.13	-.02	-.13**	-.03	-.08*	.01	-.07	-.09	-.03

** $p \leq .01$ * $p \leq .05$

Table 3.5.2.3 provides the results for six estimated models. The cells contain the standardized beta coefficients, with the associated t-statistics in parentheses. Model 1 is the predicted, or "base" model. Model 2 is a controls-only model that provides benchmark for assessing the additional impact of independent variables. Model 3 includes only the three theoretical variables

as predictors. Models 4 and 5 provide the basis for evaluating two threats to validity. Model 6 supports an analysis of the hypothesis of differently-directioned effects.

Table 3.5.2.3: Estimated Models

	Model 1 Assim Stage (N=583)	Model 2 Assim Stage (N=583)	Model 3 Assim Stage (N=583)	Model 4 Stage T2 (N=388)	Model 5 Stage- Early (N=519)	Model 6 Stage- Late (N=192)
1. Learning- Related Scale	.22**** (4.6)		.34**** (8.7)	.16** (2.7)	.12* (2.2)	.24** (2.7)
2. Related Knowledge	.29**** (7.9)		.18**** (4.5)	.30**** (6.9)	.19**** (4.5)	.23*** (3.2)
3. Diversity	.14*** (3.6)		.32**** (8.7)	.15** (2.9)	.21**** (4.7)	-.11 (-1.7)
4. IT Size	.20**** (4.4)	.31**** (7.2)		.23**** (4.0)	.22**** (4.4)	.02 (.3)
5. Host Size	-.05 (-1.2)	-.03 (-.7)		-.03 (-.6)	.00 (.1)	-.16* (-2.3)
6. Specialization	.06 (1.5)	.15**** (3.8)		.05 (1.1)	.02 (.5)	.05 (.7)
7. Education	.01 (.2)	.11* (2.8)		.04 (.9)	.01 (.3)	.01 (.1)
8. Env Complexity	.02 (.6)	.06 (1.6)		.03 (.6)	-.01 (-.2)	.04 (.6)
9. Government	-.05 (-1.6)	.08* (-2.1)		-.02 (-.4)	-.06 (-1.4)	-.02 (-.3)
10. Time						.38**** (5.7)
Adjusted R ²	.30	.18	.27	.31	.22	.27
F	28.8****	22.2****	74.3****	20.7****	17.4****	8.2****

****p ≤ .0001 ***p ≤ .001 **p ≤ .01 *p ≤ .05

An examination of the results for the base mode (Model 1) reveals that all three hypotheses are strongly supported. The beta coefficients for Learning-Related Scale, Related Knowledge, and Diversity are all highly significant,

with t-statistics of 4.6, 7.9 and 3.6, respectively. IT Size is the only other significant predictor ($t=4.4$) in this model.¹⁰

It is worth noting that the decision to include a control variable, IT Size, that strongly covaries with a theoretical variable, Learning-Related Scale, represents a conservative approach to testing the theoretical model, because this can serve to hide a genuine effect of the theoretical variable. This, in fact, did not happen: both variables were strongly significant when jointly included. Another potential effect of including both variables is to increase instability of the parameter estimates across the different models. However, this potential disadvantage is believed to be outweighed by the advantages of the approach taken, which is to strongly rule out the rival hypothesis that the Learning-Related Scale effect is solely a spurious result due to its covariation with IT Size.

The overall model is highly significant, with an F-statistic of $F=28.8$ ($p \leq .0001$), and explains 30% of the variance. A comparison of Models 1 and 2 shows that the full model explains an incremental variance of 12%, which constitutes a significant increase in variance explained (F ratio= 34.63 , $P \leq .0001$). Including the control variables on top of the independent variables, by contrast, only explains an additional 3% of the variance, as shown by a comparison of Models 1 and 3.

Standard regression diagnostics reveal no violations of statistical conclusion validity. The histogram and normal probability plot for standardized residuals both show that residuals are close to normally distributed. A scatter plot of standardized residuals versus standardized predicted values shows no significant patterns, thus confirming the absence of heteroscedasticity. The largest variance inflation factor was $VIF=2.6$, well under the rule-of-thumb of

¹⁰A multi-level logistic regression was also run for this model, because, strictly speaking, this is a more valid statistical procedure for ordinal independent variables like Assimilation Stage. In point of fact, the same predictor variable were significant, with nearly identical levels of significance.

VIF=10 used to identify potentially problematic colinearity [Cryer and Miller 1991].

Analysis of Threats to Validity

The plausibility of three major threats to the validity of the base model are examined in this section. The first threat is related to common method bias. An inspection of Model 4 shows that the independent variables remain highly significant when Assimilation Stage as measured at time 2 (using the paper-based questionnaire) is substituted into the base model, thus providing no support for the hypothesis that the results of the base model are due to the fact that the data on both predictor and outcome variables were collected at the same time with the same instrument.

The second major threat to the validity of the base model is potential reverse causation between Related Knowledge and Assimilation Stage. This would be possible if it turned out that organizations in the study predominantly accumulated experience in the indicators of the Related Knowledge construct (the C language, client-server, etc.), *after* they began using an OOPL, rather than the reverse. To assess this threat, a model was estimated (Model 5) that only included organizations earlier in the OOPL assimilation process, where Assimilation Stage ranged from 0 (Not Aware) to 3 (Evaluation/Trial). Since only organizations that have no experience developing with OOPLs are included in this model, it is not possible that the level of Related Knowledge in these organizations is due to such experience, yet the coefficient for Related Knowledge is still highly significant ($t=4.5$; $p \leq .0001$).

The third potential threat to the validity of these results also concerns Related Knowledge. It can be argued that some of the indicators of this construct, particularly client-server and GUI experience, might themselves be indicators of software process innovativeness. If true, this opens up the possibility that the relationship between Related Knowledge and Assimilation Stage results from unmeasured common causes of software process innovativeness. An

examination of the correlations in Table 3.5.2.2 renders this implausible, however, because Related Knowledge is only weakly correlated with other variables that were included in the model, and is especially weakly correlated with the strongest predictors of Assimilation Stage (Learning-Related Scale, Diversity, and IT Size).

Differently Directioned Effects

Another issue worth investigating is the possibility that the use of Assimilation Stage as the outcome variable masks the fact that some predictors have differently directioned effects depending on where organizations are in the assimilation process. Specifically, it is possible that the influence of the predictors turns negative in later stages of the assimilation process. The only variable for which a compelling rationale for differently directioned effects could be developed was Diversity: it does seem possible that a more diverse environment might make it more difficult to widely deploy a technology because of the need to address a greater array of integration-related issues.

To evaluate the plausibility of the existence of differently-directioned effects, Model 6 was constructed that only included organizations where Assimilation Stage ranged from 3 (Evaluation/Trial) to 6 (General Deployment). A variable capturing the number of years since acquisition of an OOPL (Time) was also included to control for the effects of more rapid assimilation up to the time the organization reached the Assimilation Stage 3. Interestingly, the coefficient for Diversity does turn negative in this model, although it is not significant ($p=.10$). The signs for all other coefficients are the same for this model as for Model 1.

To further explore the relationship between predictor variables and Assimilation Stage, the means for the five variables with the strongest direct association with stage—Learning-Related Scale, Diversity, Related Knowledge, IT Size, and Specialization—were computed for each level of

Assimilation Stage. Table 3.5.2.4 shows these mean values, together the ranking of the value from 1 (the category with the lowest mean value) to 8 (the category with the highest).¹¹ The mean values and ranking of Rejectors were also included, to support comparisons involving this group. While there are nearly uniformly increasing mean values for four of the strongest predictors (Learning-Related Scale, Related Knowledge, IT Size and Specialization), the mean values are non-monotonic for Diversity in the last four stages. This is not a conclusive result by any means—just six cases fall in the General Deployment category, yet they appear to be particularly influential in producing the non-significant Diversity result. It does, however, suggest that the hypothesis of a negative or mixed effect for Diversity in later stages cannot be completely ruled out.

Table 3.5.2.4: Mean Values of Predictors by Stage

	Learning-Related Scale	Related Knowledge	Diversity	IT Size	Specialization
0. Not aware	-.62 (1)	-.92 (1)	-.94 (1)	-.52 (1)	-.29 (2)
1. Aware	-.33 (3)	-.55 (2)	-.54 (2)	-.31 (3)	-.32 (1)
2. Interest	.10 (4)	-.06 (3)	.02 (4)	.08 (4)	.14 (4)
3. Evaluation/Trial	.28 (5)	.20 (4)	.70 (6)	.35 (6)	.17 (5)
4. Commitment	.51 (6)	.68 (6)	.13 (5)	.22 (5)	.45 (6)
5. Limited Deployment	.77 (8)	1.36 (7)	1.2 (7)	.64 (7)	.52 (8)
6. General Deployment	.74 (7)	5.58 (8)	-.39 (3)	1.0 (8)	.47 (7)
Rejectors	-.37 (2)	.58 (5)	.02 (4)	-.39 (2)	-.07 (3)

The results in Table 3.5.2.4 related to Rejectors bear a closer examination. The reasonable assumption that Rejectors, having already evaluated an OOPL, should have predictor values at least as high as other Evaluators (Assimilation Stage 3), is not supported. In fact, the average of the predictor

¹¹Since these constructs are linear combinations of standardized variables, the overall mean is zero. The means for the cases falling in the General Deployment stage (Stage=6) should be interpreted with extreme caution, because the number of cases in this cluster—six—is far too few to smooth out random variation.

values is much lower for Rejectors than for Evaluators for all variables except Related Knowledge. This result is fully consistent with the instrumental nature of the proposed model of innovativeness. Specifically, organizations with comparatively low values of predictor variables, are viewed to be, in effect, poor candidates for assimilation of OOPLs. Should such organizations "erroneously" include themselves among the early initiators of OOPL assimilation, the proposed model would predict they should be less likely to sustain innovation, thus leading to higher rates of rejection and discontinuance. This confirms the importance of identifying Rejectors and segregating them when analyzing the predictors of adopter innovativeness.

3.6 DISCUSSION

3.6.1 Summary of Results

This research has strongly confirmed the expected role of organizational learning-related factors in the innovation process. Specifically, it was shown that Learning-Related Scale, Related Knowledge, and Diversity are all positively related to OOPL Assimilation Stage. The full model, with an $R^2=.30$, explained significantly more variance than the controls-only mode ($R^2=.18$). In fact, a model with just the three learning-related factors explained 27% of the variance, nearly as much as the full model.

It was also found that while four of six control variables—IT Size, Specialization, Education and Government—were significant predictors in the controls-only model, only one of these, IT Size, significantly predicts Assimilation Stage in the full model. This confirms the nomological validity of the control variables, while demonstrating the potential for inappropriate inferences in the absence of the hypothesized organizational learning-related factors.

Other notable results include the finding that the profile of Rejectors more nearly resembles the profiles of those organizations in early stages

(Awareness and Interest) rather than those in later stages (Evaluation, Commitment and Deployment), which is consistent with the notion that those organizations that initiate assimilation "too soon"—as implied by their comparatively low levels of predictor variables—are less likely to sustain innovation. It was also found that the influence of Diversity becomes non-significant when analysis is limited to just those organizations in the later stages of assimilation, raising the possibility that, while Diversity facilitates early assimilation activities, its role during deployment may be mixed. Finally, the use of a disk-based mode of survey administration appears to have been a major plus, as evidenced by the high response rate, the small number of unusable responses, the absence of complaints about the medium by 90% of non-respondents contacted by telephone, and the positive attitudes of respondents towards the prospect of completing future disk-based surveys.

Numerous procedures were performed to ensure the trustworthiness of the above results (see Table 3.6.1.1 below). The research model was developed based on a thorough review of the innovation literature and more recent work in organizational learning. The survey instrument was developed over a four month period, using the procedure recommended by Dillman, culminating with five in-person pretests. The sampling frame was extracted from a high-quality list shown to be consistent with the desired theoretical universe, and a probability sample was extracted from this frame, thus ruling out the possibility of a selection threat. A large sample size (N=608) and a high response rate (45%) was achieved through careful attention to survey administration and judicious use of incentives, thus ensuring adequate statistical power. Responding informants were shown to overwhelmingly fit the target criteria of being middle-level IT managers well informed about the applications development activities and technologies at the site. Sophisticated statistical analyses of the potential for response bias, for inadequate construct validity, and for method bias, were performed and rendered these threats implausible. An analysis of standard regression diagnostics revealed no violations of the assumptions of OLS regression.

And finally, analyses were performed that rendered implausible two prominent rival hypotheses: that Related Knowledge is a cause of, not a result of, OOPL assimilation, and that the observed influence Related Knowledge,

Table 3.6.1.1 Salient Threats to Validity and How Addressed

Category of threat	Specific concerns	How addressed in this study
Statistical conclusion validity	Violated assumptions of statistical tests	Standard regression diagnostics
	Inadequate statistical power	Large N (over 600)
Internal validity	Selection bias	Random selection of sites
	Common causes, reverse causation	Inclusion of control variables; analyses of specific rival hypotheses
Construct validity	Theoretical meaningfulness of constructs	Built on existing theory
	Observational meaningfulness	Careful attention to question construction
	Convergent validity/reliability	Analysis of within-construct raw correlations; Cronbach's alpha; analysis of within construct factor loadings produced by exploratory factor analysis; analysis of lambda loadings produced by Lisrel confirmatory factor analysis
	Discriminant validity	Analysis of cross construct factor loadings produced by exploratory factor analysis; Chi-square test of nested Lisrel models
	Nomological validity	Analysis of correlations between constructs; multivariate regression
	Method bias	Follow up survey of respondents
External validity	Inability to generalize to theoretical universe	Use of high quality list of potential informants; careful construction of sampling frame; inclusion of wide variety of organizations spanning many sizes and industries; analysis of response bias
Key informant analysis	Motivational barrier	Participation voluntary; confidentiality assured; incentives provided
	Perceptual and cognitive limitations	In person observation of pre-test informants; questions kept as simple and specific as possible; use of computational aids provided by disk-based survey
	Lack of information	Employed IT managers as key informants; collected data about informants to support post-hoc assessment of qualifications

which itself might be considered an indicator of SPI innovativeness, is a spurious result of unmeasured determinants of innovativeness.

3.6.2 Potential Limitations

The main potential limitations of this research arise from the use of a cross-sectional survey design with a single key informant. This approach raises four potential issues. First, it raises concerns about the accuracy of the measured values for predictor variables. Ideally, the values for these variables should be captured at multiple times before and during the innovation process they are hypothesized to influence, but in point of fact, the variables were captured for the period immediately preceding the administration of the survey instrument. However, since the values of the predictor variables in this study tend to change only slowly over time, it is likely that the values at the time of survey administration are highly correlated with values over the preceding few years—the time in which the vast majority of OOPL-related innovation activities occurred. Nearly 90% of those acquiring an OOPL did so in the 3 years immediately preceding administration of the survey.

Second, there is a concern about time-ordering of effects. While it appears implausible that levels of the independent variables could have been significantly affected by OOPL assimilation, rather than the reverse—recall that only 10% of the informants have begun the OOPL deployment process in earnest—this possibility can not be completely ruled out.

The third concern is that for some sites, some measures may have been inaccurate because the key informant may have been either unmotivated or not very knowledgeable about the domains covered in the survey. While it appears the vast majority of respondents were in fact well motivated and well informed, the single informant design precludes the opportunity to strongly confirm the reliability of measures by making comparisons across informants.

Fourth and finally, the use of a single informant increases the potential for a method bias. Even though the best way to rule out this threat is through the use of separate sources to measure independent versus dependent variables, a secondary analysis of outcome data gathered through a follow-up paper-based survey showed no support for the method bias hypothesis.

3.6.3 Generalizability of Results

This research was designed with the goal of three kinds of generalizability. First, it was intended that these results should generalize to the universe of internal application development sites at medium to large US-based enterprises. Since the responding sites did in fact span a wide range of industries and sizes, this kind of generalizability seems assured.

Second, it was intended that results, though tested based on the case of OOPLs, should generalize to other typical software process innovations. This kind of generalizability also seems assured, because OOPLs possess the characteristics associated with knowledge barriers (abstract/demanding scientific base, fragility, etc.) that bind SPIs into an interesting class of innovations and motivated the selection of independent variables in the first place. Also, application of the model to other SPIs and to Aggregated Adoption (see Chapter 5) leads to similar results for the two non-innovation specific independent constructs, Learning-Related Scale and Diversity. (Recall that the measure for Related Knowledge is specific to OOPLs).

Third, it was expected that while Assimilation Stage should be more robust than other general measures of organizational innovativeness, the results should nevertheless be broadly consistent for other general innovativeness measures, such as time of adoption, dichotomous adoption and aggregated adoption. Once again, this kind of generalizability also seems assured, based on the finding, also reported in Chapter 5, that models in which these other measures are substituted for Assimilation Stage produce similar patterns of significant relationships for the three theoretical variables.

3.6.4 Implications and Contributions

This research has important implications for theory, methods, and practice. With regard to theory, this research supports the reconceptualization of diffusion for complex organizational technologies initiated by Attewell. Specifically, this research maps the implications of Attewell's macro-level model to the organizational level, identifies three organizational learning-related factors that should be of particular importance for organizational assimilation of complex organizational technologies, develops measures for these factors tailored to the SPI context, and provides statistical confirmation of their expected influence on innovation. The strong influence of Related Knowledge—combined with the fact that this construct does not strongly covary with "generic" innovation predictors—demonstrates the value of examining particular technologies in detail with models that incorporate innovation-specific predictors.

The strong influence of Learning-Related Scale and Diversity—combined with the fact that these constructs *do* covary with other common innovativeness predictors—suggests that future research on SPI Assimilation should seriously consider incorporating these factors to control for possible confounds. Host organization size, for example, has a significant, positive zero-order correlation with Assimilation Stage, but is nevertheless insignificant in regressions that include these other variables. This result is consistent with Tornatzky *et al.*'s contention that the well-established empirical link between innovation and organizational size is more likely a result of other variables that covary with size—such as scale, professionalism, education, and specialization.

More generally, this research has demonstrated that it is possible to construct strongly predictive variance models, even when the focal innovation is a complex organizational technology. One of the long standing criticisms of diffusion research has been the general weakness and instability of findings in studies attempting to link organizational characteristics and innovation

[Downs and Mohr 1976; Rogers 1983, ch. 10; Fichman 1992]. This research has shown that strong results can be achieved, so long as researchers confine their focus to more specific innovations and contexts, and employ theory more closely tailored to those innovations and contexts.

With regard to methods, this research confirms the value of Assimilation Stage as an innovativeness measure, especially when the focal technology has not yet been widely adopted. If the traditional approach of using time of adoption or dichotomous adoption for the outcome measure had been used instead, no gradations in innovativeness would have been captured for as much as 90% of the responding population, depending on which definition was used for "adoption." (Furthermore, it would not have been possible to evaluate the hypothesis of differently-directioned effects.) One of the historical limitations of innovation research has been its typical backward-looking focus on technologies for which diffusion has already run its course. While that approach certainly has advantages, it does limit the opportunity to draw immediate, practical implications for vendors and potential users of the emerging technologies under study. In contrast, the research done here has identified factors affecting the assimilation of OOPLs early enough in the diffusion process for this information to be of direct benefit to vendors and potential users.

Another contribution to research methods is confirmation of the value of a the disk-based survey (DBS) mode. A survey as lengthy and complex as the one used in this research simply could not have been administered using traditional telephone or paper-based survey modes. Add to this the other advantages of DBS (minimization of item non-response and inadvertent coding errors; elimination of transcription costs, delay and errors) and the apparent positive effect on response rate, and one is left with a compelling case for the use the DBS mode, where it is feasible to do so.

This research also has practical implications—for technology vendors and mediating institutions, as well as end-users. For technology vendors and

mediating institutions, this research has identified the profile of organizations more likely to initiate and sustain SPI assimilation (and in particular, OOPLs), thus providing the basis for development of more targeted marketing and promotion. Targeted market is likely to be of particular value for complex organizational technologies such as SPIs, because, as Attewell has argued, broad brush "signalling" of the existence and potential benefits of such technologies is likely to be of lesser importance in promoting adoption.¹² In addition, when broadly based marketing does succeed in encouraging adoption by organizations that do not fit the assimilator profile, this could well be a pyrrhic victory, as such organizations should be less likely to sustain innovation, and may well become influential "negative" opinion leaders. Vendors and mediating institutions should rather be more focused on identifying appropriate adoption candidates, learning about the particular challenges these organizations face, and taking a more proactive role to promote successful assimilation among these sorts of organizations. Examples of such activities in the case of OOPLs abound. Many OOP language vendors now have technology consultation divisions that supply long term, on-site assistance with OOPL-based systems development. A new breed of technology consultants—referred to as "mentors"—are now available to organizations, that, rather than just doing development themselves, work side-by-side with end-users with an explicit charter to teach organizations how to be successful with object technology. Vendor-sponsored user groups are another common tactic employed to facilitate successful assimilation.

The main implication of this research for end-user organizations is that they would be well advised to view SPI assimilation as a multi-year process of organizational learning, and to consciously assess the extent to which they fit the profile of an early and sustained assimilator. Successful assimilation requires sizable investments in organizational learning and process change.

¹²Bolstering this argument is the finding in this study that only 8% of respondents were not yet aware of OOPLs.

Considering the expense and risk involved, those that do not fit the profile should seriously consider delaying adoption, or adopting a less complex variant of the technology. If such organizations do go forward with early adoption, it should with the understanding that risks are even higher, and therefore expectation management will be especially crucial.

Those organizations that do fit the assimilator profile—other things equal—should be more vigorous in assessing emerging technologies in general, and when a decision to adopt is made, should undertake assimilation strategies that exploit their inherent opportunity for more cost effective and successful assimilation. Learning-Related Scale opens the option of more expensive assimilation strategies that may well be essential to success, such as hiring of experts, development of a redesigned process (and the infrastructure to support it), and allowing extended periods of "practice" on non-production systems. However, managers must have the wisdom and will to employ these options if they are to be of benefit in increasing the chances of successful assimilation. Related Knowledge can also be extremely advantageous, but only if it is taken into account in such as activities as technology selection, project selection, and the assignment of personnel to projects. Diversity increases the likelihood of an organization possessing a "safe haven"—an area where an SPI is a particularly good fit, and the conditions for learning are more ideal—and also affords the opportunity to "bootstrap" learning from areas where the technology is a higher fit, and fewer demands are made on organizational competence, to lower-fit areas where successful application requires greater organizational competence. Once again, however, Diversity is only an advantage for organizations that do in fact exploit it wisely.

3.6.5 Future Work

The results of this research suggest several avenues for future work. One such avenue would be perform a detailed examination of the role of Diversity to determine whether this construct does in fact have a mixed or negative influence in later assimilation stages, and why.

A second avenue for future work would be to attempt to replicate the results achieved here in other settings. While the findings reported in Chapter 5 suggest these results are likely to be confirmed in rigorous studies of assimilation of other SPIs by end-user IT departments, it would be interesting to study other kinds of adopters, such as independent software vendors or consulting firms, to determine whether results hold for them as well. It would also be useful to apply this model to other complex organizational technologies beyond SPIs, to examine the extent to which the results can be generalized to the broader set of technologies that also impose a substantial knowledge burden.

A third avenue for future work would be case-based research focusing specifically on the assimilation strategies and tactics suggested by organizational learning-related factors. Interesting research questions include: are these tactics in fact being widely employed? if so, how effectively? if not, why not? And are these tactics more likely to be employed by organizations that fit the profile of an early/sustained assimilator? This kind of innovation *process* research would provide a natural and valuable complement to the *variance* model tested here.

Exhibit 1: Questions used to classify to Stages 0 and 1

E-02

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming
3|
4|
5| How would you describe your current level of familiarity with
6| object-oriented programming (OOP)?
7|
8| Please highlight the one item that best applies and press ENTER.
9|
10|
11|    Not at all familiar with OOP
12|
13|    Have heard of OOP, but not familiar with concepts
14|
15|    Somewhat familiar with OOP concepts
16|
17|    Very familiar with OOP concepts
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

E-03

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming
3|
4|
5| Has there ever been any evaluation or use of an object-oriented
6| programming language at your location?
7|
8|
9| Please highlight the one item that best applies and press ENTER.
10|
11|
12|    Yes, some activity involving OOP
13|
14|    No OOP activity to my knowledge
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

Exhibit 2: Questions used to classify to Stages 2 and 3

E-04

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|   Which of the OOPLs below have been acquired at your location for
5|   evaluation, trial development projects, or production development
6|   projects?
7|
8|   Using the arrow keys, move the highlight bar to each item
9|   you wish to select.  Then press ENTER.  Press F1 for help.
10|
11|           C++
12|           Smalltalk
13|           Objective C
14|           Eiffel
15|           Object Pascal, Quick Pascal, or Turbo Pascal 5x
16|           CLOS
17|           NONE/NO MORE/NEXT QUESTION
~
23|   To deselect an item, use arrow keys to highlight and press ENTER.
```

F-01 (Asked of those reporting no acquisition of OOPLs)

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|   What is the current status of general purpose OOPLs at your location?
5|
6|   Please highlight the one item that best applies and press ENTER.
7|
8|
9|   Not interested in OOP/no current plans to evaluate
10|
11|   Paying attention to OOP, but with no plans to evaluate right now
12|
13|   Have definite plans to evaluate an OOPL within 12 months
14|
15|   Currently evaluating one or more OOPLs
16|
17|
18|
19|
20|   General purpose OOPLs: C++, Smalltalk, Objective C, Eiffel
21|   Object/Quick/Turbo Pascal, CLOS
```

E-09 (Asked of those reporting acquisition of one or more OOPLs)

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|   What describes the current status of the general purpose OOPLs at
5|   your location?
6|
7|   Please highlight the one item that best applies and press ENTER.
8|
9|
10|   Previously evaluated and rejected OOPLs
11|
12|   Planning to evaluate OOPLs
13|
14|   Currently evaluating an OOPL, but not doing any development work
15|
16|   Using OOPL on trial/pilot projects, but not significant production projects
17|
18|
19|
20|   General purpose OOPLs: C++, Smalltalk, Objective C, Eiffel
21|   Object/Quick/Turbo Pascal, CLOS
```

Exhibit 3: Questions used to classify to Stages 4, 5 and 6

E-08

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Object-oriented programming languages
3|
4| Has this location ever approved the development of a significant
5| production application using a general purpose OOPL as the primary
6| programming language?
7|
8| Press a number to record your answer.
9|
10| 1 Yes, at least one significant OOPL production application
11| has been approved
12|
13| 2 No significant OOPL production applications have been approved
14| at this location
15|
16| Significant means projects requiring at least one person-month of
17| development effort.
18|
19| Production application means applications to be delivered to users
20| and operated and maintained on an ongoing basis. (We do not include
21| experimental, evaluation and pilot applications.)
22|
23| General purpose OOPL = C++, Smalltalk, Objective C, Eiffel,
24| Object/Quick/Turbo Pascal, or CLOS.
-----1-----2-----3-----4-----5-----6-----7-----8

E-10

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Object-oriented programming languages
3|
4| How many significant production applications development projects using
5| a general purpose OOPL as the primary language have ever been approved?
6|
7| Please type the number of projects (from 0 to 999) in each category
8| provided and press ENTER. Categories are mutually exclusive.
9|
10| Projects approved for development but not yet initiated
11| Projects in progress
12| Projects implemented
13| Projects initiated but later canceled/not implemented
14| Don't know or other
15| All correct? (Y/N)
16| By significant we mean projects requiring at least one
17| person-month of development effort
18|
19| By production applications we mean applications intended to be
20| delivered to users and operated and maintained on an ongoing basis.
21|
22| By general purpose OOPL we mean C++, Smalltalk, Objective C,
23| Eiffel, Object/Quick/Turbo Pascal, or CLOS.
-----1-----2-----3-----4-----5-----6-----7-----8

E-12

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Object-oriented programming languages
3|
4|
5| You've indicated that while you have one or more OOPL projects
6| implemented or canceled, none are currently pending or in progress.
7|
8| Please highlight the item that best describes the current status of
9| OOPL use at your location and press ENTER.
~|
12| No projects currently in progress, but future use expected
13|
14| Discontinued pending further evaluation
15|
16| Discontinued with no reevaluation currently planned
17|
18| Don't know/Other
~|
21| General purpose OOPLs: C++, Smalltalk, Objective C, Eiffel
22| Object/Quick/Turbo Pascal, CLOS
-----1-----2-----3-----4-----5-----6-----7-----8

Exhibit 4: Questions used to classify to Stage 6

I-09

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest Implemented OOPL application
3|
4| Please consider the largest application development effort implemented
5| at your location using a general purpose OOPL as the primary programming
6| language.
7|
8|
9| How large was this project in person-months of development time?
10|
11|           Please type a number from 1 to 9999 and press ENTER.
12|
13|           Person-months
14|
~
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

I-11

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest Implemented OOPL application
3|
4|
5|
6| Would you classify any OOPL applications developed and implemented at
7| your location as being mission critical?
8|
9|           Please highlight your answer and press ENTER.
10|
11|           Yes
12|           No
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

I-14

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Frequency of Use of OOPLs
3|
4| Has your location ever used OOPLs on at least 25% of new applications
5| development in a given year?
6|
7|
8| Please highlight your answer and press ENTER.
9|
10|           Yes
11|           No
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

Exhibit 5: Questions for Learning-Related Scale indicators

P-01

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Trends in new development
3|
4| Now we would like to learn more about the specific breakdown of
5| applications development-related activity at your location.
6|
7| * * *
8|
9| Please divide your location's total applications development and
10| maintenance effort for 1994 into the categories listed below.
11|
12| Please type a percent for each item and press ENTER. Make the total
13| equal 100%.
14|
15| % Maintenance and support of existing systems
16| % Significant enhancements to existing systems
17| % Development of new systems, or rewrites of existing systems
18| -----
19| % Total
20|
21|
22| All correct? (Y/N)

~
-----1-----2-----3-----4-----5-----6-----7-----8
r-00b

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Size and Scope of IT activities
3|
4|
5| How many application developers, including analysts, programmers,
6| and project leaders are currently engaged in applications
7| development at this location?
8|
9| Please type a number from 0 to 9999 and press ENTER.
10|
11| Size of professional staff developing/maintaining applications
12|
13|
14|
15| Include all non-managerial applications personnel located at your
16| location whether in the IT department or decentralized in line
17| departments.
18|
19| Exclude systems programming, operations personnel, support personnel
20| and "power users"
21|
22|
23|
24|
25|

-----1-----2-----3-----4-----5-----6-----7-----8
P-03 & P-04

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Trends in new development
3|
4|
5| What percentage of your major applications did you redevelop or
6| replace over the past three years?
7|
8| Please type a percent between 0 and 100 and press ENTER.
9|
10| % Will be replaced
11|
12|
13| What percentage of your major applications do you expect to redevelop
14| or replace over the next three years?
15|
16| Please type a percent between 0 and 100 and press ENTER.
17|
18| % Were replaced.
19|
20|
21|
22|
23|
24|
25|

Exhibit 6: Questions for Diversity indicators

Q-05a & Q-05b

```

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|
5| What percent of your applications development staff were using each
6| of the following languages in 1993? What do you expect for 1995?
7| (Since some programmers use more than one language, columns can sum
8| to more than 100%)
9|
10| Type a percentage for each item and press ENTER.
11|
12|           | 1993 | | 1995 |
13| Assembly language ..... % %
14| COBOL ..... % %
15| C language ..... % %
16| Another 3GL (Fortran, PL/, Basic, RPG) % %
17| A traditional 4GL (Natural, Ideal) . % %
18| C++ ..... % %
19| Smalltalk ..... % %
20| Another OO language ..... % %
21|
22|           All correct? (Y/N)

```

Q-02

```

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|
5| What was the division of new applications development activities over
6| the past three years across runtime platforms?
7|
8|
9| Type a percentage for each item and press ENTER. Please make
10| percentages total to 100%.
11|
12| % Traditional mainframe
13| % Traditional midrange (mini or supermini)
14| % Client-server with mainframe host
15| % Client-server with midrange host
16| % Client-server with desktop host
17| % Networked peer-to-peer workstations or PCs
18| % Standalone workstations or PCs
19| % Other
20| -----
21| % Total
22|
23|           All correct? (Y/N)

```

Q-03

```

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|
5| What was the division of new applications development activities over
6| the past three years across application types?
7|
8|
9| Type a percentage for each item and press ENTER. Please make
10| percentages total to 100%.
11|
12| % Primarily batch MIS/transaction processing
13| % Primarily on-line MIS/transaction processing
14| % Information retrieval / reporting / query / DSS
15| % Scientific / engineering / modeling / simulation
16| % Real-time or process control
17| % Office automation / personal productivity / groupware
18| -----
19| % Total
20|
21|           All correct? (Y/N)
22|
-----1-----2-----3-----4-----5-----6-----7-----8

```

Exhibit 6: Questions for Diversity indicators (cont)

Q-01a

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|   What other industries does your company operate in, and have
5|   application development staff (at this location) supporting?
6|
7|           Highlight each item that applies and press ENTER.
8|
9|           Manufacturing - Discrete (repetitive and job shop)
10|          Manufacturing - Process (continuous)
11|          Insurance and financial services
12|          Banking
13|          Healthcare/medical
14|          Education
15|          Government
16|          Business services and other services
17|          Transportation
18|          Communications
19|          Utilities
20|          Energy (oil/gas/coal production and refining)
21|          Wholesale trade
22|          Retail trade
23|          Other
24|          NONE/NO MORE/NEXT QUESTION
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

Exhibit 7: Questions for Related Knowledge indicators

0-03

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Application Developers: Skills and Background
3|
4| What percentage of the development staff at your location have had
5| experience in the areas below?
6|
7| Please type a percent for each item and press ENTER.
8|
9|           % Programming using the C language
10|
11|          % Developing client-server applications
12|
13|          % Developing PC or workstation-based applications
14|
15|          % Developing graphical user interfaces
16|
17|          % Developing data models using entity relationship diagrams
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

Exhibit 8: Questions for control variables-Host size and Education

R-04

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Size and Scope of IT Activities
3|
4|
5|
6|
7|
8|
9|
10|
11|
12|
13|
14|
15| What is the total number of employees in your company (division)?
16|
17|           Total number of employees
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

Q-01

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Application Developers: Skills and Background
3|
4| In this section we would like to ask a few questions about the skills
5| of your applications development staff.
6|
7|           * * *
8|
9| What percentage of the applications development staff at your location
10| fit in the categories listed below?
11|
12|
13| Please type a percent for each item and press ENTER.
14|
15|           % Bachelors degree is highest obtained
16|
17|           % Masters degree or higher
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

Exhibit 9: Questions for control variables-Host IT Size

3 In the columns below, please indicate how many people:

- A) Are employed, in total, at the location(s) you are involved with.**
- B) Are employed as IS personnel at these locations.**

	A	B
	All Employees	IS Personnel
Less than 10	<input type="checkbox"/> 16 7 - 1	<input type="checkbox"/> 16 9 - 1
10 - 24	<input type="checkbox"/> 2	<input type="checkbox"/> 2
25 - 49	<input type="checkbox"/> 3	<input type="checkbox"/> 3
50 - 99	<input type="checkbox"/> 4	<input type="checkbox"/> 4
100 - 249	<input type="checkbox"/> 5	<input type="checkbox"/> 5
250 - 499	<input type="checkbox"/> 6	<input type="checkbox"/> 6
500 - 999	<input type="checkbox"/> 7	<input type="checkbox"/> 7
1,000 - 2,499	<input type="checkbox"/> 8	<input type="checkbox"/> 8
2,500 - 4,999	<input type="checkbox"/> 9	<input type="checkbox"/> 9
5,000 - 9,999	<input type="checkbox"/> 10	<input type="checkbox"/> 10
10,000 or more	<input type="checkbox"/> 11	<input type="checkbox"/> 11

5 What is the total external IS spending on hardware, software, networking, and contract services for the current year (excluding payroll and overhead)?

	For location(s) you influence	For your entire organization
Less than \$500,000	<input type="checkbox"/> 21 0	<input type="checkbox"/> 22 0
\$500,000 - \$999,999	<input type="checkbox"/> 1	<input type="checkbox"/> 1
\$1 million - \$4.9 million	<input type="checkbox"/> 2	<input type="checkbox"/> 2
\$5 million - \$9.9 million	<input type="checkbox"/> 3	<input type="checkbox"/> 3
\$10 million - \$24.9 million	<input type="checkbox"/> 4	<input type="checkbox"/> 4
\$25 million - \$49.9 million	<input type="checkbox"/> 5	<input type="checkbox"/> 5
\$50 million - \$99.9 million	<input type="checkbox"/> 6	<input type="checkbox"/> 6
\$100 million - \$249.9 million	<input type="checkbox"/> 7	<input type="checkbox"/> 7
\$250 million - \$499.9 million	<input type="checkbox"/> 8	<input type="checkbox"/> 8
\$500 million or more	<input type="checkbox"/> 9	<input type="checkbox"/> 9

**Exhibit 10: Questions for control variables-Specialization
Sector and Environmental Complexity**

Q-04

```

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|
5|   For which of the following specialized functions do you have a
6|   staff of one or more full time professionals?
7|
8|   Please highlight each item that applies and press ENTER.
9|
10|  -> Advanced technology evaluation
11|  -> Quality assurance
12|  -> Data administration
13|  -> Methods & tools development
14|  -> Metrics and measurement
15|  -> System testing
16|  -> NONE/NO MORE/NEXT QUESTION
~
-----1-----2-----3-----4-----5-----6-----7-----8

```

Q-01

```

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|   What is the primary industry of the company supported by your
5|   location's application development staff?
6|
7|   Highlight the one item that best applies and press ENTER.
8|
9|   Manufacturing - Discrete (repetitive and job shop)
10|  Manufacturing - Process (continuous)
11|  Insurance and financial services
12|  Banking
13|  Healthcare/medical
14|  Education
15|  Government
16|  Business services and other services
17|  Transportation
18|  Communications
19|  Utilities
20|  Energy (oil/gas/coal production and refining)
21|  Wholesale trade
22|  Retail trade
23|  Other
~
-----1-----2-----3-----4-----5-----6-----7-----8

```

S-01

```

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Performance Objectives
3|
4|   Provided below is a scale for rating the importance of performance objectives.
5|
6|   Very unimportant                               Extremely important
7|
8|           1.....2.....3.....4.....5.....6.....7
9|
10|
11|   How important is each objective listed below for your location?
12|
13|   Type the scale number (1 to 7) for each object and press ENTER.
14|
15|   Very rapid applications development
16|   Cost effective development
17|   Meeting project schedules & budgets
18|   Deploying very high performance applications
19|   Deploying very high reliability applications
20|   Deploying very easy to use applications
21|   Deploying very easy to change applications
22|
23|           All correct? (Y/N)
~
-----1-----2-----3-----4-----5-----6-----7-----8

```

Exhibit 11: Questions for RDBs

L-01 & L-02

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Relational Database Management Systems (RDBMS)
3|
4|
5| In the questions that follow, we are interested in RDBMSs capable of
6| developing medium to large multi-user applications, such as IBM DB2,
7| Oracle, Sybase, Digital RDB, Ingres, Informix, ADABAS, Datacom/db.
8|
9| * * *
10|
11| Has your location ever installed a RDBMS for evaluation, trial or use?
12|
13| Highlight your answer and then press ENTER.
14|
15| Yes
16| No
17|
18|
19| In which year was it first purchased?
20|
21| Type in your answer and then press ENTER.
22|
23| 19
24|
25|

L-03a

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Relational Database Management Systems (RDBMS)
3|
4| Which of the following has occurred at your location related to RDBMS?
5|
6| Please highlight each item that has occurred and press ENTER:
7|
8| Approval of RDBMSs for use on production applications
9| Implementation of a multi-user application using RDBMS
10| Implementation of a large, mission critical application using RDBMS
11| Use of RDBMS on at least 25% of all new development in the same year
12| NONE/NO MORE/NEXT QUESTION
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|

L-03b

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Relational Database Management Systems (RDBMS)
3|
4|
5| In which year did your location first use an RDBMS on at least 25% of
6| new applications development?
7|
8| 19
9|
10|
11|
12|
13|
14|
15|
~

CHAPTER 4. ALTERNATIVE MEASURES OF ORGANIZATIONAL INNOVATIVENESS: A CONCEPTUAL ANALYSIS

One of the central tasks innovation researchers face is to devise appropriate measures of the timing and extent of innovation adoption/implementation, whether by individuals or larger aggregates. These measures are viewed as indicators of adopter "innovativeness"; that is, adopting units that exhibit early and sustained adoption activities are viewed as more innovative with respect to new ideas and technologies than those who do not. Traditional innovativeness measures focus on the adoption event (e.g., physical acquisition of the innovation), and include such options as time of first adoption, dichotomous adoption/non-adoption, and number of adoptions across a set of innovations. These sorts of measures, while well suited to some purposes, have been criticized when used as the sole indicators of adopter innovativeness [Downs and Mohr 1976; Tornatzky and Klein 1982]. Time of adoption, for example, suffers because the soon-to-adopt are indistinguishable from the never-to-adopt.

In the past decade or so, a variety of newer measures of organizational innovativeness have been proposed, including degree of internal diffusion (a "breadth" of use measure), level of infusion (a "depth" of use measure), and assimilation stage achieved. These measures tend to be richer than traditional ones, and have the additional advantage of distinguishing differences in post-adoption behaviors. In situations where post adoption behaviors vary widely—such as when some organizations deploy the innovation thoroughly while others not at all—these newer measures may be especially pertinent.

This chapter provides a conceptual analysis of six prominent measures of the adopter innovativeness concept: 1) Time of Adoption, 2) Dichotomous Adoption, 3) Aggregated Adoption, 4) Diffusion, 5) Infusion, and 6) Assimilation Stage. For each measure, the analysis includes a description of the origins of the measure, potential advantages and limitations, suggested

operationalizations for the case of software process innovations, and guidance on when and how the measure is likely best applied.

There are two primary purposes in analyzing alternative measures. First, the result of this review supports the assertion that Assimilation Stage, the primary innovativeness measure employed for the OOPL study presented in Chapter 3, was in fact the most suitable measure for this purpose. Second, this analysis lays the conceptual foundation for an empirical analysis of alternative innovativeness measures presented in Chapter 5. Together, the conceptual and empirical analyses should prove useful to future researchers in selecting the most appropriate innovativeness measure and in matching the most appropriate research designs and statistical techniques to selected measures.

4.1 TIME OF ADOPTION

Time of Adoption is the most traditional measure of adopter innovativeness. In fact, Rogers defines innovativeness explicitly as "the degree to which an individual or other unit of adoption is relatively earlier in adopting new ideas than other members of a system" [Rogers 1983, p. 22]. For many years, this was the virtually the only measure employed [Downs and Mohr 1976].

Potential Strengths and Limitations

Time of Adoption is typically operationalized as the year of the adoption event. Time of Adoption has several advantages as a measure. It is simple, comparatively easy to capture, and relates directly to the traditional notion of innovativeness. Time of Adoption also has the advantage of serving as the basis for the dependent variable in most mathematical models of the macro diffusion process. As a result, Time of Adoption can serve as the linchpin linking the two major styles of diffusion research, i.e., adopter studies explaining the antecedents of individual adopter innovativeness, and diffusion studies describing the macro pattern of diffusion across a population

of potential adopters. Finally, when paired with event history analysis techniques, Time of Adoption can be used as the independent variable in models incorporating time-varying predictors [Russo 1991; Singer and Willett 1991; Pennings and Harianto 1992a].

However, Time of Adoption suffers from some important potential limitations. The first is some inevitable arbitrariness in defining the adoption event. Should adoption be defined as physical acquisition of technical artifacts? As an organizational commitment to deploy the innovation? As actual deployment of the innovation? Alternative definitions not only affect descriptive results, such as the percentage of the population that can be reported as "adopting" by a certain date, but also, potentially, the magnitude of observed effects for individual predictor variables.

Second, unless the researcher waits until the entire population has adopted, Time of Adoption is a "censored" measure because it does not capture when, if ever, adoption will occur for the pool of current non-adopters. Censoring is a greater concern to the extent that the focal innovation has not yet been widely adopted and the pool of non-adopters is large. When censoring is present, more advanced statistical approaches, and, in particular, event history analysis, are necessary to properly estimate causal models. If ad-hoc methods are used, such as eliminating non-adopters from the analysis or coding them to the maximum observed duration, serious biases in estimated parameters can result [Allison 1982, p. 65].

Even supposing appropriate modeling techniques are used, Time of Adoption still does not distinguish gradations of innovativeness among those *yet to adopt* and gives no insight into post-adoption differences among adopters, such as speed or extent of implementation.

Proposed Operationalizations

Despite potential limitations listed above, there will be circumstances when the advantages of Time of Adoption outweigh its disadvantages. Time of Adoption is most appropriate as a primary outcome measure when the following circumstances are in effect: 1) data are being gathered well along in the diffusion process, and hence, the pool of non-adopters is comparatively small; 2) adoption is less easily reversed, i.e., because sizable indivisible investments are involved; 3) comparatively little discontinuance or other forms of radical differences in post-adoption innovativeness are present, and 4) the researcher is interested in time-varying predictors.

Two alternative operationalizations are proposed for Time of Adoption for SPIs: 1) date of first purchase of the innovation, and 2) date the innovation was first approved for use on a particular production application, where a production application is defined as one that is intended to be delivered to users and operated and maintained on an ongoing basis. The advantage of the first operationalization is that it corresponds to the operationalization employed in studies based on sales data, including most marketing studies, and thus supports a strong link to the cumulative tradition of diffusion research. The second measure is probably preferably in most cases, in that it comes closer to the standard suggested by Tornatzky *et al.* for a useful concept of adoption: "In situations where a presumably irreversible act is involved . . . the adoption construct may be . . . defensible. It is less useful in circumstances where adopters may 'unadopt' at little or no cost to themselves . . ." [1983, p. 24]. Approval of an SPI for production use is arguably the most important milestone in the assimilation process, in that the organization is, in effect, committing to a long term relationship with the SPI—at minimum the expected life of the production application.

4.2 DICHOTOMOUS ADOPTION

Dichotomous Adoption (by a given date) is essentially a crude variant of the Time of Adoption measure, and as such, shares many of the advantages and limitations of the Time of Adoption measure.

Potential Strengths and Limitations

Dichotomous Adoption is easy to capture and interpret, and is particularly well suited to models that only aim to draw broad distinctions between adopters and non-adopters as a class. When used in a logit model, for example, it is possible to produce results such as "this model correctly classifies X% of the population as adopters or non-adopters" (see, for example, [Gatignon and Robertson 1989]). Some readers may find such a result easier to interpret than a comparable model that explains "Y% of the variance in innovativeness". In addition, Dichotomous Adoption has performed well even when a small proportion of the population has adopted. Brancheau and Wetherbe [1990], for example, use a dichotomous measure to successfully distinguish the first 16% to adopt spreadsheets ("innovators" and "early adopters" according to Rogers' terminology) from everyone else.

The primary limitations of Dichotomous Adoption are insensitivity to differences in innovativeness within both adopter and non-adopters populations, and the arbitrariness of elevating one event in the chain of assimilation activities as the single critical event worthy of explanation.

Proposed Operationalizations

Although Dichotomous Adoption is a thin measure of innovativeness, it can be appropriate for more exploratory studies, studies focused on establishing simpler relationships, or studies that have an especially compelling single event to explain. In addition, since Dichotomous Adoption is inevitably obtained during the course of capturing other outcome measures (i.e., Time

of Adoption, Aggregated Adoptions, Assimilation Stage), it deserves attention as a potential variable supporting secondary analyses.

In terms of operationalizations, two events are proposed as the basis for forming a dichotomous measure: 1) purchase of the SPI, and 2) approval of the SPI for production use. These alternatives parallel those recommended for Time of Adoption, and have the same relative advantages and disadvantages. Purchase of an SPI has the advantage of being consistent with much of the prior work on diffusion of commercially available products, while approval for production has the advantage of being a more meaningful candidate as "the" event to explain in the assimilation process.

4.3 AGGREGATED ADOPTION

Starting in the 1960's and 1970's, a large number of studies of organizational innovativeness were conducted that took Aggregated Adoption—the number of adoptions across a set of innovations—as the primary dependent variable. This preference for Aggregated Adoption carried over into the stream of studies conducted by IT researchers beginning in the 1980's [Zmud 1982; Zmud 1983; Zmud 1984; Nilakanta and Scamell 1990; Grover and Goslar 1993]. Meyer and Goes summarize the apparent reasons for the popularity of this measure of innovativeness as: 1) ease of measurement (respondents can simply check off which innovations have been adopted), 2) a desire for greater reliability and generalizability compared with single innovation studies, and 3) an implicit assumption that the more innovations adopted the better [1988].

Potential Strengths and Limitations

The main advantage of Aggregated Adoption is that it counters one of the major concerns associated with single innovation studies, namely, that any results produced are idiosyncratic to that one innovation. However, its popularity notwithstanding, Aggregated Adoption has been severely criticized,

with these criticisms falling into two main classes, those having to do with what is being aggregated, and those having to do with aggregation *per se*.

On the first point, Downs and Mohr have questioned whether Aggregated Adoption reliably maps to other common notions of innovativeness, such as earliness of adoption or extent of implementation [1976]. They argue that, even though a particular organization has adopted many innovations, this does not necessarily mean most of them were adopted early, or that any of them are used in depth.

The second class of criticisms, having to do with aggregation *per se*, are potentially more limiting. The very act of aggregation assumes a degree of homogeneity among a set of innovations that may be seriously misplaced. Different innovations can have strong, innovation-specific predictors. Organizational wealth, for example, may be a strong predictor of high cost innovations, but not low cost ones [Downs and Mohr 1976]. If high and low cost innovations are aggregated, then the effect of this variable will be over-estimated for low cost innovations and under-estimated for high cost innovations. In addition, some organizations, owing to the particulars of their competitive situation, might quite justifiably be more innovative with respect to some innovations than others. For example, a piece of medical equipment might create an opportunity for diversification for one hospital, but jeopardize well established services in another [Meyer and Goes 1988]. In addition, when aggregated measures are used, innovation researchers lose out on the opportunity to include predictors whose values vary across innovations in the set. These include the characteristics of the innovations themselves (e.g., complexity) and what Downs and Mohr have termed "secondary characteristics" of organizations, characteristics that vary depending on the innovations under study. To illustrate, if centralization tends to vary across departments within firms, then centralization is a secondary characteristic in a study that includes innovations adopted by different departments. Because of these limitations, Downs and Mohr have

argued that Aggregated Adoption should never be used as an innovativeness measure and that instead, innovation researchers should either use single innovation designs, or the adoption-decision design.¹

Countering this view is a recent meta-analysis performed by Damanpour [1991] that indicates fairly good stability of prior results in studies using Aggregated Adoption as the outcome variable, at least when the focus is on organizational characteristics encouraging adoption. Dampamour rejects the claim that motivated Downs and Mohr's criticism of Aggregated Adoption in the first place, namely, that prior results in the study of organizational innovation are very unstable, and that the use of Aggregated Adoption are a significant contributor to this instability.

The view taken here is that Aggregated Adoption can be a useful measure of innovativeness so long as the limitations of this measure are actively managed in the research design, and carefully considered in the interpretation of results.

Proposed Operationalizations

Four alternative operationalizations of Aggregated Adoption are proposed: 1) number of adoptions, where adoption is defined as acquisition, 2) number of adoptions, where adoption is defined as approval for production use, 3) average Assimilation Stage and 4) average Time of Adoption. The first two are simply aggregations of the two alternative definitions of the adoption event suggested earlier. The latter two operationalizations, besides being generally richer measures, address the concern that number of adoptions

¹With the adoption-decision design, the adoption decision is the unit of analysis. Adoptions are not aggregated, but rather, each adoption for each organization is treated as a separate data point. This design can support three classes of predictors: 1) innovation characteristics, e.g., complexity (the same value for complexity is duplicated for all adoptions of a given innovation), 2) organizational characteristics, e.g., size (the same value for size is duplicated for all adoptions by a given organization), and 3) characteristics of the innovation-organization intersection, e.g., management support (unique values exist for every innovation-organization combination). See, for example, Meyer and Goes [1988].

might not reliably map to other traditional notions of innovativeness. If it turned out, as Downs and Mohr have suggested is common, that a significant proportion of organizations were prone to acquire many innovations, but to deploy relatively few, this could be detected by using average Assimilation Stage as the outcome measure and/or examining the degree of correlation between average Assimilation Stage and number of acquisitions. Likewise, average Time of Adoption can be used to examine the plausibility of the assumption that those organizations that acquire many innovations also tend to be among the early adopters of particular innovations.

The following five prescriptions are suggested for studies employing Aggregated Adoption:

- 1) Use sets of innovations that are homogenous in terms of expected predictors, and homogenous in the expected ideal profile of adopting organizations. This is the approach taken by Zmud [1982], for example. In his study of modern software practices, he aggregated two sets of three innovations each, with one set being viewed as equally compatible for most organizations in the study, and the other being viewed as equally incompatible with most organizations in the study.²
- 2) Consider whether predictor variables may plausibly have dramatically varying strength of association for different innovations in the set. If so, either segregate the innovations into separate groups during analysis, or avoid the inclusion of such predictors in the first place.
- 3) Include organizational characteristics as predictors only when they are either primary characteristics, or fall toward the primary end of the spectrum. While Downs and Mohr set out primary and secondary characteristics as absolutes, in truth, many organizational characteristics that are strictly secondary characteristics, might, for a particular set of comparatively homogenous innovations, be closer to primary characteristics. For example, centralization of decision making probably takes on the same value for all modern software practices in the Zmud study mentioned above because the same department (IT) and probably the same individuals were likely to be involved in all adoption decisions. In another study, where

²Contrast this to a recent study that aggregates 15 telecommunication technologies, ranging from simple fax equipment to wide area networks and video conferencing [Grover and Goslar 1993].

innovations adopted by different departments are included, centralization falls toward the secondary end of the spectrum, because the level of centralization of decision making can vary widely across departments within the same organization.

- 4) Use average Time of Adoption or average Assimilation Stage instead of, or in addition to, number of adoptions as the basis for the aggregated measure; critically analyze the plausibility of the assumption that number of adoptions is a fair stand-in for these other notions of innovativeness.
- 5) Estimate the model for innovations individually, and report and interpret results at both the disaggregated and aggregated levels.

4.4 EXTENT OF IMPLEMENTATION

An alternative to Time of Adoption or Aggregated Adoption proposed by Downs and Mohr is the extent to which the innovation has been implemented. They argued that because all organizations are not likely to implement innovations at the same rate, Time of Adoption may not be an adequate proxy for extent of implementation. Since the extent to which the innovation has been implemented comes "closer to capturing the variations in behavior that we really want to explain," they argued that it should be captured and modeled as a separate dimension of innovativeness [Downs and Mohr 1976, p. 709]. Tornatzky and Klein [1982] reiterated this call and suggested two sub-dimensions of extent of implementation: 1) breadth of use (i.e., extent of utilization), and 2) depth of use (i.e., the degree to which key features or aspects of the innovation are implemented). In studies of information technology adoption, this first dimension has been termed "diffusion" and the second dimension "infusion".³

³Two other sub-dimensions of extent of implementation have also been suggested: *acceptance* and *routinization*. Acceptance is the extent to which individual organizational members are committed to using the innovation [Cooper and Zmud 1990]. Routinization is the extent to which work systems surrounding the innovation have been altered such that the innovation is no longer perceived as being new or out-of-the ordinary [Cooper and Zmud 1990; Yin 1979]. For a description of these two dimensions, the reader is referred to the excellent review provided by Saga and Zmud [1993].

4.4.1 Diffusion

Diffusion captures the extent of spread of use of the innovation within adopting organizations across people, projects, tasks, or organizational units. It is analogous to the traditional notion of diffusion, except the system through which the innovation is spreading is not the economy or an industry, but a single organization. Diffusion is especially appropriate in studies of innovations where implementation may occur gradually, on a person by person basis (e.g., personal computers). It is less likely to be useful in the case of technologies like material requirements planning (MRP), where initial implementation tends to be a monolithic project spanning a substantial portion of the ultimate user base.

Recent examples of IT adopter studies employing Diffusion (or a similar measure) as an outcome variable include Zmud and Apple [1992], who studied supermarket scanners, Bretschneider and Wittmer [1993], who studied microcomputers, and Howard and Rai [1993], who studied CASE tools. In addition, a stage of adoption scale originally developed by Zmud [1982; 1983; 1984] and later employed by others [Nilakanta and Scamell 1990; Grover and Goslar 1993] can also be viewed as a rough categorical measure of Diffusion.⁴

Diffusion can be used as either an alternative measure of general innovativeness or as a narrower measure of post-adoption innovativeness. In the first case, non-adopters are assigned a Diffusion value of zero and are included in the pool with adopters during analysis. Diffusion can then be seen as a combined measure of Time of Adoption (because those that adopt earlier, other things being equal, have more time to reach any given level of Diffusion) and robustness of implementation. To use Diffusion as a

⁴Zmud's scale has four explicit positions: "a few people/projects regularly use it", "a number of people/projects regularly use it", "most people/projects regularly use it," and "usage has become standard". This scale also has an implicit fifth position because organizations that were not using an innovation at all, or who are using it only on an experimental basis were effectively coded to zero.

narrower, post-adoption-only measure, non-adopters are excluded from the pool of cases during analysis, and Time of Adoption is included as a control variable (see, for example, [Rai 1993]). The remainder of the discussion will focus on this approach, as it offers the greatest contrast to traditional measures.

Potential Strengths and Limitations

As a measure of post-adoption innovativeness, Diffusion has the advantage of focusing directly on an important aspect of innovation that has been little studied. Because analysis is limited to just those organizations that have adopted, and Time of Adoption is used as a control variable, the confound with Time of Adoption is minimized. In addition, using Diffusion in this way allows the use of predictor variables (e.g., implementation tactics) that only make sense post-adoption.

However, using Diffusion in this manner creates a new set of problems. First, removing non-adopters obviously reduces the sample size, and hence, the corresponding power of the analysis. Second, the researcher is limited to explaining differences among only the most innovative segment of the population (those that have already adopted). Depending on how small the segment is, this can be quite a challenging proposition—the extent of variation in independent variables is likely to be less, and non-linearities in effects of variables (e.g., diminishing returns) may obscure genuine relationships. For example, it might turn out in a given sample that almost all organizations that have already adopted are large, but those with greater Diffusion are not discernibly larger, in general, than those with lesser Diffusion. This can lead to the potentially erroneous conclusion that size does not matter, when it might be that had many small organizational adopted, they would have had significantly less robust diffusion compared with the larger organizations, and this fact would have been statistically significant. While this problem can be lessened by waiting until the

innovation of interest has reached a later point in the overall diffusion cycle, this serves to reduce the practical interest in study results.

Proposed Operationalizations

Four alternative operationalizations of the Diffusion concept are proposed: 1) the percentage of developers currently using the SPI, 2) the peak percentage of developers in the organization that ever used the SPI, 3) the percentage of all projects on which the SPI has been used (during some time span) 4) the percentage of "appropriate" projects on which the SPI has been used (during some time span). Continuous approaches to scoring Diffusion, such as actual percentage of people or projects using the technology are recommended over categorical approaches to scoring because they have greater discriminating power.

The first operationalization has the advantage of being easiest to capture, and it also most closely matches past usages. The second operationalization is more subtle. In some organizations use of an innovation may have passed its peak usage because it has become obsolete in that organization and is being replaced by a newer technology. Since such organizations would be considered by most as more innovative than those for which the focal innovation is still in the process of being deployed, a measure of the peak number of developers that ever used the innovation can provide a more meaningful measure if there are likely to be many organizations in the process of replacing the focal innovation.

The third and fourth operationalizations are based on a distinctive facet of SPIs, namely, that they are typically deployed on a project by project basis as new development opportunities arise. It might therefore be argued that the third measure, which captures the percentage of projects on which the focal innovation has been used, provides a more appropriate innovativeness measure than the first two individual-oriented measures. The fourth operationalization, which uses only projects deemed as innovation-

appropriate in the denominator, provides a further refinement especially appropriate for more specialized innovations. For innovations where organizations markedly vary in the proportion of projects which are reasonable candidates for use of the innovation, the fourth operationalization has the advantage of not penalizing those organizations with a smaller proportion of candidate projects.

4.4.2 Infusion

Infusion is a measure that seeks to distinguish organizations that use an innovation in a comprehensive way from organizations that use only its most basic features. It is well known among software engineering professionals that adopting organizations need not utilize the full potential of an innovation: the use of 4GLs and relational databases can be crippled by "stone-age" programming techniques [Nolan 1986]; CASE can be used only as glorified drawing program [Kemerer 1989]; and Ada can be used without employing any elements of the "Ada philosophy," such as information hiding [Bayer and Melone 1989]. Since the claimed benefits of an innovation are usually predicated on adopters employing its more sophisticated features, the extent to which these features actually are being used represents a crucial and understudied area of innovation research.

Infusion is defined here as *the extent to which an innovation's features are used in a complete and sophisticated way, approaching the ideal configuration of use envisioned by the innovation's designers and proponents*. The exact definition of Infusion has been evolving, but later usages have in common this notion of increasing levels or configurations of

use.⁵ Zmud and Apple [1992], for example, operationalize Infusion of supermarket scanners as a Guttman scale with three levels: 1) front end control, 2) sales analysis, and 3) inventory management. Cooper and Zmud [1990] use an existing four level scheme to operationalize Infusion for MRP systems, ranging from Class D use (MRP system exists mainly in the data processing department) to Class A use (the MRP is a closed-loop system, used for priority planning and capacity planning).

Potential Strengths and Limitations

Infusion captures a crucial, and, until recently, virtually ignored element of innovativeness. It might be argued that unless some minimum level of Infusion is attained by an adopter, other measures of innovativeness (Time of Adoption, Diffusion) become virtually meaningless for that adopter because the innovation will have been adopted and deployed in name only.

As the above examples make clear, Infusion will always be an innovation-specific measure. This constitutes both a great advantage and perhaps the most serious limitation of the measure. Because it is innovation specific, it can be custom-tailored to capture a rich array of innovation specific outcomes. However, this context-sensitivity comes at the price of a reduced ability to generalize and compare results across studies to other innovations, or to aggregate results across innovations within the same study.

Three other potential limitations of Infusion are worth noting. First, Infusion only makes sense for organizations that have already adopted. While this opens the door to including predictor variables that also only

⁵Zmud and Apple describe level of infusion as "a sequence of configurations across discrete levels of use, with advanced incorporation enabling the deeper and more comprehensive embedding of an innovation within an organization's operational and/or managerial work systems" [Zmud and Apple 1992, p. 150]. This definition contains two parts: how the innovation is used, and the presumed effects of its use. Earlier usages focused only the presumed effects of using innovations in complete and sophisticated way [Sullivan 1985]. The position taken here is that the infusion concept should be limited to the levels of use idea, and that presumed effects should be operationalized as separate variable.

make sense post-adoption, it also limits analysis to a smaller, more homogeneous population, with the accompanying problems discussed earlier (lower statistical power, less variation in predictor variables). Second, scales for Infusion are likely to be difficult to design and measure. In most cases, researchers will have to design and validate their own Infusion instruments, and it is likely that several questionnaire items will be required to reliably classify respondents. Third, Infusion is likely to confound with Time of Adoption, in that the longer an organization has had an innovation in house, the longer it has had to infuse the technology.⁶ This suggests that, to the extent a researcher wishes to distinguish the propensity to adopt innovations early and the propensity to quickly reach higher levels of Infusion, Time of Adoption must also be captured and used as a control variable.

Proposed Operationalizations

Since Infusion is specific to the innovation, it is not possible to recommend operationalizations that generalize to all SPIs. However, it is still appropriate to consider the form operationalizations can take. Infusion can be operationalized using either a Guttman scale, describing distinct increasing levels of use, or a summative scale, containing multiple items and potentially, multiple dimensions. Examples of the former approach include Cooper and Zmud's [1990] scale for MRP Infusion and Zmud and Apple's [1992] scale for supermarket scanners. An example of the latter approach is Howard and Rai's 13 item summative scale for CASE "injection depth" [Howard and Rai 1993]). Which approach is best will depend on the technology being studied, and the relative maturity of that technology. Where a sensible Guttman scale can be constructed, this is probably the preferred approach because it is simpler to describe and interpret, and is most consistent with past usages.

⁶Zmud and Apple [1992] found, for example, that Infusion and Time of Adoption for supermarket scanners were correlated at $r=.45$.

For some technologies, however, it may not be possible to create a useful Guttman scale, because no obvious grouping of features at distinct levels of use can be devised. This seems to be the case, at present, for object-oriented programming languages. A highly infused adopter of OOPs might exhibit the following kinds of traits: use of encapsulated objects for all components of applications (rather than just for the user interface component); use of complementary object technologies, such as object-oriented methodologies and object-oriented databases; and use of commercial class libraries. Yet, as of this writing, no Guttman-like scale to distinguish levels of use of OOPs has yet been proposed. Furthermore, it is not clear that the richness of the above three dimensions could be captured within a single Guttman scale. So, rather than using a different label for technologies like this, it is proposed that the notion of Infusion be expanded to include summative scales for instances where a Guttman scale is not appropriate.

4.5 ASSIMILATION STAGE

Innovation researchers have long observed that the organizational innovation process typically proceeds through a series of identifiable stages, ranging from initial awareness of an innovation to adoption and implementation. Numerous stage models have been proposed, and depending on the focus of the research, these models have variously been called stages of "adoption" [Ettlie 1980], stages of "implementation" [Cooper and Zmud 1990], and stages of "assimilation" [Meyer and Goes 1988]. A fairly typical example is Ettlie's six stage model: 1) awareness, 2) interest, 3) evaluation, 4) trial, 5) adoption and 6) implementation. The more encompassing term, "assimilation stage," will be used here.

Although stage models have long been used for descriptive purposes, stage achieved (by a certain time) has apparently been overlooked as a potential measure of adopter innovativeness until very recently [Meyer and Goes 1988].⁷ This is unfortunate, because, as argued below, Assimilation Stage appears to remedy many of the limitations of other innovativeness measures.

Potential Strengths and Limitations

As an operationalization of innovativeness, Assimilation Stage is a combined measure of earliness of assimilation activities, speed of assimilation activities, and an absence of stalling or discontinuance. As a result, Assimilation Stage is a richer measure than Time of Adoption or Dichotomous Adoption. Since Assimilation Stage has multiple categories to distinguish organizations yet to adopt, it is sensitive to pre-adoption differences in innovativeness. In addition, since it has multiple categories to distinguish organizations that have already adopted, Assimilation Stage is sensitive to post adoption differences in innovativeness among the set of prior adopters. Assimilation Stage also has the advantage of being general: with a few exceptions, any organization can be classified, and the same measurement instrument can, with minor tailoring, be used for broad classes of innovations. A measure such as Infusion, by contrast, is innovation specific, and can only classify organizations that have already adopted.

Assimilation Stage is not without limitations. First of all, any stage model represents an idealized or normative sequencing of stages, when, in reality some organizations may have a different sequencing, may omit stages, or may conduct multiple stages in parallel [Ettlie 1980]. While most of these problems can be remedied by carefully tailoring the stages to suit the innovation(s) under study, some organizations will always be atypical.

⁷Although Cooper and Zmud refer to a stage model in their MRP study, they do not use stage achieved as an outcome variable [1990]. Rather, their stage model serves only as a conceptual backdrop for the two outcome variables they do employ, a dichotomous variable for adoption, and a Guttman scale for level of infusion.

Second, Assimilation Stage is more difficult to measure than traditional operationalizations like Time of Adoption.⁸ Third, one of the advantages of Assimilation Stage—the fact that it combines multiple elements of innovativeness—can also be viewed as a weakness. To the extent that a researcher seeks to preserve Time of Adoption and speed of deployment as separate dimensions, for example, other measures must be captured instead of, or in addition to, Assimilation Stage.

Finally, the use of Assimilation Stage is predicated on the principle that the predictors of organizational innovativeness under study are influential during most or all stages, and more importantly, have the same directionality of influence (i.e., positive or negative). However, it has been argued that for some structural features of organizations—such as centralization and formalization—effects may be differently directioned for different stages. Essentially, the argument is that more "organic" organizations with lower centralization and formalization should be more prone to embracing new ideas, and hence should be more likely to initiate innovative activities, while "mechanistic" organizations, with higher centralization and formalization, should be better able to obtain consensus and push through the later stages of the innovation process, and hence should be more likely to adopt and implement those (presumably fewer) innovations that they do initiate. This intriguing notion has been asserted by many [Zaltman *et al.* 1973; Downs and Mohr 1976; Tornatzky and Klein 1982; Zmud 1982], although actual empirical studies have not generally shown this to be true [Nilakanta and Scamell 1990; Damanpour 1991; Grover and Goslar 1993]. Nevertheless, this is a reasonable concern to evaluate. (Some methods for doing this were demonstrated in the empirical analysis provided in Section 3.5.2.)

⁸To perform a careful classification of assimilation stage for object-oriented programming languages a total of 12 questions, most with several items, were needed.

Proposed Operationalizations

A six stage model of assimilation is proposed for use in studies of software process innovations (see Table 4.5.1 below). (A comparison of this model with some other prominent stage models, i.e., Meyer and Goes [1988], Ettlie [1980], and Cooper and Zmud [1990], was provided in Chapter 3.)

Table 4.5.1: Definition of Assimilation Stages

Stage	Criteria to enter stage	Typical behaviors
1. Awareness	Key decision makers are aware of the SPI	Passive, accidental or opportunistic learning about the SPI through media, word of mouth, vendor promotions, etc.
2. Interest	The organization is committed to actively learning more about the SPI	Directed exploration of specific products and adoption issues Matching products to needs
3. Evaluation/trial	The organization has acquired a specific innovation-related products and has initiated evaluation or trial	Acquisition and installation of tools Hands-on evaluations Trial projects
4. Commitment	The organization has committed to use a specific SPI product in a significant way for one or more production projects	Initial development of production projects Initial training and infrastructure development
5. Limited deployment	The organization has established a program of regular but still limited use of the SPI product	Development and implementation of selected production systems using the SPI Continued development of infrastructure
6. General deployment	The organization has reached a state where the SPI is used on a substantial fraction of new development, including at least one large and one mission critical system	Ongoing use on a wide variety of projects, including large, mission critical ones Continued development of infrastructure

Assimilation Stage offers a unique combination of richness and generality. It is especially appropriate for studies where sensitivity to pre-adoption differences is need, e.g., because data are being collected relatively early in the diffusion process; where a more reliable and general measure is needed, e.g., because only a single innovation is being examined; and where at least some

sensitivity to post-adoption differences is needed, e.g., because many organizations may adopt without deploying, or at least not very widely. (All of these circumstances exist for the OOPL Assimilation Study described in Chapter 3.) In addition, Assimilation Stage is appropriate to use for any of the three main design options for adoption studies: 1) the single innovation design, 2) the multi-innovation design (using average stage) and 3) the innovation-decision design.

4.6 SUMMARY

This chapter has described and critiqued six alternative approaches to measuring the organizational innovativeness concept. Table 4.6.1 provides a summary of the characteristics of these measures ranked on a scale from + to + + + +. As this table makes clear, each measure has a unique combination of potential strengths and weaknesses, and no one measure dominates any of the others. Which innovativeness measure is most appropriate as the primary outcome variable for a given study will depend on goals of the research, the design of the study, and the nature of the innovation and organizations included. Traditional innovativeness measures tend to be thin, but very general. Diffusion and infusion measures are richer, but have more restricted application and generalizability. Assimilation Stage falls in between the two on both counts, and avoids many of the particular limitations of other measures.

It is also hoped that this discussion has made clear the value of capturing secondary measures, not only to provide a more rounded picture of adopter innovativeness, but to support analyses of specific validity threats associated with a given study's primary measure. Table 4.6.2 suggests some natural combinations of measures. Time of Adoption provides a valuable, and perhaps necessary complement to Infusion, Diffusion, and Assimilation Stage. Assimilation Stage complements all of the other measures, but in particular, the traditional measures of Time of Adoption, Dichotomous Adoption, and Aggregated Adoption.

4.6.1 Characteristics of Innovativeness Measures

Measure	Ease of capture	Ability to classify all organizations	Generality across innovations	Richness	Circumstances discouraging use
1. Time of Adoption	+++	++	+++	++	a,b,c,d
2. Dichot. adoption	++++	+++	+++	+	a,b,c,e
3. Aggregate adoption	++	+++	+++	++	a,b,c,e,f
4. Diffusion	++	++	++	+++	d,e,g
5. Infusion	+	+	+	++++	d,e,h
6. Assimilation stage	+	+++	++	+++	e,i

a-No single, compelling adoption event exists

b-Large variability in post-adoption behaviors (discontinuance, stalling, low infusion)

c-Researcher interested in post-adoption factors

d-Pool of prior adopters relatively small

e-Researcher interested in time-varying factors

f-Researcher interested in innovation-specific factors or secondary organizational characteristics

g-Implementation does not occur gradually across people/tasks/projects/organizational units

h-No obvious increasing levels/configurations of use exist

i-Researcher interested in predictors with differently directed effects for different stages

Table 4.6.2 Distinctions Captured in the Measure

Measure	Differences among non-adopters	Differences in adoption timing among adopters	Differences in implementation among adopters
1. Time of Adoption	No	Yes	No
2. Dichotomous Adoption	No	No	No
3. Aggregated adoption	No	No	No
4. Diffusion	No	No	Yes
5. Infusion	No	No	Yes
6. Assimilation Stage	Yes	No	Yes

Finally, this discussion has highlighted the value of a newer innovativeness measure that has not been used in any prior IT adoption study, namely,

Assimilation Stage. Assimilation Stage provides reasonable levels of generality and richness, and is the only measure that distinguishes differences in innovativeness among non-adopters. In addition, it is equally suitable for all three innovation research designs—single innovation, multi-innovation, and adoption-decision.

CHAPTER 5. ALTERNATIVE MEASURES OF ORGANIZATIONAL INNOVATIVENESS: AN EMPIRICAL ANALYSIS

The conceptual analysis of alternative innovativeness measures presented in Chapter 4 raises a number of issues that should be of interest to diffusion researchers. This chapter presents an analysis of four empirical issues, based on data on the adoption of three software process innovations (SPIs):

- Issue 1: Do richer innovativeness measures generally lead to more strongly predictive models?
- Issue 2: Does Aggregated Adoption suffer from a lack of correspondence with other innovativeness measures?
- Issue 3: Can the results achieved with Aggregated Adoption be confidently generalized to particular innovations in the set?
- Issue 4: Are the extent of implementation measures (Diffusion and Infusion) as strongly predictive as the other four measures?

The data set employed here was constructed using survey responses from the same collection of IT departments that formed the basis of the empirical analysis in Chapter 3. The dataset contains adoption data for three SPIs: object-oriented programming languages (OOPLs), relational database management systems (RDBs), and computer aided software engineering tools (CASE). Of these three SPIs, OOPLs are of greater current interest, and the dataset therefore contains more extensive adoption data for this innovation. As a result, OOPL data will support the bulk of the analysis presented below.

The focus of this chapter is on analyzing the impact of employing different innovativeness measures on model prediction, for models containing the same basic set of predictors. As described below, the set of predictors used in most models is a slightly more parsimonious collection taken from the set used in Chapter 3.

5.1 PREDICTOR VARIABLES

A total of nine explanatory variables are used in various combinations in the analyses presented below. Of the nine variables, six are generic in the sense that they should be applicable regardless of the particular SPI under study or innovativeness measure employed. These variables, which will be referred to as "generic" predictors, include Learning-Related Scale, Diversity, IT Size, Host Size, Education, and Environmental Complexity. Regarding the other three variables, Related Knowledge is specific to OOPLs; deployment facilitators (Facilitators) and time since acquisition (Time) are only applicable when post-acquisition measures are being predicted, because they presume acquisition has already occurred.

This collection of variables differs from the one used in Chapter 3 as follows. First, the two variables from Chapter 3 with the fewest significant relationships in preliminary modeling runs (Sector and Specialization) were dropped in the interest of achieving a more parsimonious model and simplifying the associated analysis and discussion. Second, two new variables were used in some analyses: Facilitators and Time. These variables are included in some models predicting extent of implementation in order to support an analysis of the incremental effect of such variables over and above the generic SPI predictors. (These variables were not included in the prior analysis of Assimilation Stage in Chapter 3 because they presume acquisition has already occurred, and therefore would have necessitated dropping all cases in earlier stages of assimilation.) The Facilitators variable was operationalized as the total number of facilitators employed on at least one production development project. Candidate facilitators included: external OOP experts, professional OOP training classes, project staff with previous OOP experience, OO seminars and conferences. Time was operationalized as the number of years since acquisition of the innovation. Table 5.1.1 provides a synopsis of all predictor variables.

Table 5.1.1: Predictor Variables

Variable	Description
1. Learning-Related Scale (LRS)	Scale of activities over which learning costs can be spread (applications development volume)
2. Related Knowledge	Diversity of application development knowledge and activities (programming languages, runtime platforms, applications architectures)
3. Diversity	Extent of staff knowledge in domains related to object-oriented programming (C programming, client-server development, PC/workstation-based development, GUI development)
4. IT Size	Size of the IT function throughout the host organization (staff and spending)
5. Host Size	Size of the host organization (total employment)
6. Education	Level of education of the IT staff at the site (bachelors and masters)
7. Environmental Complexity	Extent of environmental complexity as evidenced by importance of seven typical objectives for applications development (rapidity, cost effectiveness, schedule and budget compliance, high performance, high reliability, ease of use, ease of change)
8. Facilitators	Number of techniques/strategies used to facilitate deployment of OOPLs (external OOP experts, professional OOP training classes, project staff with previous OOP experience, OO seminars and conferences)
9. Time	Time since adoption (specifically, years since acquisition)

5.2 EMPIRICAL ANALYSIS

This section presents the results of a set of empirical analyses exploring the four issues previously identified. Multiple regression analysis is the primary analytical tool employed, although pairwise correlations are also used to support some analyses. Most of analyses are based on OOPL adoption data, although data on the adoption of RDBs and CASE are used in some models.

Issue 1: Do richer innovativeness measures generally lead to more strongly predictive models?

In the conceptual discussion analysis presented in Chapter 4 it was noted that, of the three general-purpose, single innovation measures, Assimilation Stage was the richest, Dichotomous Adoption the thinnest, and Time of Adoption

somewhere in between. Richer outcome measures are usually more desirable because the implications of results tend to be more interesting or more far-reaching. However, another goal when testing variance models is to strongly confirm hypothesized relationships. Since richer innovativeness measures capture more variation in innovation-related outcomes, it seems reasonable that they should, in general, have stronger relationships with predictor variables. To examine this and related issues, this subsection uses data on the six "generic" SPI predictor variables (Learning-Related Scale, Diversity, etc.), together with Related Knowledge, in regression models predicting Dichotomous Adoption, Time of Adoption, and Assimilation Stage for OOPLs. The operationalizations for the explanatory variables were described earlier. The operationalizations for the innovativeness measures are provided in Table 5.2.1 below.

Table 5.2.1: Operationalizations of OOPL Time of Adoption, Dichotomous Adoption, and Assimilation Stage

Innovativeness measure	Operationalization
Dichotomous Adoption	A binary variable indicating whether acquisition occurred as of the end of the observation period (end of 1993).
Time of Adoption	A binary variable indicating, for each year in the observation period, whether acquisition occurred in that year.
Assimilation Stage	Guttman scale with 7 positions: 0-Not Aware, 1-Aware, 2-Interested, 3-Evaluation/Trial, 4- Commitment, 5-Limited Deployment, 6-General Deployment. Twelve items used to classify.

Table 5.2.2 presents the results of three multiple regression analyses, one for each of the three innovativeness measures. The columns for the predictor variables contain the p-values for significant estimated coefficients (at $p \leq .05$), with blank cells corresponding to non-significant predictors. All of the significant coefficients are positively signed, as expected. The models predicting Dichotomous Adoption (1.1) and Time of Adoption (1.2) were estimated using logit analysis, and the discrete event history analysis

procedure described in [Allison 1982],¹ respectively. The specific estimation procedure for both kinds of analysis is logistic regression, although in the case of event history analysis, the unit of analysis is not the organization, but rather organization-in-year. (This explains the large N for the Time of Adoption model.) Ordinary least squares regression was used to predict Assimilation Stage.² The overall model fit statistics reported for the logistic regressions is the model chi-square statistic. Both the F-statistic and the adjusted R² are reported for the OLS regression. (These same conventions are used on all tables appearing in this paper.)

Table 5.2.2: Models Predicting Dichotomous Adoption, Time of Adoption, and Assimilation Stage for OOPLs

Innovativeness Measure	Model	n	Overall Model Fit	LRS	Rel know	Diversity	IT Size	Host Size	Education	Env Cplx
Dichot. Adopt'n	1.1	608	$\chi^2=685$.014	.000	.000	.000			
Time of Adoption	1.2	4019	$\chi^2=1257$.002	.000	.000	.000			
Assimilation Stage	1.3	583	F=36, R ² =.30	.000	.000	.000	.000			

The main pattern evident from a comparison of models 1.1 through 1.3 is that for OOPLs, the outcome measure (and associated modeling technique) matters little in the inferences drawn about significant predictors. The same pattern of significant effects holds regardless of the measure employed, and the p-values for significant coefficients are similar across measures. The three novel explanatory variables (Learning-Related Scale, Related Knowledge and Diversity) are highly significant in all models, as is IT Size. Host Size,

¹A new data set was created containing one case for every organization for every observation year in the study time frame, up until the year of adoption. (For organizations where adoption never occurs in the study time frame, there are as many cases as years in the time frame.) The outcome variable on each case was set to one if adoption occurred for that organization during that year, otherwise it was set to zero. The predictor variables were repeated for all cases for a given organization. A set of dummy variables was added to identify the observation year for each case.

²The number of cases for this model is N=583, rather than 608, because 25 respondents were classified as "rejectors". These organizations, by design, do not belong in any of the assimilation stage categories.

Education and Environmental Complexity are non-significant in all three models.

The expectation that Assimilation Stage and Time of Adoption, because they are richer measures than Dichotomous Adoption, would have a more highly significant coefficient for most predictors, is not supported. However, an alternative concern, that Assimilation Stage might systematically diminish the effects of certain predictors because it combines multiple elements (earliness of initiation, speed of assimilation) is clearly not the case either.

Issue 2: Does Aggregated Adoption suffer from a lack of correspondence with other innovativeness measures?

As described earlier in Chapter 4, Downs and Mohr have raised a concern that Aggregated Adoption might not necessarily be an adequate proxy for the propensity of an organization to adopt innovations early, or to implement innovations in depth. To investigate this issue, this analysis uses the six "generic" explanatory variables in regression models explaining three aggregated innovativeness measures—Number of Adoptions, Average Years Since Adoption, and Average Assimilation Stage. Table 5.2.3 below describes the operationalizations for the three aggregated measures.

Table 5.2.3: Operationalizations of Aggregated Adoption Measures

Innovativeness measure	Variant	Operationalization
Aggregated Adoption	Number of Adoptions	Number of acquisitions for the technology set including OOPLs, RDBs, and CASE.
	Average Years Since Adoption	Average of the square root of years since acquisition for OOPLs, RDBs, and CASE (non-adopters coded to zero).
	Average Stage	Average value of Assimilation Stage for OOPLs, RDBs, and CASE. ³

³For each SPI, assimilation stage was measured using a Guttman scale with 5 positions: 0-No Acquisition, 1-Acquisition, 2-Commitment, 3-Limited Deployment, 4-General Deployment. The scale used here has fewer categories than one used previously for OOPLs because less extensive adoption data were available for RDBs and CASE. To be consistent, OOPL stage was recoded to this more abbreviated scale for the analyses in this subsection.

As a first step, correlations were computed for these three measures. The resulting correlations ranged from $r=.84$ to $r=.88$, which strongly suggests that, for this data set, all three measures are tapping into the same construct.

Nevertheless, Downs and Mohr's contention might still hold to some extent, albeit not enough to show up in the above correlations. To assess whether this might be the case, a secondary analysis was performed. Two classes of organizations were defined with respect to OOPLs: those that had acquired OOPLs but not RDBs or CASE, versus those that had acquired OOPLs as well as RDBs and CASE. Then the average value for OOPL Assimilation Stage and the average value for Years Since OOPL Acquisition were computed for both groups. If Downs and Mohr's contention is true, then the group of organizations acquiring all three SPIs should have a lower average value for OOPL Assimilation Stage, and a lower average value for Years Since OOPL Acquisition than the group of organizations only acquiring OOPLs. An analogous procedure was performed for RDBs and for CASE.

However, as shown in Table 5.2.4 below, the average values for Assimilation Stage and Years Since Acquisition for any particular SPI are generally higher, not lower, for the set of organizations that have acquired all three innovations. This suggests that for these software process innovations, at any rate, there is little support for Downs and Mohr's contention on this issue.

Table 5.2.4: Average Assimilation Stage and Years Since Acquisition for Sole Acquirers vs. Triple Acquirers of SPIs

Number of Acquisitions	Average OOPL Stage	Average OOPL Years	Average RDB Stage	Average RDB Years	Average CASE Stage	Average CASE Years
Only acquired one SPI	1.29	1.69	2.92	4.76	1.78	3.86
Acquired all 3 SPIs	1.55	1.90	3.27	5.29	2.08	3.38

The above results imply that for most analyses, these three alternative measures of Aggregated Adoption should be nearly interchangeable. To confirm that this is indeed the case, the three measures were employed as

dependent variables in regression models using the six "generic" predictors (see Table 5.2.5 below). As before, p-values are reported for significant coefficients (all are positive), and cells for non-significant coefficients are left blank.

The estimated models are very similar in terms of variance explained. In addition, only minor variations exist in the pattern of significant variables: Host Size is (barely) significant for model 2.1 but not for 2.2 or 2.3; Environmental Complexity is significant for models 2.2 and 2.3 but not 2.1. As a result, any systematic differences that might exist between the propensity to acquire many innovations, and the propensity to acquire innovations early or to sustain implementation, does not appear to be strong enough to have a profound effect on model predictions. In all cases, the three variables hypothesized in Chapter 3 to strongly influence SPI assimilation are highly significant.

Table 5.2.5: Models Predicting Aggregated SPI Adoption

Innovativeness Measure	Model	N	Overall Model Fit	LRS	Diversity	IT Size	Host Size	Education	Env Cplx
Avg Yrs Since Adoption	2.1	608	F=78, R ² =.43	.000	.000	.000	.046	.002	
Number of Adoptions	2.2	608	F=83, R ² =.45	.000	.000	.000		.003	.002
Average Assim Stage	2.3	608	F=92, R ² =.47	.000	.000	.000		.001	.000

Issue 3: Can the results achieved with Aggregated Adoption be confidently generalized to particular innovations in the set?

Another issue raised by Downs and Mohr is the concern that results achieved when analyzing Aggregated Adoption cannot be confidently generalized to any particular innovation in the set. To investigate this issue, four regression models were estimated (See Table 5.2.6 below). The first model (3.1) employs average Assimilation Stage. (This is the same as Model 2.3, repeated here for convenience.) The other three (3.2 - 3.4) are single innovation models, using Assimilation Stage as the outcome measure. As before, p-values are reported

for significant coefficients (all are positive except Host Size regressed on OOPL Assimilation Stage), and cells for non-significant coefficients are left blank.

Table 5.2.6: Models Predicting SPI Assimilation Stage

Innovativeness Measure	Model	N	Overall Model Fit	LRS	Diversity	IT Size	Host Size	Education	Env Cplx
Average Assimilation Stage	3.1	608	F=92, R ² =.47	.000	.000	.000		.001	.000
RDB Assimilation Stage	3.2	608	F=42, R ² =.29	.000	.002	.012	.008	.034	.000
OOPL Assimilation Stage	3.3	608	F=19, R ² =.15	.002	.000	.001	(.034)	.031	.034
CASE Assimilation Stage	3.4	608	F=37, R ² =.26	.000		.000			

A comparison of these models shows that the aggregated model comes closer to the expected pattern of significant coefficients than two out of three of the single innovation models. While the strongest predictors—Learning-Related Scale, Diversity, and IT Size are consistently significant (except for the case of Diversity regressed on CASE)—the weaker predictors are not. One possible explanation is that these differences are due to variations in the level of overall diffusion achieved by the end of 1993 for the SPIs. The percentage of acquirers of RDB, OOPLs and CASE as of 1993 were 67%, 44% and 30%, respectively, and this same ordering holds for the strength of the associated models in terms of number of predictors found to be significant in the expected direction (six, five, and two, respectively). It may be that, especially for CASE, there is simply not enough variation in the dependent variable for weaker effects to be detected (70% of cases have the floor value of zero for CASE Assimilation Stage). Alternatively, these differences could be due to variations in the extent to which the different SPIs conform to the attributes of the "typical" SPI for which the model was developed.

Another result worth noting is the striking effect of aggregation on explained variance. The explained variance is R²=.47 for Average Assimilation Stage, as compared with a maximum of R²=.29 for any of the single SPI Assimilation

Stage model. An interesting question is whether these results are due specifically to the effects of using an aggregated measure, over and above the effect of incorporating multiple innovations. A secondary analysis showed that it is the incorporation of multiple innovations rather than aggregation *per se* that accounts for the pattern of significant relationships. This analysis was performed using the adoption-decision design, with Assimilation Stage as the outcome variable. With this design, rather than aggregating multiple innovation outcomes, a separate case is constructed for each innovation for each organization. A regression analysis produced the same pattern of significant coefficients as for average stage, and the p-values were nearly identical.

The above results related to Issues 2 and 3 suggest that—contrary to the position of Downs and Mohr—Aggregated Adoption can be a valid and useful innovativeness measure, at least when the research only seeks to examine more "generic" innovation predictors, and when the prescriptions suggested earlier are followed.⁴ The selection of which measure to aggregate had little effect, and the mere act of aggregation did not change the results over and above the inclusion of multiple innovations. Nevertheless, one warning issued by Downs and Mohr may hold: the results of an aggregated model can not necessarily be extended to any particular innovation, either in terms of significant effects, or overall strength of associations. What this means is that researchers who wish to draw strong conclusions about individual innovations in the set should be sure to capture the data necessary for a robust analysis at the disaggregated level.

Issue 4: Are the extent of implementation measures (Diffusion and Infusion) as strongly predictive as the other four measures?

⁴As prescribed in Chapter 4, this analysis used a homogeneous set of innovations, included organizational characteristics that fell towards the primary end of the primary-secondary continuum, compared models with both aggregated and disaggregated measures, and used alternative measures as the basis of aggregation.

One of the points made in the conceptual analysis in Chapter 4 was that the extent of implementation measures are likely to be especially difficult to predict, because analysis is necessarily confined to the most innovative organizations, those who have already begun deploying the technology. To explore this issue, several regression analyses were performed using OOPL Diffusion and Infusion as the outcome variable. The operationalizations for these measures are supplied in Table 5.2.7 below.

Table 5.2.7: Operationalizations of OOPL Diffusion and Infusion

Measure	Operationalization
Diffusion	Percentage of new development projects initiated in 1993 on which an OOPL was used.
Infusion	Summative scale with 3 dimensions: 1) Number of supporting object technologies used (object databases, object CASE tools, object methodology), 2) Number of class library types employed (GUI support, basic data structures, database access, industry specific), 3) Number of application components for which OOP is heavily used (presentation/user interface, application logic/business rules, database access/data management).

The correlation between Diffusion and Infusion is $r=.38$. The correlations of Diffusion and Infusion with time of acquisition are $r=.38$ and $r=.30$, respectively.

The results for six regression analyses (models 6.1 through 6.6) are provided in Table 5.2.8 below. As before, p-values are reported for significant coefficients (all are positive), and cells for non-significant coefficients are left blank. The general predictiveness is quite weak for all models, with F-statistics ranging from $F=1.4$ to 4.0 . The only consistent predictor is Related Knowledge. The results for these models do emphasize the value of including innovation specific and stage specific predictors, and controlling for time, when using extent of implementation as an outcome measure. Including Related Knowledge (innovation specific) and Facilitators (stage specific) increases the variance explained from $R^2=.03$ to $.21$ for Diffusion, and

$R^2=.10$ to $.25$ for Infusion. Including Time further boosts the variance explained to $R^2=.30$ and $.28$ for Diffusion and Infusion, respectively.

Table 5.2.8: Models Predicting OOPL Diffusion and Infusion

Measure	Model	N	Overall Model Fit	"Generic" Predictors						Innovation or Stage Specific Predictors		
				LRS	Diversity	IT Size	Host Size	Education	Env Cplx	Rel Know	Facilitate	Time
Diffusion	6.1	63	F=1.4, R ² =.03									
Diffusion	6.2	63	F=3.1, R ² =.21							.000		
Diffusion	6.3	63	F=4.0, R ² =.30					.039		.002		.006
Infusion	6.4	63	F=2.1, R ² =.10			.032						
Infusion	6.5	63	F=3.7, R ² =.25							.010	.012	
Infusion	6.6	63	F=3.6, R ² =.28						.044	.027	.019	

There are two classes of potential explanations for the general weakness of these models. The first is that differences among the roughly 10% of the population that has begun to deploy an OOPL are likely to be inherently difficult to explain, due to: 1) decreased statistical power, 2) decreased variation in the predictor variables, 3) decreased variation in substantively meaningful innovativeness in the responding set, and 4) potential diminishing returns for higher levels of predictor variables. Regarding statistical power, decreasing the N for analysis by 90% obviously greatly increases the probability of a type II error. Regarding decreased variation in values for variables, one property of regression analysis is that it is more difficult to separate genuine effects from various sources of noise (e.g., measurement error, omitted variables, random variation) when the substantive level of variation on either side of the regression equation is less. Finally, if there are diminishing returns above certain levels of the predictor variables, and many organizations in the set are at or above those levels, then

there may in fact be little in the way of incremental effects to detect. Each of these potential problems can be lessened, but not eliminated, by waiting for a large percentage of organizations to reach the later stages of deployment. However, this approach effectively expands the observation time frame, and potentially precludes the use of a cross-sectional approach to data collection. Furthermore, for some innovations, it may be that only a subset of organizations are destined to ever reach later innovation stages, in which case analysis would be necessarily confined to a special population regardless of the length of the observation period.

A second class of explanation is that "generic" SPI innovativeness predictor variables might, in fact, have little additional effects during deployment. That is, their influence may be felt only in getting adopters to the point of deployment. If this turned out to be the sole explanation, then waiting for more organizations to deploy would not improve matters. A careful longitudinal study would be needed to ascertain the precise reason for the general absence of significant effects for "generic" predictors on extent of implementation.

5.3 CONCLUSIONS

The above empirical analysis provides a valuable complement to the conceptual analysis presented earlier in Chapter 4. Among the more important empirical findings are: 1) richer measures do not generally lead to more strongly predictive models; 2) there are a positive relationships between the propensity to acquire many innovations, to acquire any one of those innovations relatively early, and to implement any one of those innovations in relatively greater depth; 3) Aggregated Adoption has a strong positive effect on variance explained, and leads to a pattern of significant relationships very close to that predicted by theory; and 4) the extent of implementation measures (Diffusion and Infusion) are very difficult to predict, at least when the pool of prior adopters is small. The analysis also confirms the value of

gathering secondary innovativeness measures to support an analysis of the generalizability of results.

The major implication of these results on the whole is positive: aside from Diffusion and Infusion—which may well be distinct dimensions of innovativeness, best predicted by variables not included here—the alternative innovativeness measures do appear to be tapping into a single notion of general innovativeness. Or, in other words, these results suggest that separate models of general innovativeness may not be required for most innovativeness measures. This does not mean the selection of a measure is trivial—quite the contrary. Because different measures open the door to different classes of predictors, and differ in terms of valid analytical techniques, the selection of the most appropriate innovativeness measure is still one of the most important design decisions a researcher must make. Aggregated Adoption, for example, while apparently very sensitive in identifying "generic" predictors, rules out the opportunity to include other interesting classes of predictors, such as those with time-varying values, or those that have innovation specific values. What these results do mean is that researchers for the most part have the freedom to let their substantive interest in different classes of predictors drive the research design process, while still maintaining a strong link to the cumulative tradition of innovation research.

CHAPTER 6. THE ILLUSORY DIFFUSION OF INNOVATION: AN EXPLORATORY EXAMINATION OF ASSIMILATION GAPS

6.1 INTRODUCTION

The research reported in this chapter is motivated by the following basic idea: widespread acquisition of an innovation need not be followed by widespread deployment and use by acquiring organizations. While the implications of this insight have been incorporated into some of the more recent studies focusing on the antecedents of organizational innovation [Meyer and Goes 1988; Cooper and Zmud 1990], there has not been a comparable recognition of this insight by researchers engaged in diffusion modeling studies, even though the implications for this second major style of diffusion research are equally profound.

Diffusion modeling studies are concerned with explaining and predicting the patterns innovations follow as they spread across a population of potential adopters. A typical approach is to define adoption as the physical acquisition or purchase of the innovation, and then to fit a times series of observed cumulative adoption counts or percentages to some functional form, such as the logistic [Mahajan and Peterson 1985]. When it can be safely assumed that later events in the organizational assimilation process will nearly always follow quickly on the heels of earlier events, then the observed pattern of cumulative adoptions will not vary much depending on the particular assimilation event used to define the time of adoption. The diffusion pattern that results when an earlier event (e.g., acquisition) is used to define adoption will closely mirror the pattern that results when a later event is used (e.g., non-trivial deployment), as illustrated in Figure 6.1.1.

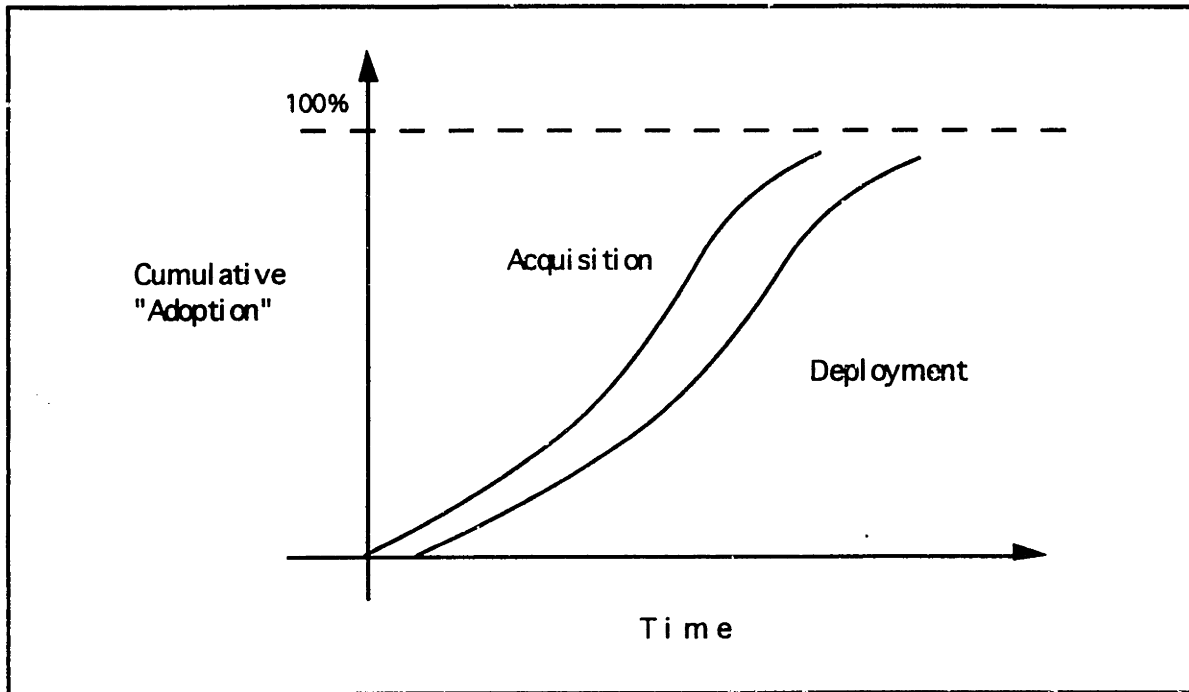


Figure 6.1.1 Diffusion Curves for Alternative Adoption Events

However, for some technologies, it may be unrealistic to assume that in most organizations later assimilation events will automatically follow earlier events. In Chapter 1, it was argued that technologies subject to knowledge barriers might be particularly prone to violating this assumption. For such technologies, successful assimilation typically requires an arduous process of organizational learning, and it can therefore be expected that many acquiring organizations will experience unusual deployment delays, while still others may never thoroughly deploy the technology all. As a result, the pattern of cumulative deployments need not closely mirror the pattern of cumulative acquisitions, but rather there may be a widening "gap" between these two curves plotted as a function of time (See Figure 6.1.2 below). Because this gap is bounded by the cumulative adoption curves associated with two alternative assimilation events, it is labeled here as an *assimilation gap*.

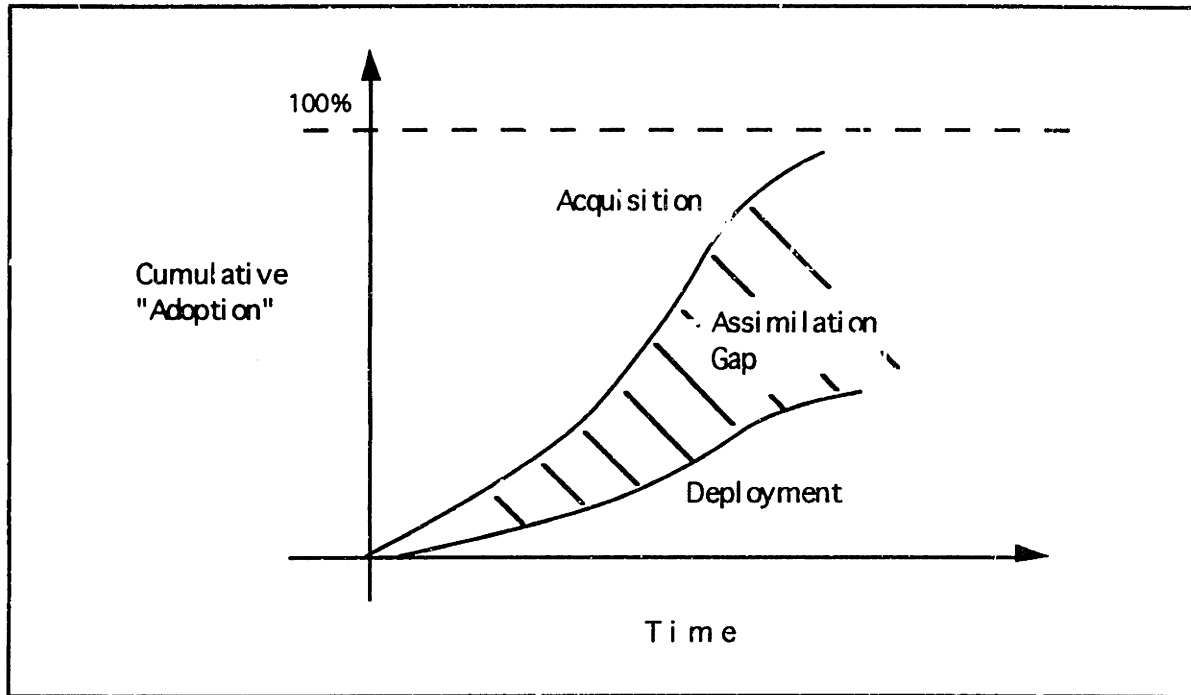


Figure 6.1.2 Assimilation Gap

To the extent that a substantial assimilation gap does exist for an innovation, the use of cumulative acquisition as the basis for diffusion modeling can present an illusory picture of the diffusion process—leading to potentially erroneous judgments by vendors and prospective adopters about the robustness of the diffusion process already observed, and the technology's future prospects. As argued in Chapter 1, a technology that initially has a large assimilation gap is unlikely to achieve critical mass and go on to dominance, because slow or failed deployment among early adopters delays the learning-by-using and other forms of increasing returns needed to make the SPI attractive to a mass market of potential adopters.

For end users, the costs of committing to a technology that never achieves dominance can be substantial, including: difficulties in hiring experienced personnel; limited availability of complementary technologies; limited enhancements to the core technology; limited availability of third party training; a general lack of accumulated industry wisdom about how to best

apply the technology; and possible loss of vendor support [Fichman and Kemerer 1993].

For vendors, making a wrong or even badly timed technology choice can spell disaster. Borland, a high-profile early adopter of object technology suffered a major set back in 1992 and 1993—Borland stock had fallen from a high of \$86 in January of 1992 to around \$20 in August of 1993—owing to serious delays in introducing key software products, delays which Borland attributed to the use of object-oriented techniques [Brandt 1993]. Chief executive Phillip Kahn was confident at the time that the early move to object-oriented techniques would ultimately be vindicated and that Borland would end up ahead of the pack as a result. Eighteen months later, however, this had not yet occurred: Borland stock was hovering around \$7, Kahn had stepped down, and the company's long term future was in serious doubt [Clark 1995].

So while cumulative acquisitions need not be mirrored by cumulative deployments—and therefore modeling only one event can present an incomplete picture of the diffusion process—the implications of this fact have not previously been incorporated into diffusion modeling studies. This research, in contrast, is directly concerned with the issue of how to model the diffusion of technologies that, like SPIs, appear likely to exhibit a pronounced assimilation gap. Specifically, this research develops novel modeling techniques and then applies them in an analysis of adoption data for three software process innovations (SPIs): relational database management systems (RDBs), fourth generation languages (4GLs), and computer aided software engineering tools (CASE).

The remainder of this chapter is organized as follows. It begins with a review of some pertinent prior research (Section 6.2). It then formalizes the assimilation gap concept (Section 6.3), and provides an empirical examination based on survival analysis techniques of the assimilation gaps for RDBs, 4GLs and CASE (Sections 6.4 and 6.5). Finally, Section 6.6 provides a discussion of

the study's limitations and contributions, and draws implications for researchers and practitioners.

6.2 PRIOR RESEARCH

Although this research introduces and develops the assimilation gap concept, the general idea that process innovations can be widely acquired but only sparsely utilized is not new. Eveland and Tornatzky describe the "sad case" of machine vision, a revolutionary process technology introduced to manufacturing in the late 1970's and early 1980's:

"Many major corporations installed machine vision systems . . . Then the trouble began . . . Many plants simply gave up. Some large, and expensive, machine vision systems were 'deinstalled.' Automation consultants, in visits to plants, found unused machine vision systems sitting boxes, relics of failed deployment." [Eveland and Tornatzky 1990, p. 123].

Liker *et al.* report that computer aided design (CAD) technologies had achieved unusually rapid market penetration in the 1980s, yet, as late as 1992 "true CAD/CAM [utilization was] still quite rare" [1992, p. 75].

Cooper and Zmud [1990], in a study of material requirements planning (MRP), report that 63% of surveyed companies had not progressed beyond Class C MRP implementation, a relatively low level of utilization that, among other things, does not support capacity planning.

Why has the idea of modeling assimilation gaps or using multiple definitions of the adoption event apparently never been previously incorporated into diffusion modeling studies? Several possibilities come to mind. First, comparatively few diffusion modeling studies have taken complex organizational technologies as their focus, so the implicit assumption that deployment nearly always follows acquisition might be reasonable.¹

¹Some notable example of recent studies that did take complex organizational technologies as their focus include Mahajan *et al.* [1988] and Mansfield [1993].

Second, for many studies, whether or not an assimilation gap exists may not be particularly relevant to the purposes of the research. For a study concerned with modeling the spread of home video cassette recorders (VCR), for example, the fact that many, if not most, early adopters used only the most basic play-back features may not matter to researchers. VCRs may have still be well worth the purchase price to most buyers even when underutilized, and the fact of widespread under utilization appears to have had little effect on the diffusion path of VCRs. Complex organizational technologies, by contrast, are of little value to an acquiring organization if not widely deployed, and sparse deployment among early adopters is likely to have a considerable effect on the subsequent diffusion path of the technology.

Third, the use of multiple adoption events introduces methodological difficulties related to data collection and analysis. Traditional sources of industry data on adoptions, such as data on first-purchase sales, only provide data based on a single adoption event: the technology purchase. In addition, the historical practice among innovation researchers, has been to collect data based one definition of adoption. As a result, secondary sources of adoption data from industry or prior innovation studies—a valuable resource frequently employed by diffusion researchers—rarely, if ever, contain data on multiple definitions of the adoption event. So researchers who wish to model multiple events are required to collect primary data, an expensive and time consuming process. With regard to data analysis, here again, since the historical practice has been to use data based on a single definition of the adoption event, traditional analytical techniques do not provide tools to model and compare diffusion curves according to multiple definitions of the adoption event. In fact, one of the primary contributions of this research is to develop such tools.

Fourth, and perhaps most importantly, the notion of an assimilation gap is an alien concept to classical diffusion theory. Rogers explains that when an innovation has been observed to diffuse rapidly and widely, that means the

innovation probably has favorable characteristics, i.e., high relative advantage, low complexity, high compatibility [1983]. Conversely, innovations that diffuse slowly and narrowly probably have unfavorable characteristics. What, then, is a diffusion researcher operating within the confines of classical theory to make of an innovation observed to have diffused rapidly according to one definition of the adoption event, but slowly according to another? An even more basic question not considered in classical diffusion theory is how such a pattern can occur at all, since it appears to imply a systematic error in managerial perceptions about an innovation. In Chapter 1, the potential for sizable assimilation gaps in the case of SPIs was deduced from two distinctive characteristics of these technologies. The argument, stated briefly, is that the expectations of future benefits, owing to an assumption of widespread future adoption and attendant increasing returns, drives rapid acquisition, but knowledge barriers associated with deploying these technologies systematically impede implementation among acquiring firms. It is worth noting that this argument turns on the existence of increasing returns and knowledge barriers, both of which are at odds with the implicit assumptions of classical diffusion.

6.2 THE ASSIMILATION GAP CONCEPT

An assimilation gap is defined here as *the difference between the pattern of cumulative acquisitions and deployments of an innovation across a population of potential adopters*. Although this definition is made in reference to two particular events—acquisition and the point of non-trivial deployment—in principle, any two assimilation events could be used to define an assimilation gap, if so warranted by the objectives of the research.

The central operational measure of the assimilation gap concept employed in this study is *the area between the cumulative acquisition and deployment curves at time T as a proportion of the area under the cumulative acquisition*

curve at time T . As illustrated in Figure 6.2.1 below, this measure can be computed as the cross hatched area divided by the area under the cumulative acquisition curve.

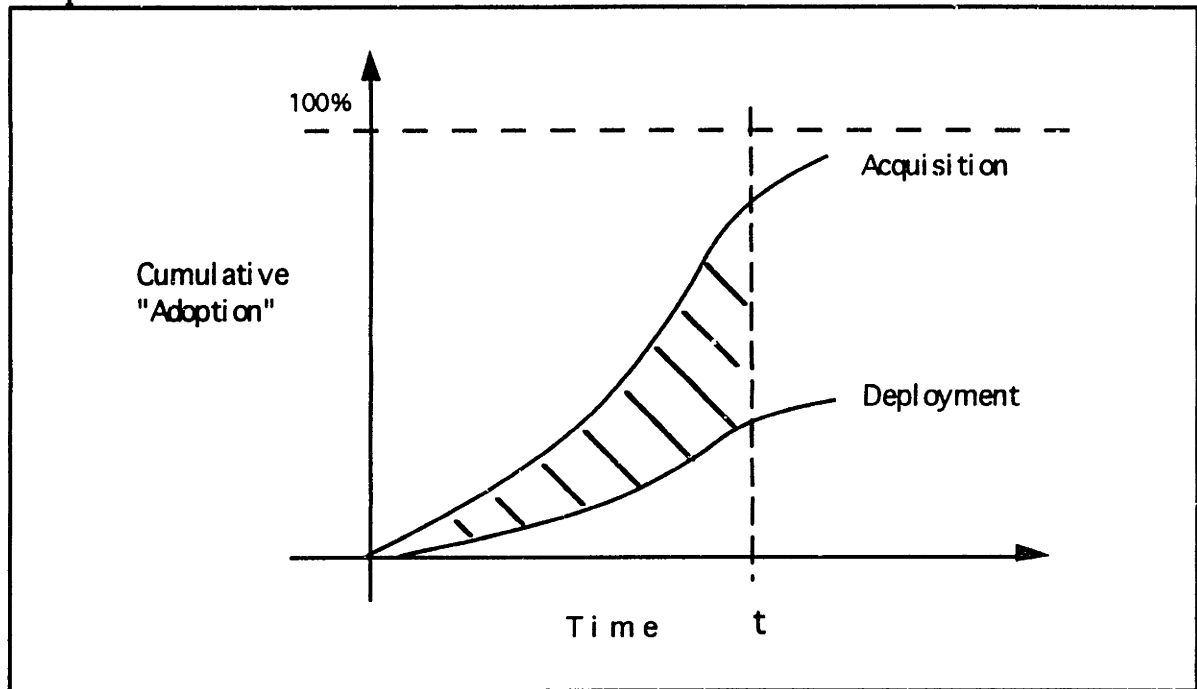


Figure 6.2.1 Computing the Assimilation Gap

While assimilation gaps are a population level concept, it nevertheless seems appropriate to consider what kinds of specific adopter-level outcomes contribute to it. First, and perhaps most benignly, are natural lags in the deployment process. If this kind of delay were the only kind of outcome that commonly occurred then assimilation gaps would not be particularly interesting from a theoretical standpoint. Yet, there are several other, less benign outcomes that potentially contribute to an assimilation gap. Some adopters might experience "excessive" deployment delays beyond what would ordinarily be expected. Such adopters eventually deploy, yet make a significant contribution to the assimilation gap in the mean time. Other adopters may fail to implement, either because implementation is never attempted (rejection), or because implementation is attempted, but no actual use of the innovation beyond experimentation is ever achieved

(implementation failure). Or, adopters may indeed begin implementation of the innovation, but then "stall" at a level of ongoing utilization too low to qualify as deployment. Finally, some adopters may achieve some level of ongoing use of the technology short of the level that qualifies as deployment, but then discontinue use. Organizations in these last four categories—rejection, implementation failure, stalling, and discontinuance—make a permanent contribution to the assimilation gap.

Which of these different organization-level outcomes are most prevalent for a particular technology is important to consider both practically and theoretically, and later sections in this chapter will present a partitioning of organization-level outcomes along these lines. Nevertheless, the primary focus here is on modeling assimilation gaps as a whole. This is viewed as a valuable and necessary first step in the general study of assimilation gaps

6.3 AN EMPIRICAL STUDY OF ASSIMILATION GAPS FOR RDBs, 4GLs AND CASE

In Chapter 1, it was proposed that SPIs are likely to be especially prone to an assimilation gap, owing to their combination of increasing returns and knowledge barriers.² This section presents an examination of assimilation gaps for three of the more prominent SPIs to emerge in the 1980s: relational database management systems (RDBs), production fourth generation languages (4GLs) and "upper" CASE tools—tools that support systems analysis and design (CASE). RDBs, aside from being simpler to understand and use than those based on prior data base models (e. g., network and hierarchical), were acclaimed for providing a quantum leap in the level of support for data independence.³ 4GLs, which provide very high level

²It is interesting to note that machine vision, CAD/CAM, and MRP, introduced earlier as examples of technologies that appear to have experienced large assimilation gaps, also arguably share this interesting combination of increasing returns and knowledge barriers.

³Data independence is the ability to make changes to the underlying structure data without changing programs that access the data. Data independence is one of the single most important reasons for the original development of database management systems.

elements to automate construction of the more common components of information systems (e.g., data entry screens and reports), were expected to quickly replace Cobol as the primary language for development in mainstream MIS groups. The third SPI, CASE tools, emerged in the mid 1980's, about a half decade later than RDBs and 4GLs. These tools were expected to do for the process of systems analysis and design what CAD did for the design of manufactured components—namely provide automated support not only for visual representation of components, but for related analytical tasks as well, such as ensuring that designs conform to structural rules.

These three SPIs were selected to support the analysis of assimilation gaps because of their prominence, and also because of the expectation that there would be contrasts among them in the size of measured gaps. This expectation was driven by both theoretical considerations and prior empirical results. Because all three SPIs enjoyed extensive, favorable "signalling" when first introduced, it was expected that rapid rates of acquisition would be observed for all three. Yet, while all three SPIs impose a knowledge burden, and hence, would be expected to display at least a moderate assimilation gap, CASE tools arguably impose a far greater knowledge burden, and should therefore have a more pronounced gap. The assimilation of CASE tools requires organizational learning on a number of separate fronts, including: learning about the underlying methodologies the CASE tools automate, which are, themselves, are quite complex [Fichman and Kemerer 1993]; learning about the CASE philosophy, which advocates an enterprise model based approach to systems development [Stone 1993]; learning how to use the actual software products that provide CASE functionality; and finally, learning how to structure project teams and incentives in light of the radically different approach to development often accompanies CASE [Orlikowski 1991]. RDBs and 4GLs, by contrast, while not absolutely simple to learn by any means, have fewer and simpler abstract elements than CASE,

and have a more limited impact on the overall development process [Fichman and Kemerer 1993].

Reinforcing these theoretical reasons to expect an especially pronounced assimilation gap for CASE are both anecdotal reports of low utilization of acquired CASE tools [Kemerer 1992] and the recent finding by Howard and Rai that, of organizations having acquired any CASE tools, only a negligible number had "substantially replaced conventional systems development techniques with CASE" [1993, p. 66]. In sum, RDBs and 4GLs were expected to display a moderate assimilation gap, while CASE was expected to display a much larger one.

6.3.1 Estimating the Assimilation Gap

The simplest approach to estimating assimilation gaps is to plot cumulative acquisitions and deployments as a function of time, and to then extract quantities necessary for assimilation gap computations directly from the resulting graphs. This approach is limited, however, in that it provides no basis for assessing the sensitivity of results to sampling variation, and can not be used to support desirable statistical inferences, such as that an assimilation gap is statistically significantly itself, or that it is significantly larger (or smaller) than some other gap. Fortunately, these limitations can be remedied by supporting the graphical approach with survival analysis techniques, as described below.

Survival Analysis

Survival analysis refers to a family of statistical techniques that support the analysis of lifetime data, that is, data on the elapsed time until the occurrence of some event of interest [Lawless 1982; Cox and Oakes 1984; Singer and Willett 1991].⁴ Survival analysis techniques were originally developed by

⁴The description of survival analysis provided here is based largely on the one provided by Singer and Willett [1991], and uses the same terminology.

biostatisticians studying human lifetimes (hence the label, "survival analysis"), although more recently, these techniques have received growing attention by social scientists for use in studies that share the central problem of analyzing lifetime data, namely, the censoring of observations. Because not all experimental units are likely to have experienced an event of interest (e.g., job turnover) during the study's observation period, the lifetimes for some units are viewed as being censored. With traditional statistical techniques, censoring of data leads to unacceptable biases in the estimation of parameters; extreme sensitivity to the length of the data collection period is also common. Survival analysis, by contrast, makes appropriate use of all observations, even censored ones, and can be used to estimate several functions of interest, including the survivor function, the hazard rate function, and the probability density function. Of these, the survivor function is most relevant for the current application.

The population survivor function is defined as the probability that a randomly selected experimental unit from the theoretical population will *not* have experienced the event of interest by elapsed time T . Alternatively, the survivor function can be viewed as showing the proportion of the population that will have lifetimes exceeding T . The survivor function begins at or near one at the beginning of the study observation period, and then monotonically decreases over time. The *sample* survivor function is estimated based on the observed lifetimes from a particular sample; if this sample is representative, then the sample survivor function at time T provides an estimate of the probability that a randomly selected unit from the original population will have a lifetime greater than T .

For the current application—the study of diffusion processes—the lifetimes of interest are the times until the adoption event (acquisition or deployment). The survivor function for a diffusion process is essentially the mirror image, rotated around an X axis, of the cumulative adoption function. As an

example, the cumulative adoption function and survivor function for RDB acquisitions are illustrated together in Figure 6.3.1.1 below.

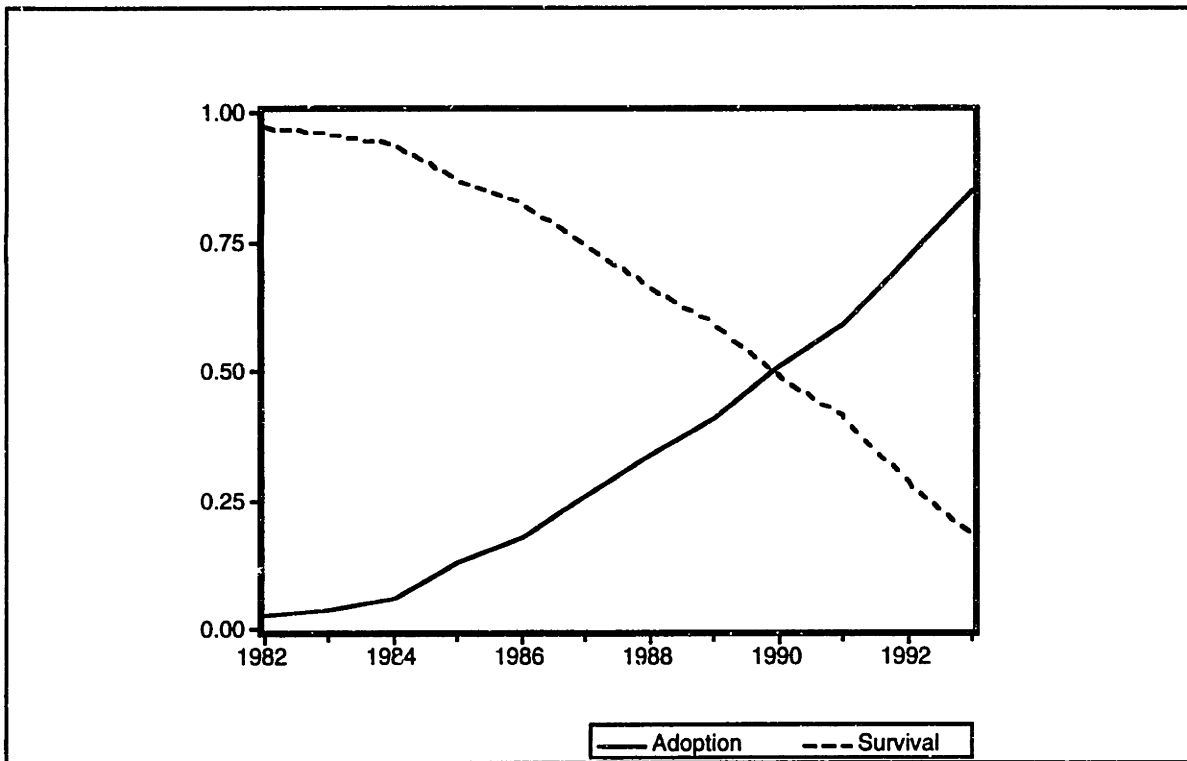


Figure 6.3.1.1: Cumulative Adoption and Survivor Function for RDB Acquisitions

Depending on the specific estimation technique used, the estimated survivor function will be very close or identical to the one's complement of the observed cumulative adoption function. As a result, survival analysis can be used to estimate assimilation gaps just as with the graphical approach (after appropriate transformations), but with an added level of refinement: survival analysis techniques produce estimates of the standard errors for survival probabilities at time T. This, in turn, provides a basis to assess the level of confidence one should place in survival probability estimates, and indirectly, assimilation gap estimates.

When survival analysis is used as described above, the survival process for a potential adopter is bounded by the start date for the diffusion process and the adoption event. It is possible, however, to use the data on times of acquisitions and deployments to model a different kind of survival process, and in so doing, to open up additional kinds of analysis related to assimilation gaps. Rather than modeling acquisitions and deployments as if they were independent events, and then comparing the resulting survivor functions, it is possible to model, just for the acquiring units, the lifetimes extending from acquisition to deployment. When used in this way, the survival clock begins for each case when an SPI is acquired, rather than at the start date for the overall diffusion process.

Although this second approach cannot be used to directly estimate the size of the assimilation gap, at least as defined earlier, it does support several other analyses of particular interest. For example, it can be used to estimate the proportion of the acquiring population experiencing more than "ordinary lags," and the proportion that are likely to never deploy. In addition, this approach can be used to support statistical tests of whether one assimilation gap is larger than another. Specifically, tests can be performed to determine whether, for example, the survivor function (for time until deployment since acquisition) for RDBs, is significantly different from the one for 4GLs. This can be used as a test of whether the assimilation gap for RDBs is different from the one for 4GLs.

Diffusion Modeling

At first glance, traditional diffusion modeling techniques would seem to be a promising approach for estimating assimilation gaps. Diffusion modeling begins with an assumption that the cumulative pattern of adoptions follows some particular functional form, such as the logistic or exponential. Statistical estimation procedures are then used to fit a curve to the observed pattern of cumulative adoptions, and to estimate model parameters, such as

the diffusion rate. The estimated rate of diffusion for adoption-as-acquisition versus adoption-as-deployment could then be used in assimilation gap computations.

Unfortunately, diffusion modeling is not a viable approach for this particular study, for two reasons. First, it has been suggested previously that to produce stable estimates of diffusion parameters, one should have data for at least 10 time periods, including the point where the adoption rate begins to decrease, i.e., the inflection point in the S-curve [Srinivasan and Mason 1986]. However, not enough data are available to meet these criteria for the cumulative deployment series for the SPIs in this study. While this problem might be solved by waiting for more data, a second limitation of diffusion modeling, having to do with traditional assumptions about the ultimate ceiling on adoption, would not. A common practice is to assume all innovations in a given study will eventually be adopted by all organizations, and to use 100% as the adoption ceiling for estimating purposes [Trajtenberg and Yitzhaki 1989].⁵ When assimilation gaps are present, however, this assumption is untenable, because there is a likelihood that adoption-as-deployment will never be achieved by a large segment of the acquiring population (much less the raw potential adopter population). While it is easy to incorporate an assumed adoption ceiling of less than 100% into the modeling process, this does not solve the problem, because then the diffusion rate parameter, which measures the rate of diffusion only through the adoption ceiling, becomes almost meaningless as a speed measure for this application.⁶ These concerns are not merely abstractions: a preliminary

⁵When it is true that at least a large majority of the population will adopt, then estimating the diffusion speed based on the assumption of 100% adoption introduces little distortion. Mansfield found, for example, that introducing an assumption of 70% rather than 100% penetration had only a negligible effect on estimates of the speed parameter in his study of flexible manufacturing [1993].

⁶A simple example illustrates the point: if one innovation takes 5 years to reach an adoption ceiling of one-quarter of a population, should this be viewed as a faster or slower diffusion rate than a second innovation that takes 7 years to reach an adoption ceiling of three-quarters of the same population? To produce sensible assimilation gap results, the second innovation must be viewed as having diffused much faster, although, assuming similar shaped curves, traditional diffusion modeling procedures would estimate a faster rate for the first innovation.

analysis was performed using diffusion modeling, but resulted in uninterpretable estimates of assimilation gaps (See Appendix 6A).

6.3.2 Study Methods

In modeling the cumulative adoption of innovation for some population over time, three measurement-related decisions are required: 1) how to define the population, 2) how to define the time of the adoption event for each member of the population, and 3) how to define the total time period to be covered.

Study Population

The population for this study is defined as those respondents to the disk-based survey (described previously in Chapter 3) who report at least five full time application developers on staff. The requirement of at least five developers is intended to confine analysis to organizations that have the minimum application development scale needed to be *prima facie* strong candidates for adoption of the SPIs under study. Eliminating cases with fewer than five developers reduced the sample size from 608 to 395; elimination of eleven additional unusable cases further reduced sample size to 384.⁷ Among the retained 384 cases, the levels of acquisition for RDBs, 4GLs and CASE were 85%, 64% and 43% respectively, as of 1993; this compares with levels of 67%, 51% and 29%, respectively, for the whole 608 case sample.⁸ Compared with eliminated cases, retained cases were more than twice as likely than to be acquirers for each of the three SPIs.

⁷Six cases were dropped because reported deployment year was after the reported acquisition year for one of the three SPIs. Five more were dropped because reported acquisition dates were too early to be credible. Four of these cases were RDB acquisitions, and had reported acquisition years of 1970, 1973, 1975, 1976, respectively. The fifth was a reported 4GL acquisition year of 1975. There were no commercially available RDBs or 4GLs in these years, and while these organizations may have been using research prototypes, it is more likely that these reported acquisition dates are simply inaccurate.

⁸All data used for analysis and reported in this chapter are unweighted. The use of weights proved to have a negligible effect on reported levels of SPI acquisition (less than 1% for all three SPIs), and hence, was viewed as an unnecessary complication.

Definition of the Adoption Event

Two alternative definitions of the adoption event are employed: adoption-as-acquisition and adoption-as-deployment. The time of adoption-as-acquisition is defined as the reported year that a particular SPI was first purchased by the site. The time of adoption-as-deployment is defined as the year that use of the SPI first reached 25% of all new development projects.⁹ The first measure is intended to parallel a definition of adoption frequently used in diffusion studies, including all of those based on first-purchase data (see, for example, [Bass 1969]). The second measure is intended to capture whether the site has gotten beyond a limited level of deployment.¹⁰ The resulting time series for the three SPIs are the cumulative number of adoption-as-acquisitions and adoption-as-deployments by year.

Observation Time Periods

The time periods to be covered by the analysis were defined in order to achieve a consistent base of comparison for the three SPIs. The end year was defined to be 1993 for all three SPIs, as this was the last full year for which data were available. To establish the start year for each series, a two step procedure was followed. First, the year in which cumulative acquisitions first reached at least 10% of the total population was calculated for each SPI, and assigned to be the "anchor year" for that SPI.¹¹ The start year for each series was then assigned to be three years prior to the anchor year. This resulted in start years

⁹Exhibits 11 and 12 at the end of Chapter 3 contain the questions used to measure these variables for RDBs, along with several other variables of secondary interest. Parallel questions were used for all three SPIs.

¹⁰The 25% level of use is considered to be reasonable cut off for these SPIs. RDBs were promoted as replacement for prior DBMS models, and there are few business applications for which, in principle, DBMSs are not the most appropriate file access method. Production 4GLs were promoted as a replacement for Cobol, a language that had accounted for 90% of the installed base for business applications when 4GLs were introduced. CASE, which was designed to support a philosophy of model based development, should, arguably, be used on every new development project, regardless of implementation environment, lest an incomplete model result.

¹¹This represents a more useful method for anchoring a series, than, for example, taking the first year in which any acquisitions occurred. This is because the start year under such a rule could be strongly subject to outliers and/or chance variation.

of 1982, 1981 and 1986 for RDBs, 4GLs and CASE, respectively. The three year rule was designed to include enough prior time periods so that the distinctive "take-off" pattern common to innovation diffusion would be evident, but not so many time periods that chance variation in reported acquisitions would be an overwhelming concern.¹² The resulting cumulative adoption-as-acquisition and adoption-as-deployment series are provided in Table 6.3.2.1 below. As this table shows, there are twelve observation periods for RDBs (1982-1993), thirteen observation periods for 4GLs (1981-1993) and eight observation periods for CASE (1986-1993). Cumulative adoptions are reported as percentages of the sample population.

Table 6.3.2.1: Cumulative Acquisition and Deployment for RDBs, 4GLs and CASE (N=384)

Calendar Year	RDBs			4GLs			CASE		
	Per iod	Acqui sition	Deploy ment	Per iod	Acqui sition	Deploy ment	Per iod	Acqui sition	Deploy ment
1981				1	3.2	0.8			
1982	1	2.9	1.6	2	4.8	1.1			
1983	2	4.0	1.6	3	8.3	1.3			
1984	3	6.1	2.7	4	10.7	1.9			
1985	4	13.1	5.3	5	19.5	5.6			
1986	5	17.9	7.8	6	23.3	7.2	1	1.1	0.3
1987	6	25.7	9.9	7	29.1	9.9	2	3.2	0.8
1988	7	34.0	14.4	8	32.6	12.8	3	5.6	1.1
1989	8	40.9	18.7	9	38.2	14.7	4	10.2	1.3
1990	9	51.3	24.3	10	43.9	17.6	5	17.1	1.9
1991	10	59.1	27.5	11	48.4	19.8	6	25.7	3.5
1992	11	72.2	34.0	12	57.2	22.2	7	34.0	5.3
1993	12	84.8	43.0	13	64.4	26.5	8	42.9	7.2

¹²Using a different numbers years prior to the anchor year, such as two years or four years, has a negligible effect on results; the key point is to use some consistent rule for all three SPIs.

6.3.3 Results

Descriptive Results

The cumulative adoption-as-acquisition and adoption-as-deployment series for RDBs, 4GLs and CASE are plotted in Figures 6.3.3.1 through 6.3.3.3 below. (To facilitate visual comparisons, the same scales have been used for all three SPIs, and a vertical line is provided to highlight the eighth year of each series.) RDBs do, as expected, appear to have a moderate assimilation gap. By 1993, about 85% (N=321) of sites had acquired an RDB, although only 51% (N=163) of acquirers had yet deployed RDBs on at least 25% of new development projects. But, both the adoption-as-acquisition and adoption-as-deployment curves show robust increasing trends, and there is no reason to conclude, based on inspection of these series alone, that deployment will not eventually be achieved by a large majority of acquirers.

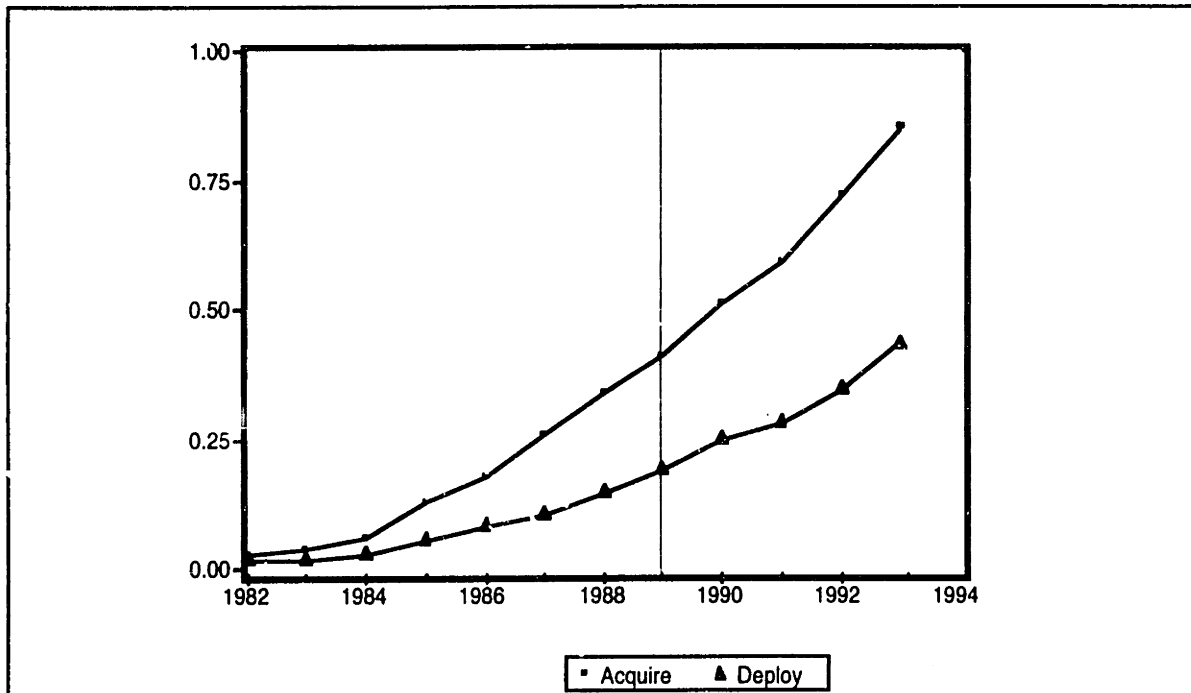


Figure 6.3.3.1: Cumulative Acquisitions and Deployments of RDBs

Production 4GLs appear to have a slightly larger assimilation gap than RDBs. By 1993, about 64% (n=242) had acquired 4GLs, but only 41% (N=99) of those acquirers had deployed. Furthermore, one must go back 6.5 years, to mid-1986, to find the year where cumulative acquisitions equals the number of cumulative deployers in 1993. For RDBs, by comparison, one must go back only 4 years, to 1989, to find the year where cumulative acquisitions equals cumulative deployments in 1993.

For CASE, the apparent assimilation gap is striking. By 1993, 43% (N=160) have acquired a CASE tool, but only 17% of those acquirers (n=27) have deployed. This works out to an acquirer/deployer ratio of 5.9 to 1 at the eight year mark for CASE. For RDBs and 4GLs, by contrast, these ratios were 2.2 to 1 and 2.6 to 1, respectively, by the end of eight years. As a result, this difference is not simply an artifact of the shorter observation period for CASE.

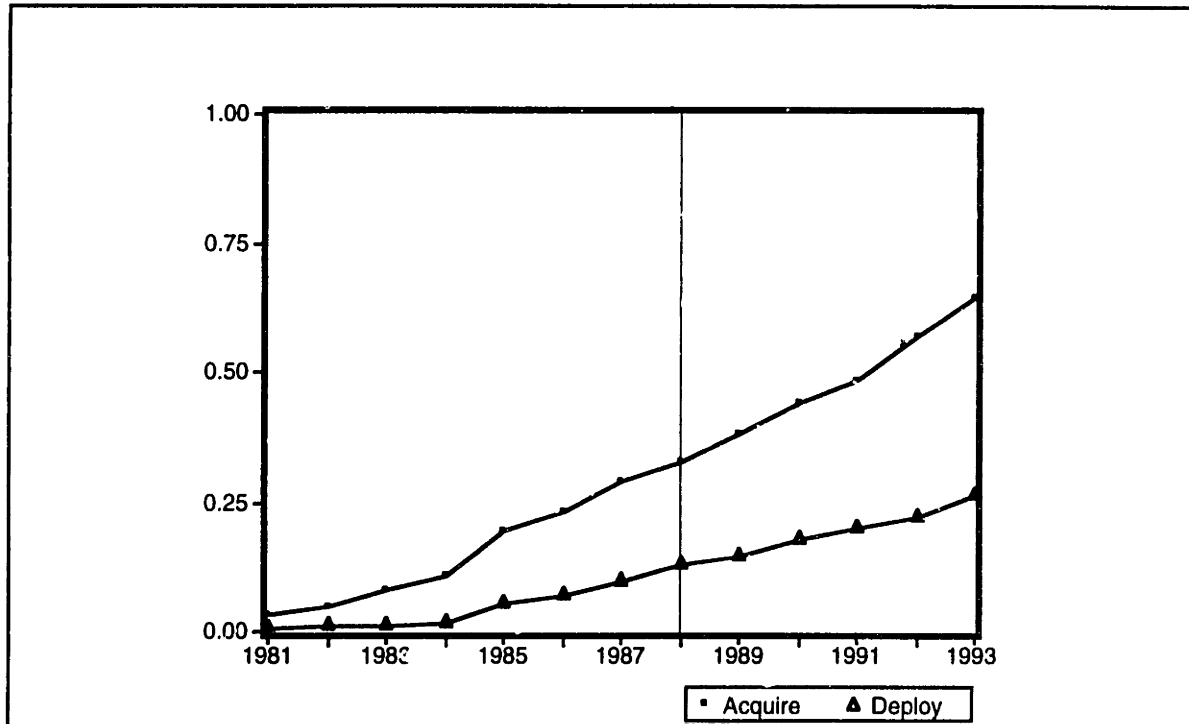


Figure 6.3.3.2: Cumulative Acquisitions and Deployments of 4GLs

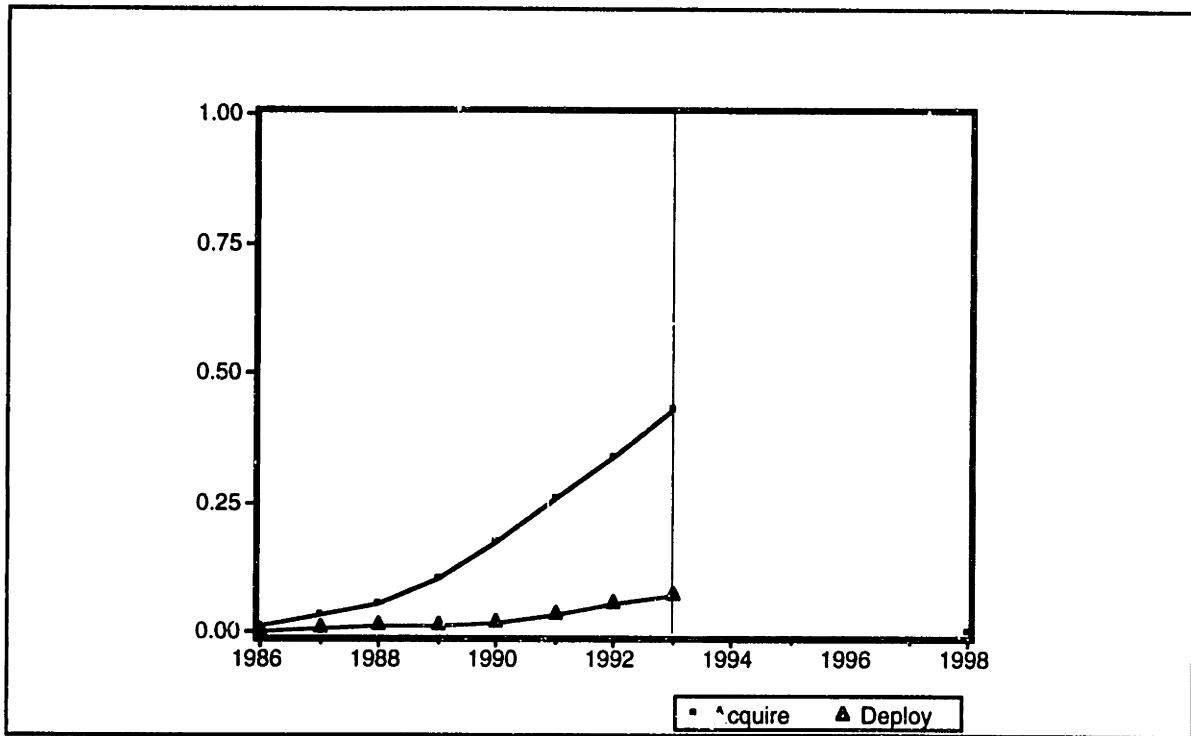


Figure 6.3.3.3: Cumulative Acquisitions and Deployments of CASE

This descriptive analysis has provided an intuitive feel for the magnitude of assimilation gaps, and actual results have appeared sensible when compared with expectations. Nevertheless, a more formalized and general procedure for analyzing assimilation gaps is needed, not only to guard against inferential mistakes (particularly on the margin), but to support replication and comparisons across studies.

Formal Estimation of Assimilation Gaps

The assimilation gap was defined earlier as the area between the cumulative acquisition and deployment curves at time T, as a proportion of the area under the cumulative acquisition curve at time T. For this study, assimilation gaps are computed using the following formula:

$$[1] \quad G_i(T) = \sum_{t=0}^T (A_i(t) - D_i(t)) / A_i(t)$$

where $G_i(t)$ is the estimated assimilation gap for the i^{th} SPI at time t , A_i is the cumulative acquisition function for the i^{th} SPI, and D_i is the cumulative deployment function for the i^{th} SPI. This represents a discrete calculation of the assimilation gap according to the graphical approach described earlier.

The resulting assimilation gaps for the three SPIs are plotted, as a function of time, in Figure 6.3.3.4 below. To support a more useful comparison across SPIs, time is defined as elapsed time since the start year for each SPI, rather than calendar time. The curves are plotted through years 12, 13 and 8, for RDBs, 4GLs and CASE, respectively; this is the number of years for which data were available for each SPI.

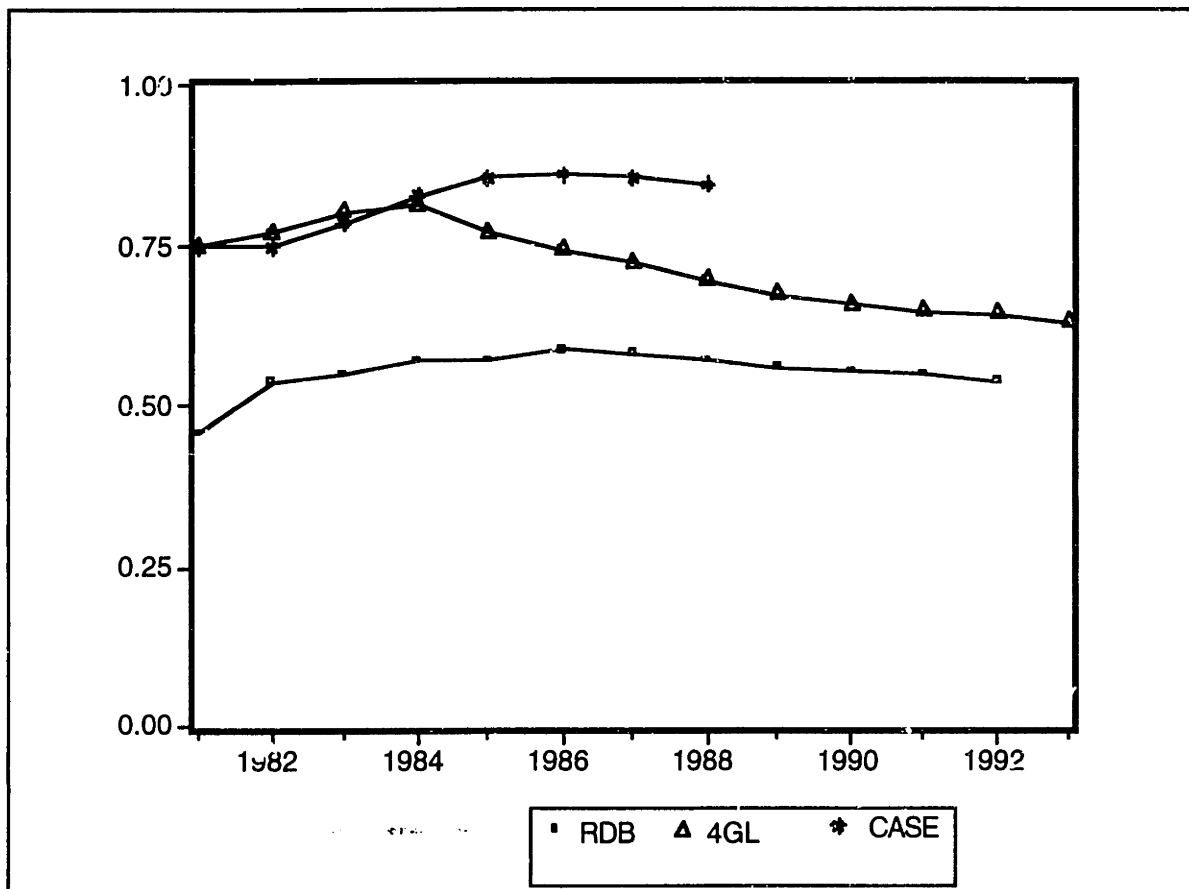


Figure 6.3.3.4: Assimilation Gaps

The patterns evident in Figure 6.3.3.1 are broadly consistent with the results of the descriptive analysis presented in the previous section. RDBs show the smallest gap, with 54% of the area under the cumulative acquisition curve in year 12 (the last observation year) being accounted for by the area between the acquisition and deployment curves. The estimated RDB assimilation gap is stable over time, ranging only from .54 to .59 over the last eleven observation periods. The estimated gaps for 4GLs and CASE are similar for the first three years after their respective start years, and then diverge. Table 6.3.1.1 below highlights the estimated gaps just for years 8 and 12. In year 8, the estimated gap for 4GLs is 21% larger than the one for RDBs, although by year 12, the gap is only 17% larger. In year 8, the CASE estimated assimilation gap in year 8 is 49% larger than the one for RDB, and 23% larger than the one for 4GLs.

Table 6.3.1.1: Estimated Assimilation Gaps - Selected Time Periods

SPI	Year 8	Year 12
RDBs	.57	.54
4GLs	.69	.63
CASE	.85	n/a

The next step in the assimilation gap analysis used survival analysis techniques to assess the level of confidence in the above assimilation gap estimates. This required the creation of six new datasets, one for acquisitions and one for deployments for each of three SPIs. Two variables were included in each SPI dataset, a dummy variable identifying whether the case was censored or not, and a time variable. The time variable contained the elapsed time until adoption for non-censored cases (calculated as the adoption year minus the start year for the SPI), and the total observation time for censored cases (calculated as 1993 minus the start year for the SPI). The product limit approach, as described by Lawless [1982], was used to estimate the survivor

function for acquisitions and deployments, for RDBs, 4GLs and CASE.¹³ Interval endpoints were defined to be 0 to 1, 1 to 2, 2 to 3, and so forth, up to the end of the observation period for each SPI. For the current application, this approach produces survival function estimates exactly equal to the one's complement of the empirical cumulative adoption function for the associated diffusion process. As a result, estimated assimilation gaps based on the survivor functions are exactly equal to those computed using equation [1] above.

The advantage of survival analysis is that it also produces estimates of the standard errors for the survivor function estimates at time T.¹⁴ Figure 6.3.3.5 shows the estimated survivor functions for RDB acquisitions and deployments, together with confidence intervals equal to plus or minus one standard error. This figure shows that even in the unlikely event that all the true survival probabilities for acquisitions were one standard error higher than the estimated probabilities, while at the same time, all the true survival probabilities for deployments were one standard error lower than the estimated probabilities, there would still be a substantial measured assimilation gap. An inspection of the confidence intervals for the survivor functions for 4GLs and CASE (not pictured here) led to the same conclusion for those SPIs.

¹³This non-parametric approach is preferred here over the parametric methods because it requires no assumptions about the functional form for uncensored lifetimes. The Lifetest procedure provided by SAS was used to perform all computations.

¹⁴The SAS Lifetest procedure employed here computes standard errors based on Greenwood's formula (see [Cox and Oakes 1984, p. 50]).

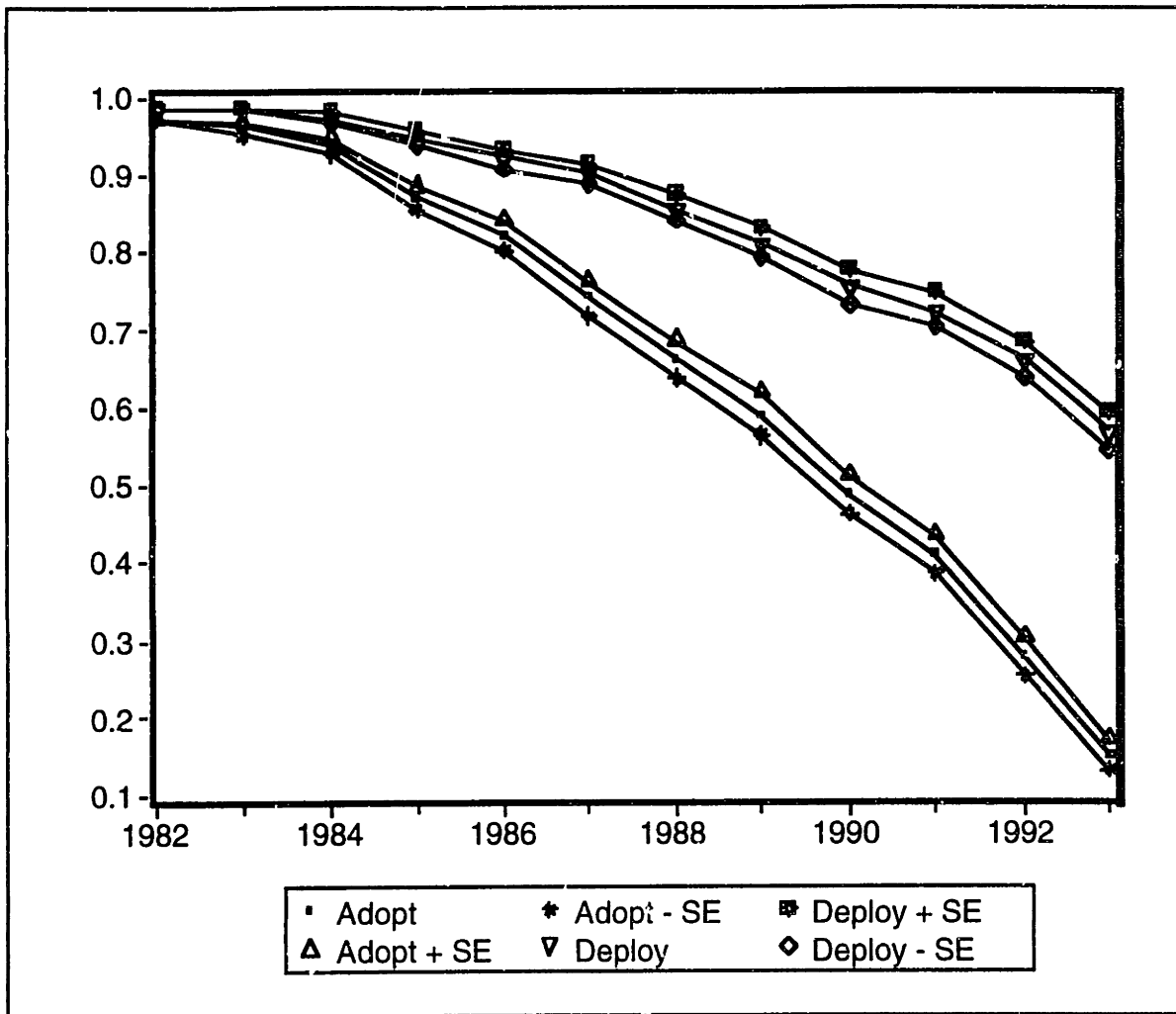


Figure 6.3.3.5: RDB Survivor Functions for Acquisition and Deployment, with Confidence Intervals

Comparing Assimilation Gaps

In the above analysis, the time until acquisition and the time until deployment are treated as separate survival processes, with durations for both beginning in the start year for the associated SPI. As described earlier, it is possible use data on the timing of acquisitions and deployments to model a different kind of survival process, with durations starting at the time of acquisition and stopping at deployment.

As before, a new data set was created for the survival process associated with each SPI. Two variables were included in each SPI dataset, a dummy variable identifying whether the case is censored (because deployment is not observed by 1993, and a time variable. The time variable contains the reported deployment year minus the reported acquisition year for non-censored cases, and the total observation time for censored cases (calculated as 1993 minus the year of acquisition). Interval endpoints were defined to be 0 to 6 months, 6 to 18 months, 18 to 30 months, 30 to 42 months, and so forth, up to the longest observation period for each SPI.¹⁵ Once again, the product limit approach was used to estimate the survivor function for each SPI. The resulting survivor probabilities at the end of each interval are provided for RDBs, 4GLs and CASE Table 6.3.1.2. The survivor functions are plotted in Figure 6.3.3.6.

Table 6.3.1.2: Estimated Survival Probabilities at for the Acquisition to Deployment Process for RDBs, 4GLs and CASE

Time (years)	RDBs	4GLs	CASE
0	1.0	1.0	1.0
.5	.87	.87	.95
1.5	.69	.75	.88
2.5	.53	.62	.84
3.5	.45	.60	.82
4.5	.41	.54	.76
5.5	.37	.54	n/a
6.5	.36	.49	n/a
7.5	.34	.46	n/a
8.5	.27	n/a	n/a

¹⁵These intervals are off-set one half year earlier than the intervals used in prior survivor function computations because, for this process, acquisition happens half way through the reported year of acquisition, on average (assuming a uniform distribution of acquisition times through the year).

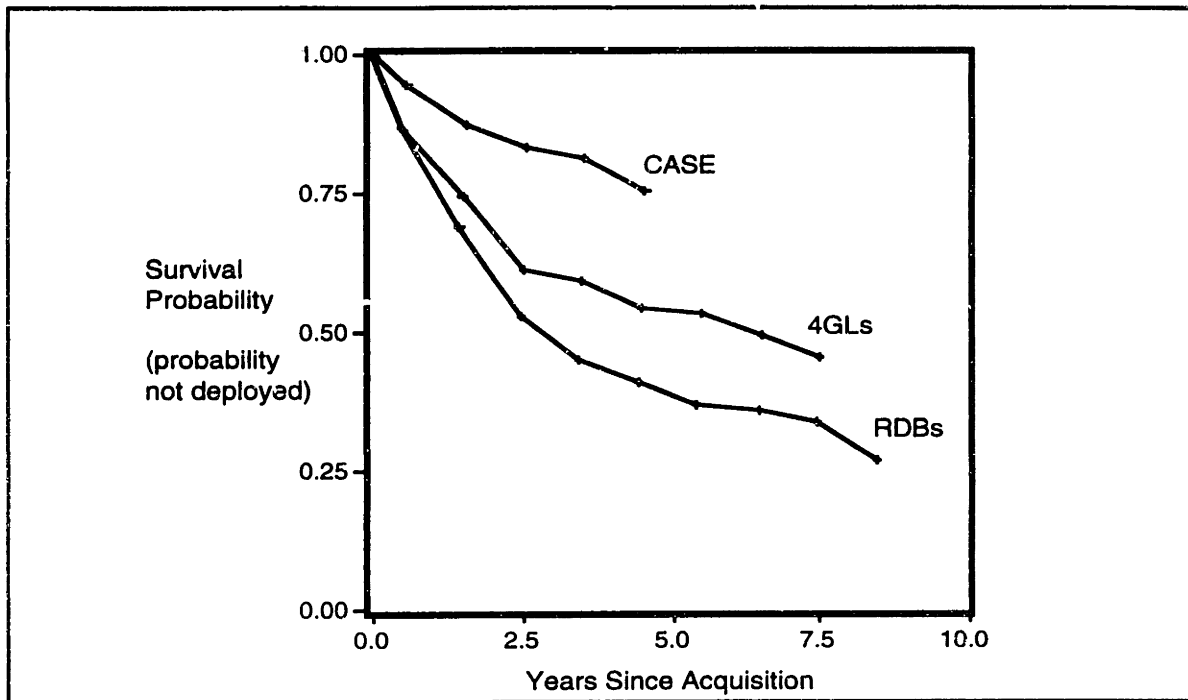


Figure 6.3.3.6: Survivor Functions for Time to Deployment Since Acquisition

The survivor functions plotted above provide a valuable alternative (although still consistent) view of assimilation gaps compared with the area-based measures computed earlier. The more rapidly the survivor function decreases, the smaller the associated assimilation gap will be. While the estimated survivor functions are based on different lengths of observation windows (12 years, 13 years and 8 years for RDBs, 4GLs and CASE, respectively), the estimation procedure is not sensitive to the length of the data collection period (see Appendix 6B).

As before, these plots suggest that RDBs have the smallest assimilation gap, with a slightly larger gap for 4GLs, and very pronounced gap for CASE. By 4.5 years, the latest time for which estimated survival probabilities could be computed for all three SPIs, the survival estimates are .41, .54 and .76 for RDBs, 4GLs and CASE, respectively. Or in other words, 59% of RDB acquirers deploy within 4.5 years, compared with only 24% of CASE acquirers.

If it is assumed, for discussion purposes, that 2.5 years is the maximum duration for "ordinary" lags, then the survivor function can be used to determine estimates of the proportion of SPI acquirers experiencing this outcome. Notice that the survival probabilities for RDBs, 4GLs and CASE at 2.5 years are .53, .62 and .84, respectively. This suggests that 47% of RDB acquirers experience only ordinary delays, compared with 38% for 4GL acquirers and just 16% for CASE acquirers.

The survivor function can also be used to determine a bound for the estimated maximum number of acquirers expected never to deploy. For RDBs, the survival estimate of .27 at 8.5 years suggests that, at most, 27% will never deploy. The survival estimate of .46 for 4GLs at 7.5 years likewise suggests that, at most, 46% will never deploy. Since survival estimates could only be calculated through 4.5 years for CASE, the survivor function value at this time (.76) provides little insight on what the ultimate proportion of CASE non-deployers might be, although a value greater than 50% would not be surprising.

The final step in the analysis of assimilation gaps is to test whether the observed durations for different SPI pairs could have been produced by identical population survivor functions. Two different homogeneity tests were performed for each pairwise combination of SPIs: the Log Rank test, and the Wilcoxon test (see [Cox and Oakes 1984]).¹⁶ The Log Rank test is more sensitive to differences at larger survival times, while the Wilcoxon is more sensitive to differences at shorter survival times. All tests reject the hypothesis of homogeneous population survivor functions at $p \leq .05$, with the exception of a near miss for the Wilcoxon test for homogeneity between RDBs and 4GLs ($p \leq .058$) (See Table 6.3.5.2). These results support the conclusion

¹⁶A third test, the Log Likelihood test, is not used here because it assumes an underlying exponential function for uncensored survival times, an assumption not borne out by the data. If an exponential function were appropriate, then the negative log of survival probabilities versus time should produce a linear curve through the origin—but such a pattern does not exist for any of the three SPIs.

that the survivor functions are different, and by extension, that the assimilation gaps are different as well.

Table 6.3.5.2: Tests for Homogeneity of Survival Processes

Comparison	Log Rank		Wilcoxon	
	χ^2	P-value	χ^2	P-value
RDBs vs. 4GLs	7.6	.006	3.6	.058
RDBs. vs CASE	35.3	.000	28.3	.000
4GLs vs. CASE	16.4	.000	15.2	.000

6.4 ORGANIZATIONAL LEVEL OUTCOMES CONTRIBUTING TO THE ASSIMILATION GAP

As explained in Section 6.2 above, several different kinds of organizational-level outcomes can contribute to the assimilation gap, including: 1) ordinary lags, 2) excessive delay, 3) rejection, 4) implementation failure, 5) stalling, and 6) discontinuance. Ordinary deployment lags can be viewed as a relatively benign occurrence. "Excessive" delays beyond what would ordinarily be expected are of greater concern, because they are a sign of deployment difficulties, and defer the arrival of benefits to both the adopter and the broader adopting community. The other classes of contributing outcomes are of the greatest concern, because these are organizations that have not deployed, and are unlikely to ever deploy the SPI, yet they appear to have done so when cumulative acquisitions are modeled.

Although it was not part of the original scope of this study to collect all the data that would be needed to rigorously partition outcomes contributing to the assimilation gap, some data were nonetheless captured that shed light on this issue. Specifically, data were gathered (for those sites that had at least acquired an SPI) on whether future use of the SPI was expected to increase, stay the same, or decrease. These data, when coupled with data on the elapsed

time until deployment, can be used to support a rough partitioning of outcomes contributing to the assimilation gap, as illustrated below.

To begin this task, organizations acquiring any of the three SPIs in the 1989 to 1991 window were extracted for analysis. This time window was selected to ensure the following: that all included organizations would have had a minimum of two years in which to deploy the SPI; that at least 50 cases would be available for analysis for each SPI; and that reports of future expected use would not be too far removed from the time of acquisition for most cases.¹⁷ This resulted in the selection of 94, 59, and 75 cases, respectively, for RDBs, 4GLs and CASE, distributed across years of acquisition as shown in Table 6.4.1.

Table 6.4.1: Number of SPI Acquirers in 1989, 1990 and 1991

	RDBs	4GLs	CASE
1989	26	21	17
1990	39	21	26
1991	29	17	32
Total	94	59	75

As the first step in partitioning organizational-level outcomes, acquirers of each SPI were classified according to whether they had deployed by the end of 1993, or not. For the purpose of this analysis, those that deployed a particular SPI by 1993 are classified as having experienced "ordinary" lags for that SPI. If the assumption is made that adoptions and deployments are distributed uniformly within the reported years, then about 80% of these organizations required two years or less to deploy, and virtually all required 2.5 years or less. Those not deploying by the end of 1993, are classified as experiencing "excessive" delays or worse. All of these organizations had at least two years in which to deploy, and many had three or four. The results of this

¹⁷Using different length time periods, such as 1990 to 1991, or 1988 to 1992, has little impact on results.

categorization are presented in Table 6.4.2. For RDBs, 51.1% deployed by the end of 1993. A similar proportion of 4GLs acquirers, 42.4%, deployed by the end of 1993, while only 16.0% of CASE acquirers did so. (The full survivor function estimates that 47%, 38% and 16%, of RDB, 4GL and CASE, acquirers, respectively, deploy within 2.5 years.)

Table 6.4.2: Breakdown of Deployment by 1993

	RDBs (N=87)	4GLs (N=55)	CASE (N=73)
Deployed by 1993	51.1%	42.4%	16.0%
Not deployed	48.9%	57.6%	84.0%
Total	100%	100%	100%

The second step in the analysis is based on those cases than had not yet deployed by the end of 1993, and hence, ultimate outcomes can only be surmised. The number of such cases is 46, 34 and 63, for RDBs, 4GLs and CASE, respectively. These cases were categorized according to whether the sites expected to increase their use of the SPI to develop production applications in the next three years, to decrease their use, or to maintain current levels of use. Table 6.4.3 provides the results of this classification for the three SPIs. About four-fifths of RDB acquirers that had yet to deploy nevertheless expected their use to increase. About three-fourths of 4GL non-deployers expected usage increases. Only one-half of CASE non-deployers expected their use to increase.

The proportion of non-deployers that expected their current level of use to stay about the same were 17.4%, 17.6% and 38.1% for RDBs, 4GLs and CASE, respectively. Finally, the percentage of non-deployers reporting expected usage decreases were 2.2%, 5.9%% and 17.6% for RDBs, 4GLs and CASE, respectively.

Table 6.4.3: Expected Future Use for Sites Not Deploying by 1993

	RDBs (N=46)	4GLs (N=34)	CASE (N=63)
Use expected to increase	80.4%	44.0%	52.4%
Use expected to stay the same	17.4%	17.6%	38.1%
Use expected to decrease	2.2%	5.9%	17.6%
Total	100%	100%	100%

Those organizations still increasing their deployment at the end of the observation window are at least candidates to one day reach the 25% level. For this analysis, all these organization are classified as future deployers, that nonetheless experienced "excessive" delay. Those organizations reporting that future use was expected to stay at current levels or decrease are classified as falling in one of the four remaining categories, i.e., rejection, implementation failure, stalling, or discontinuance. Without additional data, it is not possible to determine with any confidence which of these four particular outcomes best applies in individual cases.

Table 6.4.4 provides a summary of the combined results of both steps in the analysis, including the first step, which identified cases for which outcomes were known, and the second step, for which ultimate outcomes could only be surmised.

Table 6.4.4: Summary of Presumed Outcomes for SPI Acquirers

	RDBs (N=87)	4GLs (N=55)	CASE (N=73)
Normal lag	51.1%	42.4%	16.0%
Excessive delay	39.4%	44.1%	44.0%
Rejection/failure/ stalling/discontinuance	9.6%	13.8%	40.0%
Total	100%	100%	100%

For RDBs, over half the cases (51.1%) fall in the "ordinary" lag category, and only 9.6% fall in the least favorable categories (rejection, implementation

failure, stalling, and discontinuance). For 4GLs the distribution is not quite as promising, with 42.4% falling in the fall in the "ordinary" lag category, and 13.8% falling in the least favorable categories. CASE has by far the worst distribution, with only 16.0% falling in the "ordinary" lag category, and 40% falling in the in the least favorable categories. As a result, not only does CASE have largest estimated assimilation gap, but appears to have the least favorable distribution across the organizational-level outcomes contributing to the gap. This is as would be expected, since "ordinary" lags and "excessive" delay contribute the gap for a finite time (until deployment actually does occur), while the other categories continue to contribute for the remainder of the observed diffusion process. Even so, the results for CASE are striking: at least 40% of CASE acquirers for the 1989 to 1991 time period appear headed for an outcome of rejection, implementation failure, stalling, or discontinuance.

6.5 DISCUSSION AND CONCLUSIONS

This research has found sizable assimilation gaps do exist for RDBs, 4GLs and CASE. At the end of eight years, that latest time for which assimilation gaps can be estimated for all three SPIs, the estimated gaps are .57, .69 and .85 for RDBs, 4GLs and CASE, respectively. This means that for all three SPIs, the assimilation gap comprises more than half of the area under the cumulative acquisition curve through the first eight years. At twelve years into the diffusion process, the assimilation gap for RDBs is .54, and the one for 4GLs is .63. Survival analysis techniques confirmed that, as expected, the assimilation gap for CASE is significantly larger than the one's for RDBs and 4GLs; it was also found that the assimilation gap for RDBs is significantly larger than the one for 4GLs. Finally, a partitioning of the organization-level outcomes contributing to the assimilating gap leads to the conclusion that CASE has the least favorable distribution across outcomes (i.e., a smaller proportion "ordinary" lags, and a larger proportion of cases where the SPI is unlikely ever to be deployed.).

This is an exploratory study, and as such, is subject to some limitations. First, only limited data were available to support the important task of partitioning organization-level outcomes contributing to assimilation gaps.

Second, managers were asked to provide retrospective reports on the timing of events that, in some cases, occurred many years in the past. This can be expected to introduce noise some into the measurement process. It might also introduce a systematic bias. For example, it might be that organizations that have reached the point of 25% deployment are systematically more likely to over report the time since acquisition compared with those that have not reached the point of 25% deployment. Gross over reporting of this kind seems unlikely, since the plotted cumulative adoption curves are similar to what one would be expected based on general knowledge of when the innovations were introduced. And in any case, systematic over reporting like this could not account for the study's most important finding, namely that CASE has the largest gap and RDBs the smallest. This is because RDBs have the highest percentage of prior acquirers and CASE the lowest, and therefore any systematic over reporting of this kind would act to narrow the observed differences in assimilation gaps rather than to exaggerate them.

Third, the rule to define deployment (25% use) is somewhat arbitrary, and while, as argued earlier, this appears to be a fair standard for the SPIs included in this research, it might be too strict a standard (or not strict enough) to serve as an indicator of "significant" deployment for other SPIs.

Finally, especially for CASE, the portion of the whole diffusion process that was covered in the observation periods was less than optimal. For all three innovations, only the portion of the innovation process for which both acquisitions and deployments were still experiencing steady growth was observed, thus accounting for the relative stability of measured gaps over time. It would be interesting to examine the pattern of assimilation gaps for a process where the acquisition levels, deployment levels, or both, were approaching saturation.

Nevertheless, these potential limitations are not sufficient to bring into serious question the broad conclusions of the study. This study has invented the assimilation gap concept, has demonstrated that it can be sensibly measured, and has demonstrated that its measured size is largely consistent with *a priori* expectations for three prominent IT innovations. In addition, this research has identified statistical techniques to assess assimilation gap estimates and to draw conclusions about the size of one assimilation gap in relation to others.

In verifying the existence of substantial assimilation gaps and identifying techniques for their study, this research has opened a new domain of diffusion analysis, with potentially far reaching implications for researchers and other innovation stakeholder groups (i.e., users, vendors, mediating institutions). For innovation stakeholders, the assimilation gap concept provides a new tool for assessing the robustness of diffusion of an innovation in the first several years after introduction, and by extrapolation, its future prospects. The information technology field has produced many innovations that had extraordinary promise, and appeared to be diffusing quite rapidly, but never did approach expected levels of use and impact. This research has shown that it is possible to identify, well before the sales peak, that such a pattern may be in evidence. Based purely on acquisitions, CASE would be viewed as the most robust of the three SPIs studied here; yet the pattern of actual deployments paints a much different, and more ominous picture, one that is more consistent with emerging industry opinion on the prospects and role of CASE tools.

For researchers, the assimilation gap concept implies, first, that any study that models diffusion based on a single definition of the adoption event must select this event with the utmost care, or else risk producing an illusory picture of the diffusion process. Second, the assimilation gap concept in itself provides a rich area for potential study. The question of why some innovations might be vigorously acquired but sparsely deployed is in need of

further study. One potential explanation for the case of SPIs—that the expectations of future benefits, owing to increasing returns, drives rapid acquisition, but knowledge barriers impede deployment—has been suggested by this research, but there may be other explanations that plausibly account for this phenomenon.

APPENDIX 6A: ASSIMILATION GAP ESTIMATION BASED ON DIFFUSION MODELING TECHNIQUES

Many of the functional forms commonly employed in diffusion modeling include a parameter that is traditionally interpreted as the overall "rate" of diffusion. The estimated values for this parameter can be used to compute an operational indicator of the assimilation gap concept based on the difference in diffusion rates for cumulative acquisitions versus cumulative deployments. More specifically, an operational indicator can be computed as *the difference between the acquisition rate and deployment rate divided by the acquisition rate.*

Many prior studies have looked to estimated diffusion rate parameters as a basis for comparing innovations [Trajtenberg and Yitzhaki 1989]. One of the most common functional forms employed in such studies is the logistic [Teece 1980; Mansfield 1993], and this is the form employed here. The logistic appears to more closely fit the general shape of the observed cumulative acquisition and deployment series than other common forms, such as the exponential or Gompertz,¹⁸ and use of this functional form promotes comparability with previous studies.

The analysis presented in this appendix is modeled after the recent work by Mansfield [1993] investigating the diffusion of flexible manufacturing systems, but with one important exception: while Mansfield used ordinary least squares regression (after appropriate logarithmic transformations) to estimate model parameters, this analysis uses non-linear least squares to estimate equations directly. Recent research has found that non-linear least squares leads to better estimates of diffusion parameters and standard errors,

¹⁸A pattern of increasing slopes is observed throughout the observation period for all six series. As a result, the exponential, which exhibits decreasing slope for the entire function, is clearly inappropriate. The Gompertz, like the logistic, begins with a pattern of increasing slopes, but expects the inflection point to occur at the 38% level of penetration. For the actual series, it appears that the inflection point occurs at the 50% level or later, so the logistic, which expects the inflection point to occur at the 50% point, is more appropriate.

compared with linear and maximum likelihood approaches [Srinivasan and Mason 1986].

Two sets of analyses are done, one based on the first 8 years of data (the maximum window for which data were available for all three SPIs) and the second based on the first 12 year (the maximum window for which data were available for RDBs and 4GLs). The purpose of using both 8 and 12 years of data is to support an assessment of the sensitivity of estimates to the number of time periods used in the analysis, an issue raised by the availability of only 8 years of data for CASE.

The specific equations to be estimated are:

$$[1] \quad Y_{ij}(t) = [1 + e^{-(L_{ij} + \phi_{ij}t)}]^{-1}$$

where $Y_{ij}(t)$ is the proportion of potential adopters in time t of the i^{th} SPI (RDBs, 4GLs or CASE), using the j^{th} approach to defining the adoption event (acquisition or deployment). L_{ij} and ϕ_{ij} are parameters, with the latter corresponding to the diffusion rate.

The analysis begins with estimation of the rates of adoption-as-acquisition ($j=1$) for the three SPIs. Table 6A-1 shows the resulting estimates, computed for the two alternative time windows (the first 8 years and the first 12 years.

Table 6A-1: Estimated Rates of Adoption-as-Acquisition for RDBs, 4GLs and CASE

SPI	8 Year Series			12 Year Series		
	ϕ_{ij}		R^2	ϕ_{ij}		R^2
	Estimate	t -value		Estimate	t -value	
RDBs	.42	17.28	.99	.42	25.65	.99
4GLs	.34	10.39	.96	.27	17.45	.98
CASE	.47	17.79	.99	n/a	n/a	n/a

All estimated equations show a very high fit to the data, with R^2 values ranging from .96 to .99. All estimates of the diffusion rate parameters are likewise highly significant, with t -values ranging from 10.39 to 25.65. The estimates for ϕ_{ij} were the same for RDBs regardless of whether 8 or 12 years were used, where as the difference in time windows had a notable impact on the estimates for 4GLs. Specifically, using the 12 year time window lowers the estimate of ϕ_{ij} about 20%, from .34 to .27. This latter result casts some doubt on the stability of the ϕ_{ij} estimate for CASE, which was based on only the first 8 years of observed diffusion. According to the 8 year estimates, CASE is diffusing the fastest (.47), followed by RDBs (.42) and 4GLs (.34). These values appear reasonable, considering that as of the end of eight years, acquisition levels of 43%, 41%, and 33%, had been achieved by CASE, RDBs, and 4GLs, respectively (See Table 6.3.2.1).

The next step in the process of computing assimilation gaps is to estimate the parameters for the adoption-as-deployment curves ($j=2$). Table 6A.2 provides the results of these estimates.

Table 6A.2.: Estimated Rates of Adoption-as-Deployment for RDBs, 4GLs and CASE

SPI	8 Year Series			12 Year Series		
	ϕ_{ij}		R^2	ϕ_{ij}		R^2
	Estimate	t -value		Estimate	t -value	
RDBs	.38	20.22	.99	.31	27.35	.99
4GLs	.40	10.12	.97	.24	11.05	.95
CASE	.42	18.64	.99	n/a	n/a	n/a

As with the estimates for adoption-as-acquisition, these estimates are all highly significant. Unfortunately, however, they are not stable. Increasing the observation window from 8 to 12 years reduces the ϕ_{ij} estimate 17% for RDBs, and 39% for 4GLs. Furthermore, the 8 year estimates of the diffusion

rates are highly counter-intuitive. RDBs, which had the highest actual cumulative percentage of deployments by the end of 8 years (18%) has the lowest estimated diffusion speed (.38) for deployments based on 8 years of data. Likewise, CASE, which had by far the lowest cumulative percentage of deployments at the end of 8 years (7%), had the highest estimated speed (.42) in the first 8 years.

Not surprisingly, the assimilation gap values computed based on these estimated diffusion rates are uninterpretable (see Table 6A.3 below). Even when using 12 years of data for RDBs and CASE, the resulting estimates, which show an RDB assimilation gap 2.5 time greater than the 4GL assimilation gap, are at completely odds with other evidence considered so far regarding the relative size of the assimilation gaps for these SPIs.

Table 6A.3: Estimated Assimilation Gaps

SPI	Time Windows	
	8 year series	12 year series
RDBs	.11	.25
4GLs	-.18	.10
CASE	.11	n/a

To conclude, the conceptual concerns described earlier about the validity of the diffusion modeling approach for this data set and application appear to have been well placed. Not enough data are available to properly estimate models for the adoption-as-deployment series. Yet, even if more data were available, there would still be the problem associated with the interaction of the adoption ceiling and the diffusion rate. An assumption of a 100% adoption ceiling for processes unlikely to reach the 50% would likely introduce unacceptable distortion in the estimation process, yet introducing an assumption of a less than 100% adoption ceiling renders the diffusion rate parameter almost meaningless as an indicator of diffusion speed for this application, because the rate would be based on differing subsets of the sample.

APPENDIX 6B: SENSITIVITY ANALYSIS FOR SPI SURVIVOR FUNCTIONS

As with the diffusion modeling approach described in Appendix 6A, a sensitivity analysis was performed for the estimated survivor functions for the durations from acquisition to deployment. Calculations were performed using data on acquisitions and deployments for 8 versus 12 year time windows for RDBs and 4GLs.

Figure 6B.1 shows the calculated survivor functions for the durations from acquisition to deployment for RDBs. Unlike with the diffusion modeling approach, the use of 8 versus 12 years of data has little impact on estimation. When more data are available the procedure is of course able to produce survival probability estimates for more time periods; but for the time period for which both approaches can produce estimates, the estimates are quite consistent.

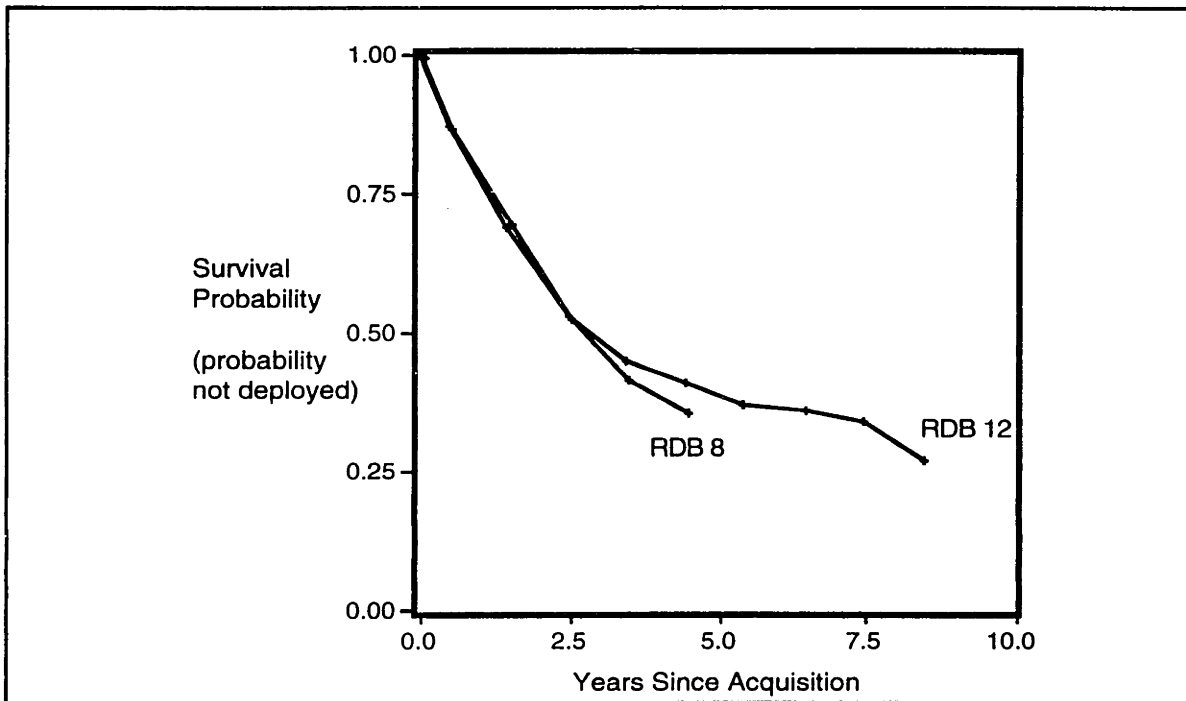


Figure 6B.1: Survivor Function For Time to RDB Deployment Since Acquisition

The survivor functions for the time from 4GLs acquisition to deployment, again calculated based on 8 year (1981-1988) and 12 year (1991-1992) observation windows, are provided in Figure 6B.2 below. Once again, the survival function estimates are quite similar.

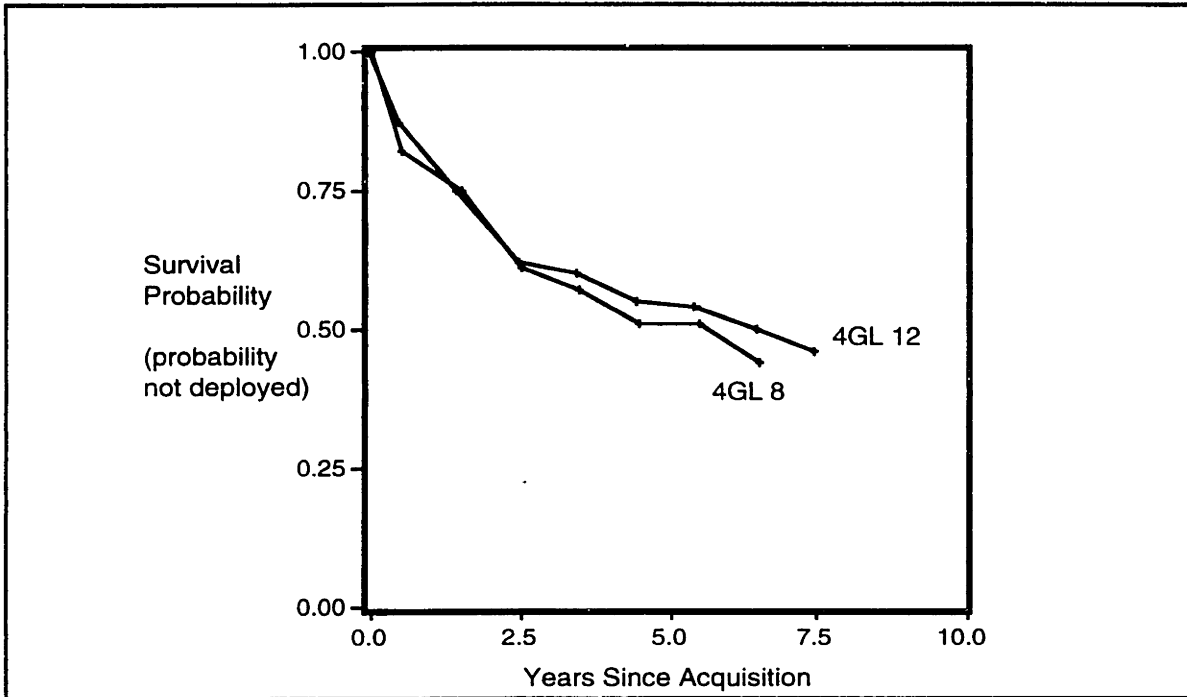


Figure 6B.2: Survivor Functions for Time to 4GL Deployment Since Acquisition

The comparative stability of survivor function estimates is likely due to the fact that the estimation procedure makes no assumptions about durations yet to be observed, but rather, computes estimates based solely on observed data points. As a result, so long as there is not strong, systematic relationship between survivor probabilities and acquisition cohort (as defined by the calendar year of acquisition), then one should expect few differences in computed probabilities beyond those due to sampling variation.

CHAPTER 7. SUMMARY AND CONCLUSIONS

This chapter summarizes the results of this research, discusses theoretical and managerial implications of the results, and suggests directions for future research.

7.1 SUMMARY OF THE RESEARCH

Chapter 1

Chapter 1 presented a first step towards of a macro-level theory of the assimilation and diffusion of software process innovations, and in so doing, provided a broad conceptual foundation for all the remaining chapters. The chapter began by reviewing the classical model of diffusion developed by Everett Rogers [1983], a natural starting point since the basic questions guiding this research are the same ones that have occupied diffusion researchers for decades. Several criticisms were then articulated that shed doubt on the suitability of the classical model as a general model applicable to all innovations and contexts. In particular, it was argued that SPIs possess two distinctive characteristics—increasing returns to adoption [Arthur 1988] and knowledge barriers impeding adoption [Attewell 1992]—that are strongly at odds with the implicit assumptions of the classical model. The theoretical implications of increasing returns and knowledge barriers, each taken singly, were then summarized. Finally, Chapter 1 proposed six components of a theory of the diffusion of SPIs derived from the combined implications of increasing returns and knowledge barriers, summarized here as follows:

- 1) When an SPI is strongly subject to increasing returns, a wide discrepancy will exist between its initial performance and its network potential.
- 2) Due to the presence of knowledge barriers and the immaturity of the technological network, most early adopters will find the initial performance of SPIs to be lower than pre-existing best practices.
- 3) SPIs will be prone to an assimilation gap (a gap between the cumulative acquisition and cumulative deployment curves) during

the early adoption cycle, because the promise of benefits resulting from increasing returns drive rapid acquisition, but knowledge barriers and other costs associated with joining an immature technological network impede deployment.

- 4) The deployment of an SPI makes a much greater contribution to increasing returns than simple acquisition.
- 5) When a large assimilation gap occurs during the early adoption cycle, a stalled bandwagon will usually follow, because slow or failed deployment among early adopters delays the learning-by-using and other forms of increasing returns needed to make the SPI attractive to the mass market of potential adopters.
- 6) Concerted actions by supply side and mediating institutions (heavy investments in learning-by-doing and learning-by-using; technology standardization; technology sharing arrangements), can, in principle, lower knowledge barriers sufficiently, even in face of an initial assimilation gap, to provoke a sustained bandwagon.

Chapter 2

Chapter 2 presented a detailed examination of the relationship between knowledge barriers and innovation, first introduced in Chapter 1. Based on this examination, it developed a theoretical model explaining differences in the propensity of organizations to initiate and sustain innovations in software process technology. Following Downs and Mohr's position that researchers should develop theories of the "middle-range" focusing on the distinctive qualities of particular kinds of technologies, the proposed model placed factors promoting organizational learning at its core. Specifically, it was proposed that organizations with a higher propensity to innovate would be distinguished by three characteristics: 1) greater scale of activities over which learning costs can be spread ("learning-related scale"), 2) more extensive pre-existing knowledge in areas related to the innovation [Cohen and Levinthal 1990; Pennings and Harianto 1992a], and 3) greater diversity of knowledge and activities in general [Cohen and Levinthal 1990; Swanson 1994]. It was argued that organizations possessing these characteristics should be better able to amortize learning costs, should be better able to assimilate any

given amount of information related to SPIs, and should have a lesser burden of organizational learning in the first place.

Chapter 3

Chapter 3 presented an empirical test of the model developed in Chapter 2, based on data collected from over 600 information technology departments on the assimilation of object-oriented programming languages (OOPLs). Three main hypotheses were tested:

H1: Learning-Related Scale is positively related to Assimilation Stage.

H2: Related Knowledge is positively related to Assimilation Stage.

H3: Diversity is positively related to Assimilation Stage.

Assimilation Stage was operationalized as the highest stage achieved by the site, based on the following Guttman scale: 1) Awareness, 2) Interest, 3) Evaluation/Trial, 4) Commitment, 5) Limited Deployment, and 6) General Deployment. Learning-Related Scale was operationalized as the volume of new development activity at the site. Related Knowledge was operationalized as the extent of staff development experience in four areas related to developing OOPL-based applications: 1) C-language development, 2) client-server application development, 3) PC/workstation application development, and 4) development of graphical user interfaces. Diversity was operationalized as the number different categories of programming languages, runtime platforms, and applications architectures employed in a significant way at the site.

All three hypotheses were strongly supported. When the three independent constructs and six control variables were regressed on Assimilation Stage, the estimated coefficients for Learning-Related Scale, Related Knowledge, and Diversity were all positive and highly significant (p-values all less than .001). The full model, with an $R^2=.30$, explained significantly more variance than

the controls-only model ($R^2=.18$). It was also found that while four of six control variables were significant predictors in the controls-only model, only one control variable, a measure of IT department size, significantly predicted Assimilation Stage in the full model. Several analyses were performed to assess construct validity, as well as internal, external, and statistical conclusion validity; no unmanageable threats to the research results were revealed.

Secondary results included the finding that the profile of organizations classified as Rejectors more nearly resembled the profiles of those organizations in early assimilation stages (Awareness and Interest), rather than those in later stages (Evaluation, Commitment and Deployment). This was consistent with the expectation that organizations initiating assimilation "too soon"—as implied by their comparatively low levels of predictor variables—are less likely to sustain innovation. It was also found that the influence of Diversity became non-significant when analysis was limited to just those organizations in the later stages of assimilation. This raises the possibility that, while Diversity facilitates early assimilation activities, its role during deployment may be mixed. Finally, the disk-based mode of survey administration appeared to be quite effective. A high response rate was achieved with only a small number of unusable responses. The vast majority of respondents had positive attitudes towards the prospect of completing future disk-based surveys, and there were few complaints about the medium voiced by those non-respondents contacted by telephone.

Chapter 4

Chapter 4 provided a conceptual analysis of six measures of the adopter innovativeness concept, including: 1) Time of Adoption, 2) Dichotomous Adoption, 3) Aggregated Adoption, 4) Diffusion, 5) Infusion, and 6) Assimilation Stage. For each measure, the analysis included a description of the origins of the measure, potential advantages and limitations, suggested operationalizations for the case of software process innovations, and guidance

on when and how the measure would likely best be applied. The purpose of the analysis was to lay a conceptual foundation for the empirical analysis of alternative innovativeness measures presented in Chapter 5, and to confirm that Assimilation Stage, the primary innovativeness measure employed in this dissertation, is in fact the most suitable measure.

The chapter concluded that each measure had a unique combination of potential strengths and limitations, and that no one measure dominated any of the others. The more traditional measures, such as Time of Adoption and Dichotomous Adoption, were found to be thin but very general. Newer measures, such as Diffusion and Infusion, were found to be richer but have more limited application and generalizability. Assimilation Stage was found to provide a combination of moderate richness and broad generalizability, and was further found to be suitable for all three major research designs (single innovation, multi-innovation, and innovation-decision). In addition, some particular strengths were noted (e.g., the ability of Time of Adoption to support studies with time-varying predictors) and well as some particular weaknesses (e.g., the inability of Aggregated Adoption to support studies with innovation-specific predictors).

Chapter 5

Chapter 5 provided an empirical complement to the conceptual analysis of alternative innovativeness measures provided in Chapter 4. Data on RDBs, 4GLs and CASE were used to support an analysis of four empirical issues:

- Issue 1: Do richer innovativeness measures generally lead to more strongly predictive models?
- Issue 2: Does Aggregated Adoption suffer from a lack of correspondence with other innovativeness measures?
- Issue 3: Can the results achieved with Aggregated Adoption be confidently generalized to particular innovations in the set?

Issue 4: Are the extent of implementation measures (Diffusion and Infusion) as strongly predictive as the other four measures?

A more parsimonious version of the model employed in Chapter 3 was used to support most analyses. Major results included: 1) richer measures do not necessarily lead to more strongly predictive models; 2) positive associations exist between the propensity to acquire many innovations, to acquire any given innovation early, and to implement innovations in depth; 3) models using Aggregated Adoption have the greatest variance explained, and 4) Diffusion and Infusion are especially difficult to predict.

Chapter 6

Chapter 6 presented an empirical analysis of one of the propositions developed in Chapter 1, namely, that SPIs would be prone to an "assimilation gap". The chapter began by formalizing the assimilation gap concept as the difference between the pattern of cumulative acquisitions and deployments of an innovation across a population of potential adopters. Organizational level outcomes contributing to gap were identified, including: 1) "ordinary" lags, 2) "excessive" delay, 3) rejection, 4) implementation failure, 5) stalling, and 6) discontinuance. Techniques for the measurement and analysis of assimilation gaps were then proposed, including a graphical approach and an approach based on survival analysis.

The operational definition for the assimilation gap employed was the area between the cumulative acquisition and deployment curves at time t as a proportion of the area under the cumulative acquisition curve at time t . The year of acquisition was defined as the year in which the SPI was first installed. The year of deployment was defined as the first year in which the SPI was used on at least 25% of new application development projects. Moderate assimilation gaps were found for RDBs and 4GLs, and a very pronounced gap was found for CASE. At the end of eight years of diffusion, the measured gap

for CASE was 49% larger than the one for RDBs, and 23% larger than the one for 4GLs.

Next, for each SPI, a survival process was defined extending from acquisition to deployment. Survival analysis techniques were used to support three additional analyses related to assimilation gaps. First, a visual inspection of the estimated survivor functions for these three process was performed. The survivor function provides estimates of the probability that a randomly selected acquirer will not have deployed by time t ; therefore, the more slowly the survivor function decreases, the larger the associated assimilation gap will be. As expected, the survivor functions for CASE decreased much more slowly than the ones for RDBs and 4GLs. At 4.5 years after acquisition—the latest time for which estimated survival probabilities could be computed for all three SPIs—the survival estimates were .76, .41, and .54 for CASE, RDBs and 4GLs, respectively. Or in other words, a scant 24% of CASE acquirers were estimated to deploy within 4.5 years, compared with 59% of RDB acquirers and 46% of 4GLs acquirers.

A second analysis was performed to estimate the proportion of the acquiring population experiencing more than "ordinary" lags. It was assumed for discussion that 2.5 years was the maximum duration for "ordinary" lags. The survival probabilities for RDBs, 4GLs and CASE at 2.5 years were estimated as .53, .62 and .84, respectively. This suggests that 47% of RDB acquirers experience only ordinary delays—the most benign outcome contributing to the assimilation gap—as compared with 38% for 4GL acquirers and just 16% for CASE acquirers.

Third, an analysis was performed to determine whether the observed durations, from acquisition to deployment, for the different SPI pairs could have been produced by identical population survivor functions. Two homogeneity tests were performed for each pairwise combination of SPIs. The hypothesis of homogeneous population survivor functions was rejected in all cases at $p \leq .05$, with the bare exception of the two tests between RDBs and

4GLs ($p \leq .058$). These results support the conclusion that the assimilation gap for CASE is significantly larger than the one for 4GLs, which, in turn, is significantly larger than the one for RDBs.

7.2 IMPLICATIONS AND CONTRIBUTIONS

The research described above can be viewed as three distinct but related studies:

- 1) An examination of the antecedents of OOPL assimilation (Chapters 2 and 3);
- 2) An analysis of alternative measures for the adopter innovativeness concept (Chapters 4 and 5);
- 3) An examination of SPI assimilation gaps (Chapter 6).

The implications of these three studies are discussed below.

Study 1: Antecedents of OOPL Assimilation

The OOPL assimilation study has implications for theory, methods, and practice. With regard to theory, this study has strongly confirmed the expected influence of three novel organizational characteristics—Learning-Related Scale, Related Knowledge, and Diversity—on the innovation process. This result has several implications that should be of considerable interest to the innovation diffusion researchers. First it demonstrates the value, if not the necessity, of focusing on more narrow innovations and contexts, and developing theory built around the distinctive elements of those innovations and contexts. This study focused on one class of innovation (SPIs) adopted by a particular function (the IT department) of a particular kind of adopter (end-user organizations). One of the criticisms of prior research in organizational diffusion has been the absence of strong and consistent findings [Downs and Mohr 1976; Rogers 1983]. While more recent research suggests that this criticism may have been exaggerated [Damanpour 1991], it is nevertheless still

true that studies with results as strong as this one have been comparatively rare, especially when the focal innovation has been IT [Fichman 1992]. One of the more plausible explanations for this may well be that working at a theoretical level too far abstracted from particular innovations and contexts precludes strong results.

Second, this study identifies a set of three novel variables that appear to be crucial in predicting assimilation of an important class of innovations. The theoretical model employed in this dissertation was developed to address the specific question of what kinds of organizations should have a higher propensity to innovate which technologies subject to knowledge barriers. While the theory was tested using innovations considered to be exemplars of such technologies, the theoretical model should generalize to other technologies likewise strongly subject to knowledge barriers (e.g., manufacturing process innovations, such as flexible manufacturing). Only a small percentage of prior diffusion studies have focused on such technologies; and among those that have, learning-related factors have generally been absent. (Two notable exceptions are Attewell [1992] and Pennings and Harianto [1992a].) The strong influence of Related Knowledge—combined with the fact that this construct does not strongly covary with "generic" innovation predictors—illustrates the value of examining particular technologies in detail, with models that incorporate innovation-specific predictors. The strong influence of Learning-Related Scale and Diversity—combined with the fact that these constructs *do* covary with other common innovativeness predictors—suggests that future research on the assimilation of complex organizational technologies should seriously consider incorporating these factors to control for possible confounds.

With regard to methods, this study confirms the value of Assimilation Stage as an innovativeness measure, especially when the focal technology has not yet been widely adopted. If Time of Adoption or Dichotomous Adoption had been used instead, no gradations in innovativeness would have been

captured for as much as 90% of the responding population. Another contribution to research methods is confirmation of the value of the disk-based survey (DBS) mode. A survey as lengthy and complex as the one used in this research simply could not have been administered using traditional telephone or paper-based survey modes. Add to this the other advantages of DBS (minimization of item non-response and inadvertent coding errors; elimination of transcription costs, delay and errors) and the apparent positive effect on response rate, and one is left with a compelling case for the use the DBS mode where it is feasible to do so.

This study also has implications for technology vendors and mediating institutions, as well as end-users. For technology vendors and mediating institutions, this study has identified the profile of organizations more likely to initiate and sustain SPI assimilation—and in particular, OOPLs—thus providing the basis for development of more targeted marketing and promotion. Targeted market is likely to be of particular value for complex organizational technologies such as SPIs, because, as Attewell has argued, broad brush "signalling" of the existence and potential benefits of such technologies is likely to be of lesser importance in promoting adoption [1992]. Vendors and mediating institutions should rather be more focused on identifying appropriate adoption candidates, learning about the particular challenges these organizations face, and taking a more proactive role to promote successful assimilation among these sorts of organizations. For example, some OOPL vendors now have technology consultation divisions that supply long term, on-site assistance with OOPL-based systems development. Others have teamed up third party firms that supply so called "mentors"—technology experts that work side-by-side with end-users, with an explicit charter to teach organizations how to be successful with object technology. Participation in industry groups dedicated to technology promotion and standardization, such as the Object Management Group, represents a third approach to promoting an environment more conducive to successful SPI assimilation.

One of the main implications for end-user organizations is that, because of the combination of increasing returns and knowledge barriers, the common assumption that first movers always have the advantage may be seriously misplaced. Rather, potential adopters need to consciously assess the extent to which they fit the profile of an early and sustained assimilator. Those that do not fit the profile should consider delaying adoption, adopting a less complex variant of the technology, or going forward with adoption anyway, but with the understanding that risks are higher, and therefore expectation management will be especially crucial. Those that do fit the profile—other things equal—should be more vigorous in assessing emerging technologies in general, and when a decision to adopt is made, should undertake assimilation strategies that exploit their inherent opportunity for more cost effective and successful assimilation. These include employing relatively expensive strategies made feasible by Learning-Related Scale, such as hiring of experts, development of infrastructure, extended periods of "practice" on non-production systems; leveraging Related Knowledge in technology and project selection, and the assignment of personnel to projects; and exploiting the possibility that Diversity provides a "safe haven", a domain that represents an especially good fit with the focal technology and can more easily support initial organizational learning surrounding the technology.

Study 2: Alternative Measures of the Adopter Innovativeness Concept

The major implications of the analysis of alternative innovativeness measures revolves around the notion, first suggested by Downs and Mohr, that separate models of general innovativeness might be required for different innovativeness measures. For four of the measures—Time of Adoption, Dichotomous Adoption, Aggregated Adoption, and Assimilation Stage—this does not appear to be true, since the same basic patterns of explanatory factors emerged for all four measures. Or in other words, these four measures all appear to be tapping into a fairly unified notion of general innovativeness. This means that in choosing among these measures,

researchers can (and should) turn their attention to other considerations, such as the ability to support time-varying factors (Time of Adoption), the ability to support the adoption-decision design (Time of Adoption, Dichotomous Adoption, and Assimilation Stage), or the ability to support innovation-specific predictors (Time of Adoption, Dichotomous Adoption, and Assimilation Stage). Aggregated Adoption supports none of these design elements, although it does appear to have some other advantages. The models incorporating Aggregated Adoption had by far the highest variance explained, and had patterns of significant relationships that were very close to those predicted by theory. For research aimed at identifying more general predictors of innovativeness, Aggregated Adoption may well be an effective measure.

The other two measures—Diffusion and Infusion—had a very sparse pattern of significant explanatory factors. This may be because these measures tap into different dimensions of innovativeness that require different explanatory factors than those included in this study. Or it may be that the absence of expected relationships is due to the general difficulties involved with the study of post-adoption measures of innovativeness (smaller sample sizes, less variation in predictor variables).

Study 3: An Examination of SPI Assimilation Gaps

This study developed the assimilation gap concept, demonstrated that it can be sensibly measured, and showed that its measured size was consistent with *a priori* expectations for three prominent SPIs—RDBs, 4GLs and CASE. In verifying the existence of substantial assimilation gaps and identifying techniques for their study, this research has opened up a new domain of diffusion analysis, with implications for researchers and managers. For managers, the assimilation gap concept provides a new way to assess the prospects of a new technology comparatively early in the overall diffusion cycle. This study has shown that the pattern of cumulative acquisitions can present a largely illusory picture of the diffusion process. Based purely on

acquisitions, CASE would be viewed as the most robust of the three SPIs studied here, having diffused to nearly 50% penetration in only eight years. Yet the pattern of actual deployments paints a much different picture, one of a technology for which diffusion has just begun—in the sense of changing the actual practice of software development in end user organizations. Technology vendors and mediating institutions can incorporate these insights into their internal studies of the current state and future prospects of emerging technologies. End user organizations, while not likely to be conducting market research themselves, can profit from a more sophisticated understanding of what a report of strongly growing pattern of sales for an innovation says—and does not say—about its future prospects.

This study also has important implications for diffusion researchers. While it is well known that cumulative acquisitions need not be mirrored by cumulative deployments, the implications of this insight have not previously been incorporated into diffusion modeling studies. Prior studies have not modeled diffusion based on multiple definitions of adoption, or considered the difference between the diffusion patterns that might ensue. The results of this study suggest that when assimilation gaps are likely to be present, diffusion modeling studies should use adoption-as-deployment, either instead of, or in addition to, adoption-as-acquisition. In addition, the assimilation gap concept in itself provides a new area for study. The question of why some innovations might be rapidly acquired but sparsely deployed is in need of theoretical explanation. The explanation proposed here is that the expectations of future benefits, owing to increasing returns, drives rapid acquisition, but knowledge barriers impede deployment. There may be other viable explanations, however, that can account for this phenomenon.

7.3 FUTURE WORK

The implications of increasing returns and knowledge barriers, taken singly and in combination, open up a wide range of potential research topics; this thesis represents just first step in a broader research agenda focusing on the

assimilation and diffusion of SPIs, and other technologies sharing their distinctive characteristics. The remainder of this chapter will highlight three promising topic areas that go beyond, but still complement, the work already completed.

7.3.1 Potential Studies Building Directly on Dissertation Results

Several potential avenues for future work that build directly on the results of this dissertation were suggested in previous chapters. To recap, it was suggested (at the end of Chapter 3) that useful work building on the OOPL Assimilation results would be to: 1) replicate the results achieved with OOPLs using data on other technologies subject to knowledge barriers, and 2) establish more firmly whether or not Diversity has differently-directioned effects across different assimilation stages. At the end of Chapter 5, it was suggested that the empirical analysis of alternative innovativeness measures would be well complemented by a longitudinal study capable of establishing the reason for the observed difficulty in predicting Diffusion and Infusion.

The assimilation gap study reported in Chapter 6 represents perhaps the most fertile ground for immediate follow-on research. Only limited space was available in the DBS to capture data related to the analysis of assimilation gaps, because the primary focus of the dissertation as a whole was on identifying the antecedents of organizational innovativeness. As a result, there is considerable opportunity to perform a more detailed empirical study in this area. Such a study should include one or more of the following three elements:

- 1) A method for defining a "baseline" assimilation gap;
- 2) Inclusion of a larger set of SPIs;
- 3) More detailed measurement of variables needed to partition the organization-level outcomes associated with assimilation gaps.

1. A Method for Defining a Baseline Assimilation Gap

A method is needed to define a "baseline" for use in drawing conclusions about the absolute size of an observed assimilation gap. At present, event history analysis only provides a means for testing whether the survival processes (from acquisition to deployment) associated with two assimilation gaps are significantly different. However, if the survival process for a baseline assimilation gap could be established, then comparisons with this baseline could be used to draw conclusions such as "this observed assimilation gap is large enough to be substantively significant." One approach to defining a baseline would be to establish the distribution of survival times that informed observers would agree is reasonable for "ordinary" delay, and to use this to generate the survival data associated with a baseline assimilation gap. Another approach would be to use the actual survival times for the SPI in the study with the smallest assimilation gap, and let that serve as the baseline. However it is accomplished, the ability to draw conclusions about the absolute size of an assimilation gap would be a significant advance.

2. Inclusion of a Larger Set of Innovations

A second way to improve on the research already conducted would be to capture data on a larger set of innovations, including some for which the global diffusion process is closer to being complete. Including SPIs for which the global diffusion process is nearly complete would provide the data necessary to revisit use of traditional diffusion modeling techniques as a means for measuring and analyzing assimilation gaps. Including a greater number of innovations would provide the opportunity for enough comparisons to begin making a systematic link to theory. For example, expert judges might be used to measure the relative magnitude of the knowledge burden associated with the different SPIs. Then the predicted assimilation gap ranks implied by the expert judgments could be compared to the observed assimilation gap ranks.

3. More detailed Partitioning of the Organization-Level Outcomes

The third way to improve on the research already completed would be to gather data detailed enough to do a precise partitioning of the organization-level outcomes that contribute to the assimilation gap (i.e., "ordinary" lags, "excessive" delay, rejection, implementation failure, stalling, discontinuance). All of these outcomes have different theoretical and managerial implications. Organizations that experience "excessive" delay do nevertheless eventually deploy the technology, and in so doing, at least potentially derive net benefits from using the technology. At the other end of the spectrum, those experiencing implementation failure have no opportunity to recoup innovation costs related to the focal innovation, and may even discourage future innovation attempts. Furthermore, such organizations are more likely to become negative opinion leaders with the potential to have a considerable adverse impact on diffusion of the technology at large.

7.3.2 The Role of Supply Side Institutions

The focus of this thesis, and the first suggested new line of research, is on adopting organizations. However, as Attewell has clearly articulated, supply side institutions also play a crucial role in the diffusion of complex organizational technologies:

". . . supply side institutions have to innovate, not only in their design of products, but especially in the development of novel institutional mechanisms for reducing this knowledge burden or learning burden on end users." [1992, p. 15]

This suggests that a new line of complementary research would be to replicate Attewell's work for the case of SPI diffusion. It appears that the role of supply side institutions in the diffusion of object technology represents a rich area for potential study. Attewell highlights several kinds of supply side institutions, including service firms, consulting firms, standards organizations, and technology vendors. In the case of object technology, it appears that all four

kinds of institutions are indeed developing novel institutional mechanisms to reduce the knowledge burden associated with object technology.

A new class of service firm has emerged concurrently with object technology, which supplies "mentors" to facilitate the development of organizational competencies, rather than supplying traditional contractors to do the work themselves. Large consulting firms have taken a lead role in the adoption of object technology, and in particular, have devoted considerable resources to developing new methodologies to guide the process of developing software systems based on object technology. These methodologies can be viewed as repositories of collected object technology know-how, and while they cannot substitute for learning-by-using in adopting organizations—and indeed impose a substantial knowledge burden in their own right—they at least guide the process of learning-by-using, and provide a kind of scaffolding upon which tacit knowledge developed through learning-by-using can be hung.

In the area of standards, the Object Management Group (OMG) has emerged as a vigorous force for promoting object technology and lowering knowledge barriers. Although OMG is certainly not the first standards organization ever to be formed, it does have some unique elements in its charter. One of these elements is a strong commitment to facilitating learning about object technology via sponsorship of trade shows, conferences and technical seminars. A second element is the nature of the standards sponsorship process in the OMG. To be promoted by the OMG, a standard must be embodied in a commercially available product, the specifications for which must be made freely available to all OMG members. As a result, the OMG seeks to produce results that have an immediate positive impact on the infrastructure for object technology applications; this can be as lowering knowledge barriers for other vendors as well as end user organizations.

Finally, vendors have also been developing novel institutional mechanisms, that, whether intended or not, appear to be playing an important role in facilitating diffusion. OOPL vendors have developed internal divisions

supplying training and consulting services, not only regarding the use of their own products, but in the use object technology in general. Additionally, several large integrated IT hardware/software vendors have embraced object technology in developing software systems for their own internal use. They appear to have done this, at least in part, to establish competencies that will facilitate development of new IT products optimized to an environment where object technologies (so they believe) will be the rule, rather than the exception. This has had a spill-over effect, in that some of the most important learning-by-using has occurred in these organizations, and these organizations have freely shared the fruits of their gained wisdom with the broader IT community in conferences, and via published articles and books. (See, for example [Dietrich *et al.* 1989; Leathers 1990; Coleman and Hayes 1991; Rumbaugh *et al.* 1991]). Finally, the use of object technology, and in particular, OOPLs, has spread especially rapidly among independent software vendors as a tool to support development of their commercial software products [Vellante *et al.* 1992]. As with integrated IT vendors, this has a learning-related spill over effect to end users, but in addition, it can be viewed as contributing to economies of scale/learning-by-doing for OOPL vendors.

In sum, it appears that object technology presents a excellent opportunity for a detailed case study of institutional mechanisms for lowering knowledge barriers, and as a potential illustration of how supply side institutions might be able to provoke a sustained adoption bandwagon among end-user organizations for a technology that would otherwise be expected to have a stalled bandwagon, due to likely deployment difficulties among early adopters on the end-user side.

7.3.3 Comparative Case Studies of SPI Deployment

A third suggested line of future research is case-based research focusing on the extent to which assimilation strategies and tactics suggested by organizational learning-related factors are actually invoked by adopters, and with what

degree of success. The factors contained in the model proposed in Chapter 2 and tested in Chapter 3 are structural features of adopter organizations. While these factors are important, especially when predicting general innovativeness, it is likely that many other factors are also influential, particularly when the focus narrows to later assimilation stages. The general innovation literature suggests many such variables, including top management encouragement and support; technology sponsors and champions; and training and infrastructure support. Other more recent work has articulated the importance of appropriate implementation strategies [Leonard-Barton 1988b], and in particular, active steps to adapt both the organization and technology [Leonard-Barton 1988a]. However, in addition to these elements, there are other strategies and tactics suggested by organizational learning-related factors; these, for the most part, have not previously been studied by diffusion researchers.

Among such strategies and tactics are organizational investments in "rich" mechanisms for learning and knowledge transfer, such as using mentors or introducing extended periods of on-site "practice" with the technology. Another is the exploitation of diversity by identifying a "safe haven" for the technology, an area where the technology is a particularly good fit and the learning process can begin. Still other tactics that potentially facilitate organizational learning include "grafting" organizational knowledge via hiring [Huber 1991], "organizational prototyping," [Chew *et al.* 1991], learning-oriented joint ventures [Kogut 1988], "task partitioning" [von Hippel 1994], and inter-organizational networking [Pennings and Harianto 1992b].

Interesting research questions include: are these tactics in fact being widely employed? if so, how effectively? if not, why not? Are these tactics more likely to be employed by organizations that fit the profile of an early/sustained assimilator? In what circumstances are different tactics particularly effective, or even crucial? Such questions are difficult if not

impossible to answer with conventional survey methods, but rather, require the kind of richer investigation afforded by case study designs.

APPENDIX A. MINI CASE STUDIES OF EARLY ADOPTERS OF OBJECT TECHNOLOGIES

This appendix summarizes pertinent results from a set of "mini" case studies of early adopters of object technologies. The case studies were conducted between April 1992 and June 1993, and involved four organizations:

- 1) A US-based international energy company, oil exploration and production division ("EnergyCo E&P");
- 2) A retail bank based in the United Kingdom ("BankCo UK");
- 4) A US-based integrated financial services company, retail marketing services division ("FinancialCo RMS")
- 3) A UK-based brokerage division of the above mentioned financial services company ("BrokerCo UK");

The goals of the case studies were as follows: 1) to help assesses the face validity of assumptions and propositions from the literature regarding SPI assimilation; 2) to generate novel research questions and hypotheses; and 3) to gain a generally richer understanding of the assimilation process for object technologies from the perspective of industry participants.

The case studies spanned four organizations and numerous topic areas. It is clearly not possible, nor desirable, to attempt to cover all of the themes that emerged from the case studies here. Instead, this appendix will focus on two themes that are especially pertinent to the material presented elsewhere in this dissertation.

First, this appendix will use case study data to illustrate the nature of the knowledge burden associated with OOPL assimilation. A central assumption of this dissertation is that software process innovations in general and OOPLs in particular impose a substantial knowledge burden on would-be adopters. It is this assumption that motivated the use of Attewell's macro level theory of technology diffusion as a guiding framework in Chapter 1, and that motivated the decision to narrow the focus, in the variance model developed

and tested in Chapters 2 and 3, to organizational characteristics that should be associated with a greater propensity to innovate in the presence of knowledge barriers. The cases provide a rich array of support for contention that OOPL assimilation does in fact impose a substantial burden of organizational learning; they illustrate, for example, what such characteristics as "an abstract and demanding scientific base" really mean for actual adopters of OOPLs.

Second, this appendix will depict some of the diverse array of mechanisms employed by organizations to cope with, and with varying degrees of success, to overcome the knowledge burden imposed by OOPL assimilation. These strategies provide another window into the knowledge burden imposed by OOPLs, when one considers their cost, and the difficulties organizations encountered in employing them. Additionally, one can infer what types of organizations are positioned—in terms of various kinds of endowments—to be able to use these strategies, and by extension, to be more likely innovators.

The remainder of this appendix is organized as follows. A description of research methods is presented in Section A1, followed by a synopsis of the characteristics of each case study site in Section A2. Finally, the case analysis is provided in Section A3.

A1. RESEARCH METHODS

The mode of data collection at all case sites was the semi-structured interview, supported in some instances by a review of written documentation supplied by study participants. The main sources of guidance on developing and conducting semi-structured interviews were Patton [1990] and Oppenheim [1992].

The primary informants were IT managers and developers directly involved with activities related to object technology at the site. A total of 24 interview sessions were held, spanning 30 informants. Interview sessions ranged from 45 minutes to two hours, with most lasting about one hour (See Table A1).

Extensive notes were taken for all interviews. With the exception of the first round of interviews at EnergyCo E&P, all interview sessions were tape recorded.

Table A1: Interview Summary

	EnergyCo E&P (Round 1)	EnergyCo E&P (Round 2)	BankCo UK	FinancialCo RMS	BrokerCo UK
When	April 1992	June 1993	January 1993	June 1993	January 1993
Number of interview sessions	Two	Seven	Four	Six	Five
Average duration	2 hours	1 hour	1.5 hours	1 hour	1 hour
Informants	1 director 2 managers	1 director 2 managers 3 developers	2 executives 5 managers 2 staff con- sultants	1 director 2 managers 4 developers	1 director 1 manager 1 project mgr 2 developers

An interview guide was used to provide a general structure to the interviews. While the interview guide did evolve somewhat over the course of the case studies, the following basic topics were covered at all sites:

- 1) General company data, e.g., size, type of business.
- 2) IT department data, e.g., size, typical development methods and runtime technologies.
- 3) Data about a focal OO project, e.g., how initiated, kind of application, project team makeup, project management techniques.
- 4) OO adoption decision drivers, e.g., pre-existing organizational issues potentially addressed by OO, expected benefits.
- 5) Success factors present/absent, e.g., top management support, investment in learning and infrastructure, hiring practices.
- 6) Barriers present/absent, e.g., technical immaturity, learning curves, potentially unrealistic expectations.
- 7) Future plans and expected impacts, e.g., how OO might be used in the future, what organizational changes this might cause or require, and how these changes might be managed.

The typical interview session proceeded as follows. The general purpose of the interview was explained as being to help the interviewers learn about the assimilation process for object technologies. The topics to be covered were briefly summarized, and then the informants were questioned on each major topic, following a general-to-specific pattern. The initial questions on each topic were always open-ended. For many topics of interest, the interview guide also contained a checklist of potential prompts for issues/questions to raise should responses to more open-ended questions not provide adequate coverage of the topic. In point of fact, the prompts were rarely required.

A2. SYNOPSIS OF CASE SITES

A2.1 EnergyCo E&P

EnergyCo is an international diversified energy company headquartered in the United States. It employs over 30,000 people worldwide and had revenues of over \$30 billion in 1993. Its US operations are spread across three main subsidiaries: oil and gas exploration and production; oil and gas transportation; and oil and gas refining and marketing. The information technology function for US operations has a matrix structure. There is a centralized information technology department (corporate ITD) for the whole company, which is concerned primarily with issues related to IT strategy and standards. In addition, each subsidiary has its own dedicated information systems (IS) staff reporting both to the executive of that subsidiary, and to the corporate ITD chief.

The business unit covered in this case is the oil and gas exploration and producing subsidiary, referred to herein as EnergyCo E&P. In 1993, EnergyCo E&P employed 7000 people and had 1993 sales of \$1.7 billion. Two rounds of interviews were held with the IS staff of E&P, the first in April 1992, and the second 14 months later in June 1993.

The primary informants were the IS director in charge of applications for all of EnergyCo E&P, two applications development managers in charge of OO projects, and four application developers working on OO projects. The director and managers were interviewed during both rounds of interviews; the application developers were interviewed in just the second round.

The organization reporting to IS director (EnergyCo E&P Applications) is a professional staff of approximately 80 analysts, programmers and support personnel (such as database administrators). The total budget for EnergyCo E&P Applications was \$15 million for 1992, and about the same for 1993. Prior to the introduction of object technologies, applications were developed and implemented in a fairly typical commercial environment, using mainframes running IBM operating systems (e.g., MVS), IBM database management systems (e.g., IMS and DB2) and third generation languages (e.g., PL/1). Prior to the introduction of object technologies, the standard development methodology was based largely on the work of Ed Yourdon. This methodology was comprehensive but slow: a typical large application took five years to deliver.

The catalyst for EnergyCo E&P's interest in object technology was a restructuring of the EnergyCo E&P subsidiary initiated in 1988, and an associated business process reengineering effort. It was recognized that new information systems would be needed to support the reengineered business, and a consulting firm involved with reengineering strongly urged E&P to consider using state-of-the-art technologies to support such applications, and in particular, to consider object technologies. In response, EnergyCo E&P initiated a project in 1989 to evaluate five alternative environments, four of them based on conventional tools and one based on object technologies. In the end ObjectWorks, a Smalltalk language environment from ParcPlace, was selected as the target environment for all application redevelopment projects emerging from the business reengineering project. The ultimate technical architecture employed for OO applications was client-server, with client

applications running on Sun Unix workstations (for developers) and Microsoft Windows PCs (for users); Banyan Vines for local area connectivity; ParcPlace ObjectWorks (a Smalltalk development environment) and VisualWorks (a GUI development tool) for applications development; and IBM DB2 for data management.

Two large applications were selected as the first to be redeveloped to support the reengineered business: a Land Management System and a Gas Management System. In ensuing four years, from 1990 through 1993, well over \$10 million was spent on OO-related technologies and activities, with about half of this going to project-related activities, and the other half going towards development of "infrastructure" tools to support the OO applications development process.

The history and status (as of June 1993) of the Land System, the main focus of the case study, is summarized as follows. After three and a half years of development by a team of about six developers (on average), a portion of the original scope of Land the system was ready to be constructed. The development effort unfolded as follows: about one year was devoted to systems analysis and development of infrastructure tools. Next, about 18 months were spent on a failed attempt to implement the system based on an design that ultimately proved to be inadequate. The main problems with the design were that it was not sufficiently detailed, focused too much on data capture and not enough on data transformations, and failed to make key distinctions between different kinds of class and object relationships (i.e., general-to-specific relationship were not clearly distinguished from part-to-whole relationships). The last year was spent redesigning and developing a prototype for a portion of the system covering about one third of the original system scope.

As of June of 1993, the plan was to spend six months developing a production version of the portion of the Land System covered by the prototype, and then to implement it for a period of parallel operation with the existing Land

System. Needless to say, the development of the Land System was viewed internally as quite a struggle. The Gas System project involved some similar difficulties, but fared a bit better. After an expenditure of 20 person years of development effort spanning three calendar years, implementation of about 35% of the system was scheduled to go live on August 1, 1993, after six months of parallel operation with the existing Gas system.

Even given the challenges of assimilating the OO approach to systems development, EnergyCo E&P remains firmly committed to using OO technology. Although there are no current plans to aggressively redevelop the installed base of applications, OO technology, and in particular, Smalltalk, is still considered the technology of choice for all major new development projects.

A2.2 BankCo UK

BankCo UK is one of the main four large UK national clearing banks. Its primary business is in retail banking, and it operates over 3,000 branch offices. In total, BankCo UK employs about 100,000 people. BankCo UK's IT function is a centralized group of about 4,000. About one third of this staff dedicated to systems development (ITSD), one third to supporting networks and distributed systems (including automated teller machines), and the final third to centralized computer operations. A separate group of about 25 professional is involved with IT strategy and planning (ITS&P). The total IT budget for 1993 was £400 million.

The focus of the interviewing at BankCo UK was two-fold: to learn the particulars of a recent completed OO pilot project and to learn about the strategic role foreseen for object technology for the IT function as a whole. Four main interview sessions were held in January 1993, three of which involving multiple informants. The pilot project was jointly sponsored by ITSD and ITS&P. Two of the interview sessions were held with ITSD staff, and two with ITS&P staff.

One of the primary reasons for BankCo UK's interest in new development technologies in general, and object technologies in particular, is a goal of replacing all branch-based systems with modern workstations in a client-server architecture within five years. This, in turn, is but one component of a larger organizational thrust towards stronger "relationship" banking, where the focus of banking activities is at the customer level, rather than at the account level. BankCo UK sees this capability as essential in the future to support, among other things, the timely introduction of new products.

The OO pilot project was intended to provide an evaluation of the OO approach to systems development. The project was initiated in March of 1992 and finished eight months later in November. The total budget for the project was £400,000 of which £250,000 was for salaries of a staff of five. A total of about four person-years of development time were devoted to the project. The project was staffed with internal personnel, and an external "mentor" who was used to help establish management procedures and to facilitate learning about OO concepts and techniques.

The main application developed as part of the pilot project was a system called the Business Process Manager (BPM), to support the role-out of new business procedures in one of the branch offices. One of the ongoing problems in the branches is the difficulty in establishing new procedures in a standardized way. The purpose of the BPM is to allow a non-technical person to design a new business procedure and then implement it. The result is an automated process that takes branch personnel step-by-step through the new procedure. As a result, the system is not a traditional transaction processing style system, but rather, more like a process control system for administrative personnel.

The original BPM was never implemented in intended branch office due to technical difficulties considered to be unrelated to OO. The branch office wanted the BPM to operate under MS Windows, but the network they were using did not support Windows. Nevertheless, a slightly revised version of

the BPM was developed and implemented within the IT function itself, to assist with the process of tracking mortgage applications being processed on behalf of the IT staff.

Having only completed a pilot project conducted primarily for evaluation purposes, BankCo UK does not view itself as an "early adopter" of OO technology. It is, however, actively interested in OO technology, and has other OO thrusts in progress or planned. BankCo UK has sponsored and taken an active interest in general research on object technology as part of an industry consortium called the Object Interest Group (OIG). OO evaluation efforts by groups involved with data architecture and business process reengineering are also currently underway.

A2.3 FinancialCo RMS

FinancialCo is one of the world's leading financial services firms, with divisions providing brokerage, mutual funds, banking, life insurance, credit cards and institutional investments. Founded in the United States, FinancialCo also has operations in Europe, Asia, Latin America and the Middle East. In 1993, FinancialCo employed over 4,000 people and managed over \$130 billion in investments. In both the United States and overseas, the many divisions of FinancialCo are decentralized, with each division taking responsibility for bottom line profits.

The main IT function for FinancialCo's US-based operations is organized as a separate business supplying services to other internal divisions, and occasionally, to outside organizations. The IT division has over 400 employees. In addition, some of the operational subsidiaries have their own IT staffs. One such division is Retail Marketing Services (FinancialCo RMS).

Interviews were held with seven members of the FinancialCo RMS IT staff in June 1993 to discuss current OO projects. Informants included a senior executive in the division, two managers over OO projects and four

developers. The main topics covered in the interviews were a recently completed OO project, the history OO development within FinancialCo, and expectations about the future role of OO.

Up until the mid 1980's, the bulk of applications at FinancialCo were developed on mainframes, with third generation languages such as COBOL. In the mid 1980's a new direction was set towards the client-server architecture, and over a dozen so-called "workstation" development projects were initiated, many of them based on MS/Windows and the C language. In 1990, an executive in RMS spearheaded FinancialCo's first foray into object-oriented development with the development of a system to support processing of correspondence from credit card customers. This executive had previously come to the conclusion that client-server was the right architecture, but viewed conventional tools like C as inadequate to rapidly develop robust, maintainable, and extensible systems. The RMS executive hired a systems developer/consultant with extensive OO-related academic training and industry experience, who then assembled the rest of the staff of eight developers. All developers were from outside FinancialCo, and most had prior OO experience. The RMS executive viewed the resulting system as a success from a technical standpoint—it was constructed rapidly, and provided capabilities well beyond any existing client-server systems in FinancialCo. It also served to spark interest in object technologies throughout FinancialCo. Yet, the system was developed outside the main IT division, and political infighting over ongoing control of the system between RMS and the IT division, led, in the RMS executive's opinion, to the system's ultimate failure.

In 1992, after a year working for another division, the RMS executive returned to the RMS division, reassembled several of the original members of the credit card correspondence processing system staff, and initiated a new OO project to develop a mass mail system and a marketing decision support

system (herein referred as the Mass Mail/DSS system). Both were to be supported by information in a large customer database.

The team developing the Mass Mail/DSS system was composed of two working managers from the original credit card project with extensive OO experience (from prior work both within and outside FinancialCo), four of the developers from the credit card project, and three new hires. All of the new hires had prior experience with developing workstation applications; most had at least some prior experience with OO development.

The original technical architecture for the system was MS/Windows and C++ on the workstation side, and a relational database on the server side, although a strategic decision was made to port the systems to more powerful NeXT workstations and Objective C after the system was nearly completed, in June of 1992.

The initial version of the system took just eight months, from February through September 1992. The RMS executive viewed the Mass Mail/DSS system develop as a complete success. The original goals for the development—robust functionality, rapid development, and high levels of reuse—were all met. The initial system was developed in six months, and the staff was able to reuse as much a 85% of the original code when the decision was made to port the system to the NeXT platform. As of the time of the interviews, the system had been in production use for several months, some enhancements had been completed, and an additional series of enhancements were planned or underway.

Although the RMS executive is committed to using object technology for all major new projects in his area, he is unsure of the long term role for object technology within FinancialCo at large. He believes there is a great deal of support for the OO approach at the grass-roots level, but at the top levels of the IT division support has been mixed, and in his opinion, reflects a lack of genuine understanding of what OO development is all about.

A2.4 BrokerCo UK

BrokerCo UK is a brokerage firm serving investors primarily in the United Kingdom. It also has branch offices in Europe and Asia. It is a subsidiary of FinancialCo described above, and, as with other subsidiaries, operates with a great deal of autonomy. BrokerCo UK employs about 1000 people, making it the smallest of the four case study sites. The IT function is a centralized group located in the greater London area.

Five interviews were held with IT staff in January 1993. The primary informants were the director in charge of systems development, the manager in charge of OO projects, and three developers working on OO projects.

The systems development organization has a staff of 50 professionals, and a yearly budget of £3.2 million for personnel-related costs. The primary technical environment for the installed base of applications is the IBM A/S 400. BrokerCo UK traditionally relied very heavily on purchased packages; for those custom systems that were developed, traditional development practices were employed.

Two events lead to BrokerCo UK's eventual adoption of object technology. The first event was the arrival of a new systems development (SD) director in the summer of 1991, from the United States. The new SD Director's immediate four or five predecessors all were viewed as ineffective and had short tenures. The SD Director arrived to an environment with a skeptical user base and a demoralized development staff. No significant systems had been delivered in years. The SD Director was given a mandate to make whatever changes were necessary to create a more effective and responsive development organization. He had previous experience with open systems and client-server technology, but not object technologies. He believed that, as a precondition for moving forward, the IT function would have move to modern platforms employing open, client-server architectures.

The second event leading to adoption of OO was a visit, also in the summer of 1991, of the two principle players in FinancialCo's OO activities, i.e., the RMS Executive who had initiated the use of OO at FinancialCo, and the experienced OO developer who had assembled the original credit card processing team. Together, they convinced the SD Director that, given he wanted to move to a client-server approach, he should strongly consider using object technologies. After extended discussions, the SD Director agreed that object technology was indeed the best way to proceed, and set about assembling a staff to develop a new set of applications based on object technologies. The SD Director viewed the adoption of OO as just one of many changes towards developing a more effective systems development operation.

The systems architecture settled upon for OO projects had three tiers: PCs running MS/Windows for the users (tier 1), connected to Sun workstation servers running Sybase (tier 2), ultimately connected to A/S 400 serving as a back-end repository (tier 3). The primary object technologies employed were the C++ language and a class library initially developed by FinancialCo RMS in the US.

The SD Director selected an internal manager with prior PC application development experience to manage the effort to develop OO applications. The first application to be developed was a Brokerage System to support the daily activities of retail brokers. The initial staff for the Brokerage system consisted of six professionals, all brought in from outside BrokerCo UK. Three were brought in from FinancialCo sites in the United States, and three were hired in the UK. All new staff either had prior experience developing OO applications using C++, or had some other specific technical skill related to the technical architecture (e.g., Sybase experience). As a result, there was very little need for technical training.

The elapsed time for development of the Brokerage System was about one year; about six person-years were required for the initial version of the

system. One developer was devoted full time to maintaining and enhancing a "tool kit" providing classes to support basic system functions (data structures, screen objects). This toolkit had been inherited from FinancialCo RMS, but was substantially revised and enhanced.

The SD Director viewed the Brokerage System as quite successful. At the time of the interviews, the system was in every day use by five brokers, with a volume of 500 transactions a day. Although some schedule slippage occurred, the SD Director stated that this was mostly expected, due to his prior decision to set unrealistic deadlines (as a motivational device). Another prime contributor to slippage was the decision to skip a formal analysis phase and to rely entirely on rapid prototyping for requirements analysis. Because of a lack of formal analysis, gaps in system functionality were not uncovered until the user testing phase. As a result of this, a formal analysis phase is to be included on future OO projects.

The SD Director was uncertain about the long term role for object technologies in BrokerCo UK. Object technologies were being used on all new, workstation-based applications, but no move had been initiated to use object-oriented techniques on the A/S 400, or to redevelop A/S 400 applications using OO. The SD Director was pleased with results achieved with object technology, but said that the future role for OO would depend on whether the expected benefits of the object-oriented approach were obtained on the next two or three applications. Expected benefits included: 1) rapid development and implementation of new systems, 2) substantial reuse of classes already developed for the Brokerage System, and 3) lower systems maintenance costs than for the A/S 400 environment.

A2.5 Summary

A summary of characteristics for the four case sites is provided in Table A2 below.

Table A2: Case Site Characteristics

	ENERGYCO E&P	BANKCO UK	FINANCIALCO RMS	BROKERCO UK
Description of parent company	US-based international energy company	UK-based retail banking firm (one of four largest)	US-based international financial services company	US-based international financial services company
Business unit Interviewed	IT organization for Exploration and Production division	Corporate IT department	IT organization for Retail Marketing Services division	IT organization for UK-based brokerage division
Total employment	7000 employees (E&P division)	100,000 employees (company)	Not available	1500 employees (UK brokerage division)
IT Development budget	\$15 million (division)	£ 400 million (company)	Not available	£ 3.2 million (division)
IT development staff	80 (division)	1,000 + (company)	75 (division)	50 (division)
Start date for OO activities	Summer 1990	March 1992	January 1990	Summer 1991
Main languages and tools	ParcPlace Smalltalk and VisualWorks, internally developed tools, Unix, Windows	Intellicorp Kappa PC, DOS/Windows	C++, Windows, Objective C, NeXT Step, internally developed tools	C++, Windows, and internally developed tools
Status as of last contact (Date of last contact)	Implementation of a large production system imminent (June 1993)	Initial pilot/evaluation completed (January 1993)	Multiple production projects completed (June 1993)	One production project completed (January 1993)
Expected future usage	For all significant new development within division	Not sure; several evaluation projects ongoing	For all significant new development within division	For most significant new development within division

A3. CASE STUDY ANALYSIS: MAJOR THEMES

This section presents a description and analysis of two interrelated themes emerging from the case studies, the first being the nature of the knowledge burden imposed by OOPL assimilation, and the second being the kinds of strategies invoked to overcome this burden.

A3.1 Nature of the Knowledge Burden Imposed by OOPL Assimilation

i) A Radical Shift

Although a few dissenting voices have asserted that object orientation is no more than a repackaging of well known software engineering principles [Page-Jones and Weiss 1989], most OO researchers and advocates hold that OO requires a radical mental shift on the part of developers schooled in traditional methods [Coad and Yourdon 1991; Fichman and Kemerer 1992; Booch 1994].¹ This latter view is supported by the case data at all four sites, but is perhaps best illustrated by the experiences of EnergyCo E&P and BankCo UK, the two sites that relied primarily on existing developers with no prior OO experience.

At EnergyCo E&P, a team of fifteen developers was assembled from existing staff to work on OO projects. All developers were sent through a "six week indoctrination period" consisting of a two weeks of off-site introductory training, followed by four weeks on on-site advanced training. According to one OO development manager:

"Everyone would be working with Smalltalk for a couple of months and then have a religious experience, where the light bulb goes on . . . you have this tremendous awakening—like a curtain parting" [Manager, EnergyCo E&P].

¹As one recent commentator recently put it: "object oriented technology is so radically different that we must break and reset programmers' coding hands before they can write object oriented rather than procedural programs" [Lewis 1994].

At BankCo UK, a much smaller scale initial development project was attempted, but nevertheless, learning OO principles and how to apply them was a struggle:

"It's very difficult for people to get a handle on what object orientation is all about—because it is so different—until they actually use it and then it takes time to really understand what they are doing." [Manager, BankCo UK]

Interestingly, respondents were divided about what kinds of prior experience and knowledge best positioned developers new to OO to be able to grasp its basic principles. One manager at FinancialCo RMS emphasized that system programmers with an assembly language background had an advantage because they were accustomed to working with computer science abstractions. This manager, and others working at FinancialCo RMS also mentioned that because C++ programming involves considerable use of pointers, prior experience with pointers, such as one gets programming in assembly language or C, was a key advantage. One of the more surprising observations at FinancialCo RMS was that developers with strong musical ability tended to learn OO more rapidly.

At EnergyCo E&P, the prevailing opinion among managers was that there was nothing in particular that could be identified in advance, except for general technical ability:

"As far as ease of adoption, it did not really matter what a person's background was or how much experience they had; however good they were before, that's how good they were with Smalltalk" [Manager, EnergyCo E&P].

Managers at BankCo UK likewise were unable to specify exactly what kind of background or characteristics predisposed developers to more easily grasp OO concepts. Experience with data modeling was mentioned as plus, because of the similarity between data entities and objects in the real world. In one interesting exchange, extensive prior experience in some domains, and in particular, structured programming, was cited as a disadvantage because of the tendency for programmers to revert to old habits:

"It seems that if people have gotten heavily into structured programming, when they learn OO they will always try to escape back into doing data and programs separately, in the old way." [Manager, BankCo UK]

A manager at FinancialCo RMS echoed a similar sentiment, but with a different rationale. He concurred that those who were most proficient in the old way of doing things had particular trouble making the transition to OO. When asked whether this was due intellectual or motivational barriers, he replied:

"I think it's probably a motivational barrier. It turns out that part of the transition is: 'I'm king of the hill here, I know every single byte in the system—now I don't know any of the bytes in the system, and not only that, but you're also telling me that all the stuff that I have learned the last 15 years doesn't apply anymore.'" [Manager, FinancialCo RMS]

Another FinancialCo RMS manager expressed the same basic sentiment, albeit a bit more colorfully:

"A good C programmer does not [automatically] turn into a good C++ programmer. A structured analyst is not necessarily going to make a good OO analyst. And a hot-sh** Cobol programmer is likely to become unbelievably demotivated when he is a lousy C++ programmer for the first year." [Executive, FinancialCo RMS]

Yet a complete lack of programming experience was also viewed as damaging. When asked to explain why one staff member had a particularly difficult time picking up object-oriented programming, a manager at BankCo UK replied:

"We are not sure what the problem was. Partially he has been slow to understand OO itself because he has not had the discipline of writing programs...Although OO is very different, when you get down to the very detailed level—the person who writes the methods—within that method is code, procedural code in very small chunks." [Manager, BankCo UK]

In summary, although the case sites differed considerably in terms of OO technologies adopted, prior technologies in place, time of adoption, and overall adoption strategy, the general theme that OO is a radical, competency destroying change nevertheless emerges from almost every quote about the mental shift required by OO.

ii) The Need for Learning-by-Doing

To become skilled in the application of OOPLs developers must master the novel principles of encapsulation, inheritance, polymorphism and others, and thus, object-oriented programming does appear to have the kind of "abstract and demanding scientific base" that Attewell notes is indicative of a technology subject to knowledge barriers [Attewell 1992]. As one informant at FinancialCo RMS put it: "knowing what the words [OO principles] are" and "knowing what they mean" are two different things. This latter kind of knowledge requires an extended period of learning-by-doing. It was common for respondents to remark that, although classroom training and book learning was important, it was not until considerable hands-on experience, typically over a period of months, that true understanding of OO principles was achieved. One developer at EnergyCo E&P explained how a core group of programmers who had spend many months engaged in an intensive effort to develop infrastructure tools "were the only set of people who gained a very in-depth understanding of Smalltalk". A similar point was expressed by other developers at EnergyCo E&P and FinancialCo RMS:

"I've found that being able to 'think in objects' is really the critical step that came from coding and having a mentor there to walk through things. It's a shift you make at some point." [Developer, EnergyCo E&P]

"Actually, it wasn't until I was forced to work in a group and write code and do all this stuff that I really started to understand how OO works. I thought at the time that I had a good idea of what was going on, but I really didn't." [Developer, FinancialCo RMS]

The need for learning-by-doing is a key element of the knowledge burden imposed by OO, because it means that unless experienced OO developers are hired from outside, learning is necessarily going to be an expensive, time consuming, and unpredictable process. At both EnergyCo E&P and BankCo UK, managers felt that with the on-site assistance of one or two "mentors" to guide a team of novice OO programmers and to look over their work, this burden was ultimately manageable. At FinancialCo and BrokerCo UK, a different opinion was expressed. Managers at both organizations felt that a

team capable of developing large-scale OO applications could only be built top-down, by hiring in OO developers with years of experience. With the primary burden of development being carried by experienced personnel, they said it might be possible to place novice OO programmers in minor roles, and "grow" them into proficient developers over a period of many months. When asked to comment on the viability of a strategy similar to the one employed at EnergyCo E&P, where a team of novice OO programmers carried the primary burden of development, a manager at FinancialCo RMS replied:

"I think long term if they had managers that were committed to it and that were going to make the people do it, then they would have some chance. But I think that is a recipe for frustration. If you are going to do it that way, then you better be committed [enough] to have your first couple of projects fail. Or your first couple of projects show no net gain. [Managers would have to] say, there's this long term plan to stick with it for two or three years. Back in my consulting days we came in on the second wave of lots of project like that. At least management was committed enough to know it was the process [that was at fault]." [Manager, FinancialCo RMS]

Interestingly, this description of what might befall the hypothetical organization relying on existing staff is fairly similar to the actual experiences of EnergyCo E&P. At EnergyCo E&P, assimilation of OO was an ongoing struggle, yet because of a strong managerial commitment to see the process through, there was still the expectation that assimilation would ultimately be successful.

iii) Knowledge of Related Technologies

As argued above, OO development imposes a knowledge burden, not only due to the radicalness of its basic principles, but also because of the need for learning-by-doing on the part of novice developers. For some developers, this burden is further compounded by a lack of experience with related technologies typically used on OO development projects. This was a particular problem at EnergyCo E&P. As explained by one of the OO development project managers:

"[Learning how to be an OO developer] was a series of hurdles. Some people had the hurdles of never having done anything on workstations, they'd never used a mouse, they'd never done anything in a graphical environment. I went to college late enough where I had that experience in college. I was on somewhat familiar ground. Other people seemed to have barriers..." [Project Manager, EnergyCo E&P]

At FinancialCo RMS and BrokerCo UK, this additional knowledge burden was avoided by almost exclusively employing people on OO projects that were at least proficient in surrounding technologies, such as the C language, GUI development, and workstation based development. At BankCo UK, this problem was largely avoided in another way: they used a high level OO development language that insulated developers from many of the technological details of the underlying platform. The apparent benefits—and costs—of this strategy will be discussed in much greater detail below, but for now it is worth noting this illustrates how technology simplification can lower knowledge barriers over time [Attewell 1992].

iv) Immaturity of the Technological Network for OO

At all four case study sites the costs of adoption, and in particular learning-related costs, were magnified considerably by the general immaturity of the technological network for OO. As noted by Fichman and Kemerer, the benefits of a mature technological network for a software process innovation—e.g., greater availability complementary tools, trained personnel and general industry wisdom—accrue only over time, after many organizations have adopted an innovation [Fichman and Kemerer 1993].

Some researchers have argued that many organizations will be reluctant to join an immature network, for fear they will not be followed, and therefore a mature network will never develop [Farrell and Saloner 1987]. Organizations must be followed for adoption to be worthwhile, because, as argued in Chapter 1, in the case of SPIs the inherent knowledge burden compounded by the other costs imposed by an immature network serve to lower the net performance most organizations are likely to achieve to levels below current

best practices. Support for these ideas can be found in statements by managers at two of the case sites:

"Initially progress has been slower than it probably would have been with some of the other [more mature] technical options evaluated early."
[Manager, EnergyCo E&P]

"[Internal critics at FinancialCo] go out and read articles and say show us the 4 to 1 development improvement, or the 10 to 1, or what ever. But there were a couple of pieces missing. There were no good class libraries, we were writing our own substantially for what we were doing. There were no real benefits in productivity to doing OO development unless you have some class libraries. It's fun and all but it isn't real productive until you have a code base that you can work on top of. And it hurt." [Executive, FinancialCo RMS]

This is not to say that it is always better to wait, but organizations that do join an immature network must have the capacity to weather an extended period using a technology with, at best, uncertain net benefits. Such companies must have the resources to either hire in top talent, or to support a robust internal training program. They must be willing to risk adoption of a technology that might have to be ultimately abandoned if it never does become widely adopted by others. They must be willing and able to develop their own complementary tools (or to do without) and to contend with the immaturity of initial vintages of the core technology.

One of the more interesting comments emerging from the cases was the extent to which respondents from EnergyCo E&P were (unpleasantly) surprised by the immaturity of OO tools:

"OO is advertised as a fast development tool, and as being more mature than it really is. Initially the learning curve was very steep." [Manager, EnergyCo E&P]

"Well I think that [our adoption decision process] was a very simple, very naive—that there was this new technology that was out there that encompassed function and data, and that modeled things as they were in the real world. And we kind of got the impression that there we a lot of reusable components already available to us at the time. And that part of it turned out to be false" [Developer, EnergyCo E&P]

Managers at both FinancialCo RMS and BrokerCo UK, while not surprised at the immaturity of the OO network, nevertheless did complain about it:

"I'm a senior VP of financial services and I have 70 or 80 people in this group. The fact that I know that we are in another Windows class library evaluation—its like, why? How is that possibly in [FinancialCo's] best interest? And NT compatibility. I don't want to think about this stuff. It's a waste of time." [Executive, FinancialCo RMS]

"Also the toolkit—that's not an area where we excel, and there are probably 200 vendors that could do a better job of maintaining and enhancing it...So we are really well out in front of the industry; [this] puts us in the position of doing things that do not add value to the business; at the end of the day people, our clients, don't care about our toolkit" [Manager, BrokerCo UK]

Three case sites—EnergyCo E&P, FinancialCo RMS, and BrokerCo UK—all devoted considerable resources to developing complementary tools that they expected would one day be available commercially on the market. FinancialCo RMS and BrokerCo UK both developed tools to support the development of graphical user interfaces, and to support integration with other technologies, such as relational database management systems. EnergyCo E&P made a far greater investments in custom tools. In order to make Smalltalk viable for large-scale development, EnergyCo E&P developed, at a cost of millions of dollars, a portfolio of five "infrastructure tools": 1) a CASE-like design modeling tool, 2) a graphical user interface development support tool, 3) a relational database schema generator, 4) a tool to manage a dynamic interface between Smalltalk and a relational database, and 5) an ad-hoc query support tool. The latter three tools were all required owing to the perceived absence of an industrial quality object-oriented database management system:

"The OO databases we are familiar with can not support the number of concurrent users that we need. And generally their stability is, as far as we know, just not acceptable yet for this critical of an application. The intent is that as they become better then we would move towards that." [Developer, EnergyCo E&P]

It is worth noting that when the original adoption decision was made at EnergyCo E&P, it does not appear that the additional costs due to an absence

of complimentary tools were explicitly considered.² In the end, four of five of these tools were discarded, some because the need for the tool was obviated by other changes to the technical approach, and others due to the subsequent arrival of commercial products performing the same functions. As one manager put it:

"[EnergyCo E&P] has a position that we are in the business to build applications, not tools. Our management does not want to pay for the maintenance of those tools" [Manager, EnergyCo E&P]

The costs of being part of an immature network were not confined only to a lack of complementary tools. All organizations remarked on the difficulty of finding experienced staff. When asked how one goes about hiring experience personnel, a manager at FinancialCo RMS explained:

"Partially it's luck. You have to get a core team in place. You need to have at least one person you trust, hire some people they select, and build connections from there...We have new people talk to every one in the group... Tell me about a class library you have designed. An extension you have made to a class library. Tell me about class library design. Tell me about some class libraries you have used and stress the differences between them." [Manager, FinancialCo RMS]

An interesting observation here it appears that considerable OO expertise is needed if for an organization to be able to identify candidates that themselves possess genuine OO expertise. Managers at BankCo UK, for example, said they wanted to hire experienced staff, but simply could not find any:

"Because of the nature of OO we were very keen on getting someone externally, but we could not recruit anyone because it was such a new technology" [Manager, BankCo UK]

In contrast, BrokerCo—located in the same geographical area as BankCo—was able to locate experienced developers, through the assistance of OO experts from FinancialCo RMS.

²The original plan was to use an OO database management system from Servio Corp. to support data management, which would have forestalled the need for three of the tools. The need for a fourth tool, to support GUI construction, became apparent only after EnergyCo E&P was well into the assimilation process.

Even the organizations that used experienced OO developers (FinancialCo RMS and BrokerCo UK) suffered because of a general lack of acquired industry wisdom to draw on about how best to manage the process of OO development. A particular challenge at BrokerCo UK was reconciling two purported benefits of OO that they eventually came to realize were initially in conflict: rapid development cycle times and developing components for reuse. BrokerCo UK learned the hard way that even though a formal analysis phase adds considerably to cycle times, it is necessary if significant levels of reuse are to be achieved later:

"One problem was that we had not gone through an effective object-oriented analysis [OOA] effort, due to time pressures and due to the fact that guys using the technology were still in the learning process." [Manager, BrokerCo UK]

"Because no OOA was performed [in the first application] there is little reuse...There's a tradeoff; reuse is good but comes at a cost. Now we see the costs of getting to a true OO environment: doing a global business model, including future business plans" [Manager, BrokerCo UK]

But finally realizing that this issue exists does not make gathering the organizational will and resources to resolve it any easier:

"[We] don't have luxury to take three or four people now to figure out how to do code sharing and code reusability" [Project Manager, BrokerCo UK]

EnergyCo E&P also experienced considerable difficulties because of the lack of established methodologies to provide guidance on the development process. The first of the major systems developed at EnergyCo E&P had to be redesigned and redeveloped after the programming staff belatedly discovered that the design model they had been working from was produced from an idiosyncratic and incomplete technique, and simply could not be implemented as it had been designed.

v) The Looming Organizational Transformation

Although OO has become the technology of choice for significant new development projects at three of four case sites (all but BankCo UK), none of

the sites have reached the point where OO is the standard or dominant approach in their respective organizations. The concerns that these managers expressed about the prospect of making the transition to OO as a dominant approach provides yet another view of the burden of organizational learning imposed by the assimilation of OO technology.

BankCo UK was the least far along in the assimilation process, having only completed a small pilot project. Not surprisingly, managers at this site expressed the most uncertainty about long-term transition issues:

"First, there are not many proven cases on large-scale systems; there's no question it works with a small, tightly-knit group....We are also concerned with the change in culture: Can you evolve towards OO, or is a revolution needed? Third, management is skeptical that OO is just another new oversold technology, such as DBMS, Expert Systems, CASE. We are also concerned about control of developers, e.g., code libraries, repositories, tool support; how do you find objects in the repository? Our last concern is that OO [might prove to] be a technology for new development only." [Manager, BankCo UK]

Having completed part of one large system, and struggled for years to complete another, a manager at EnergyCo E&P expressed the current major concerns of his unit's management team this way:

"We have three current concerns about OO: how to communicate the true benefits of OO to an executive audience, the costs of learning a new technology, the lack of tools". [Manager, EnergyCo E&P]

It is interesting to note that learning-related issues still feature prominently in current concerns at EnergyCo E&P, more than four years after the original adoption decision was made. Another persistent barrier has been the ongoing costs of having to make up for a lack of an acceptable object-oriented database management system:

"...I think we need to proceed slowly until there is a better alternative to DB2 as a repository. I'm not at all satisfied with what we are having to do with that Smalltalk-DB2 connection to get the response that we need." [Director, EnergyCo E&P]

At BrokerCo UK, the installed base of applications on the A/S 400, and the people who support them is seen as one of the a main hurdles to organization wide assimilation of OO:

"The cost of retraining is going to be a big, big issue; because you can't just throw away the knowledge people have of the existing business systems."
[Manager, BrokerCo UK]

"Is it better to hire fresh people, untainted by 18 years of whatever, or can you really retrain people? Because it's not [as easy as] learning C—we can teach everyone here how to program in C, it's the way of thinking [about objects that's hard]. So that poses very unique challenges for training and frankly not everyone is going to be a candidate. [Executive, BrokerCo UK]

FinancialCo RMS was also concerned with the looming task of retraining existing staff, but appeared more optimistic about a natural process of attrition smoothing in the process:

"And honestly, a lot of people are not going to make it [the transition to OO]. Maybe 50 percent. But this is OK since the process is slow; and you need people to maintain the legacy systems." [FinancialCo RMS]

A3.2 Assimilation Strategies

The previous section illustrated the magnitude of the burden of organizational learning associated with OOPL adoption and use; this section considers the strategies employed to manage this burden. Among the four case sites, three distinct strategies were evident. At EnergyCo E&P, the primary strategy was to invest heavily in various mechanisms for facilitating learning by existing staff. At BankCo UK, this burden was managed by adopting a simpler technology variant, i.e., a high level OOPL. Finally, at FinancialCo RMS and BrokerCo UK, the burden of organizational learning was managed by aggressive hiring of experienced technical personnel, i.e., external staff with extensive prior OO experience were "grafted" onto the organization. These three strategies are described and discussed in the subsections below.

i) Heavy Investments in Existing Staff

Early in the assimilation process, EnergyCo E&P made a conscious decision to staff OO development projects with existing personnel, rather than hiring in expertise from outside. Although two "mentors" were employed for several months, and did do some limited development work themselves, the vast majority of the system development effort was conducted by a staff of fifteen internal employees that had been retrained in OO techniques.

EnergyCo E&P provides an exemplary example of an organization willing to invest any necessary sums in organizational learning and establishing a robust infrastructure. In many organizations, no formal training in new techniques is provided, and developers are expected to learn "on the job." At EnergyCo E&P, by contrast, staff members were given six full weeks of training. Moreover, EnergyCo E&P was willing to hire, at considerable expense, the aforementioned "mentors" to guide and review of the efforts of internal personnel, thus providing an ongoing mechanism to facilitate training and experiential learning. EnergyCo E&P also invested in the development of infrastructure tools that were intended, in part, to simplify the process of systems development according to OO principles. And lastly, although it was not the primary intention of managers, the development of the infrastructure tools did in fact serve as an extended training ground for novice developers.

Ultimately, however, even extraordinary measures such as these appear to be insufficient to guarantee success on initial development efforts such as those undertaken at EnergyCo E&P. This is because crucial gaps in knowledge are almost inevitable whenever a large-scale project, using a radically new technology, is staffed primarily with people that have never previously seen a project through from start to finish. The inability to recognize that the original design model for the Land Management system was not completely object-oriented, and could not be implemented as it was, led to over a year of rework:

"Before September of 1992 I think what did not work out well is we did not have a clear understanding of what exactly we were supposed to get out of this [the design] model, what we were supposed to implement. We had a brand new technology, a brand new modeling technique and there was no single point of understanding for the whole thing." [Developer, EnergyCo E&P].

"When Smalltalk was first picked we had hoped that the technology and methodology would come together, but the biggest problem was that the methodology people did not know enough about the technology and the technology people did not know enough about the methodology to make those two things meet." [Developer, EnergyCo E&P].

In a less patient organization, the need for rework on this scale might have been grounds to abandon the project—and with it the OO approach. Ongoing problems with the linkage between Smalltalk and DB2 provide another illustration of a crucial gap in knowledge at EnergyCo E&P. The initial approach had unacceptable performance, leading, in one instance, to an eight minute response time for an online transaction:

"We had some major architectural pieces missing, to be honest. Our database connection was inadequate. We were using an open gateway using dynamic SQL to map objects onto relational structures and it was slow in execution.... It is really difficult to do any functional kind of prototype when it takes you eight minutes to retrieve something for users." [Developer, EnergyCo E&P].

Ultimately a new approach was devised that appears to be marginally satisfactory, but at a cost of discarding two infrastructure tools that had been developed to support the original approach (a schema generator tool the module providing the dynamic link between Smalltalk and DB2).

Gaps in knowledge were not confined to sites without experienced staff. At BrokerCo UK there were also costly (though not project-threatening) gaps in knowledge. The most notable gap was evidenced by the belated realization that a formal object-oriented analysis phase is necessary if an organization wishes to develop systems generic enough to support significant levels of subsequent reuse.

"Because of time pressures there was no up front analysis and so no reuse; we had a prototype and were told to code it. Even on [the second

application] we have not gone high enough up in the business to get the true benefits of the OO approach; what has to come first, before anything, is a business model, a model of how the business really operates." [Developer, BrokerCo UK].

In the end, the advice of the experienced OO development manager at FinancialCo RMS quoted earlier is telling: a typical organization might eventually be able assimilate OO techniques relying primarily on existing staff, but it must be prepared to have initial projects to fail, or to show no net gain over what would have been achieved with existing techniques. Few organizations have the patience or resources for such a scenario.

ii) Adoption of a Simpler Technology Variant

Although BankCo UK does not necessarily qualify as early OO adopter, the approach taken on their OO pilot project (the BPM system) does nevertheless illustrate a second OO assimilation strategy, namely the adoption of a simpler technology variant. BankCo UK selected Kappa PC, rather than a Smalltalk or C++ based development environment, for the explicit reason that it was thought to be simpler to understand and use, and insulated staff members from many of the technological details of developing GUIs and interacting with Microsoft Windows.

For an early adopter, however, this strategy is inherently limited. While it is true that in the long run, if many organizations adopt, there will likely be tools that combine full functionality with comparative simplicity, in the short run, simpler technology variants typically sacrifice flexibility and operational performance. It is worth noting that the BPM system could not be implemented, mainly because the target organization was not using a local area network that supported Microsoft Windows—the target implementation environment for Kappa PC. If a more general purpose tool had been employed, then an insurmountably technological barrier might have been less likely. One of the primary reasons for using a language like Smalltalk at

EnergyCo E&P, by contrast, was precisely that it supported the ability to develop, if necessary, custom tools and interfaces for systems integration.

Nevertheless, if an organization's primary goal is learning about OO development and gaining some initial staff experience with OO concepts—rather than constructing large-scale production systems—then the approach taken at BankCo UK can be an attractive initial strategy to pursue while waiting for the technology to mature. The strategy is comparatively low cost, and has a fairly high probability of success.

iii) "Grafting" Experienced Personnel

FinancialCo RMS and BrokerCo UK both employed a strategy of primarily using externally hired developers with extensive prior experience in OO and/or related technologies to staff OO projects. This can be viewed as an instance of the "grafting" approach to organizational learning suggested by Huber [1991]. This approach appears to have been quite effective at both sites, thus providing some evidence countering the assertion by Cohen and Levinthal [1990] that grafting usually is not a feasible approach to organizational learning, owing to the need to have both technical and organizational knowledge residing in the same staff members.

Clearly, however, this is not a strategy that all early adopters could simultaneously employ because when a technology first emerges experienced personnel are necessarily scarce. Furthermore, some existing organizational knowledge must already exist for it to be able to recognize candidates that are genuine technology experts:

"Knowing what the words are and [not] knowing what the words mean in OO is probably a lot more prevalent of a problem, especially today. And much more prevalent of problem that it is using relational databases. There is so much misuse of the words, that I think that assuming that somebody else is going to be able to do with the technology what you expect them to do, is a problem." [Executive, FinancialCo RMS]

As mentioned by one manager at FinancialCo RMS, there is an element of luck involved. In 1990, he had an opportunity to hire one of the few OO experts available at the time, and leveraged this person to seek out and hire additional experienced staff. Subsequently, this person also helped assemble the development staff for BrokerCo UK. If not for that one initial hire, then a completely different scenario might have resulted at FinancialCo RMS, and BrokerCo might never have adopted OO at all. Another inhibitor of the external hiring strategy is the likelihood that organizations may have to go outside established pay scales to attract people experienced with the new technology.

In addition, an organization that follows this approach may be more prone to provoking the kind of us-versus-them mentality that nurtures political infighting:

"We started using object-oriented technology here almost four years ago and any time we admitted we were doing it , we got whacked by a two-by-four, had the project questioned, had to go through a series of reviews, had people call our management and ask them if they knew that we were [figuratively] taking drugs and the rest of that stuff." [Executive, FinancialCo RMS]

Finally, an organization that follows the grafting approach for initial projects may be jeopardizing a later transition of the larger organization to the new way of doing things. Existing staff may become resentful of the externally hired "stars" and blame the technology. As one manager at BrokerCo UK put it, in expressing one of his concerns about a potential organizational transition to OO:

"You need an insider to spread the gospel [of OO]; we're very much a group of outsiders." [Manager, BrokerCo UK]

The above concerns notwithstanding, the grafting approach, while costly and difficult to implement, appears to be the most likely to lead to success on initial development projects of any significant scale or complexity. And while this approach may indeed result in a more difficult subsequent organizational transformation, for many (if not most) organizations success

on initial projects is necessary for a technology to be even considered for more widespread subsequent use.

A4. SUMMARY AND CONCLUSIONS

This appendix has used case study data from four OOPL adopter to provide a rich illustration of the nature of the knowledge burden imposed by SPIs in general and OOPLs in particular. The magnitude of the knowledge burden was evidenced by the reported difficulties developers had in learning basic OO concepts; by the need for lengthy periods of learning-by-doing; by the need for knowledge of related technologies; by misjudgments about the actual level of maturity of OO technologies; and finally, by the apprehensions expressed about retraining existing staff should OO technologies ever become the dominant approach to systems development at the various case sites.

Among the four case sites, three distinct strategies for innovating in the presence of knowledge barriers were observed: 1) heavy investments in learning by existing staff, 2) adopting a simpler technology variant, and 3) "grafting" of experienced personnel onto the existing organization. The first strategy appears to be the most expensive, and appears to involve greatest risk of failure on initial projects. Yet, it may well help to avoid political battles initially, and later, may smooth the ultimate transition of the broader organization to the new approach. The second strategy, adopting a simpler technology variant, involves the least cost and risk, but at the price of inflexibility and a likely inability to scale-up to the larger systems that support core business processes. Nevertheless, those organizations that feel they can develop needed systems adequately with existing technologies, but still concerned about keeping current with new technologies, might consider embracing this approach until the technology matures. Ultimately, if the technology does succeed and become dominant, even full-function technology variants will have become much simpler to use. The third strategy, "grafting" of experienced personnel, may well be a necessary condition for ensuring success on initial projects. However, it is a strategy

that, due to the limited availability of qualified staff and the difficulty of identifying them, can only be employed by a small minority of potential early adopters. Furthermore, it may well be that organizations employing this approach run a greater risk of invoking widespread skepticism and resentment among existing staff, and hence, may face a more challenging organizational transition in the long run.

APPENDIX B: LISTING OF DBS SCREENS

Title
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|
4|
5|
6| SURVEY OF INFORMATION TECHNOLOGY MANAGERS
7|
8|
9|
10|
11|
12| Sponsored by: Programmed by:
13| International Data Corporation Para Technologies
14| Framingham, Massachusetts Costa Mesa, CA
15|
16|
17|
18|
19|
20| Press any key to continue.
21|
22|
23| (c) 1993 International Data Corporation, Para Technologies, Robert Fichman
24| All Rights Reserved
25|
-----1-----2-----3-----4-----5-----6-----7-----8

A-00a
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Hello and welcome to our
3|
4| NATIONAL SURVEY OF INFORMATION TECHNOLOGY MANAGERS
5|
6|
7|
8| The purpose of this survey is to find out how emerging applications
9| development technologies are being used.
10|
11| If you have any questions on how to complete the questionnaire,
12| please call us at the toll free number 1-800-720-7111.
13|
14| Thank you for your assistance!
15|
16|
17|
18| Press any key to continue.
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

A-00b

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           KEY BOARD INSTRUCTIONS
3|
4|
5|
6|   If you aren't sure about an exact answer, please respond with your
7|   best estimate.
8|
9|   To exit this survey at any time, press F10.  You can restart the survey
10|  at the same question by typing RESTART.
11|
12|  Press the F1 key to display context sensitive help.
13|
14|  Press the ESC key to move back to a previous item.
15|
16|
17|
18|
19|
20|
21|
22|           Press any key to continue.
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

B-00a

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|
4|
5|   The brief set of questions that follow are intended to make sure this
6|   questionnaire is appropriate for you to complete.
7|
8|
9|
10|
11|           Press any key to continue.
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

B-01a

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|   At how many locations does your company have a staff of full
4|   time information systems professionals developing custom software
5|   applications?
6|
7|
8|   Using the arrow keys, please highlight the one item that best applies
9|   and press ENTER.
10|
11|   -> No locations
12|   -> One location
13|   -> 2 to 5 locations
14|   -> 6 to 9 locations
15|   -> 10 to 19 locations
16|   -> 20 to 50 locations
17|   -> More than 50 locations
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

B-01b

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|
4|
5|
6|   For the remainder of this survey, "your location" (or "this
7|   location") refers to the one location you personally know
8|   the most about relating to software application development
9|   activities and technologies.
10|
11|
12|
13|
14|           Press any key to continue.
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

B-01c

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|
4| How knowledgeable are you about the software applications development
5| activities and supporting technologies used at your location over the
6| past several years?
7|
8|
9| Please highlight the one item that best applies and press ENTER.
10|
11|     Not at all knowledgeable
12|     Slightly knowledgeable
13|     Moderately knowledgeable
14|     Very knowledgeable
15|     Extremely knowledgeable
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

B-02

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|
4| What best describes your managerial level?
5|
6|
7| Please highlight the one item that best applies and press ENTER.
8|
9|
10|     Vice president or above
11|     Director
12|     Manager
13|     Supervisor
14|     Other managerial
15|     Project leader
16|     Staff/non-managerial
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```


B-03

-----1-----2-----3-----4-----5-----6-----7-----8

1|
2|
3|
4|
5|
6|
7|
8|
9|
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|

What percentage of all the custom applications development at your location are you and/or your subordinates responsible for?

Please highlight the one item that best applies and press ENTER.

- None
- Less than 10%
- Between 10% and 24%
- Between 25% and 49%
- Between 50% and 74%
- Between 75% and 100%

-----1-----2-----3-----4-----5-----6-----7-----8

C-01

-----1-----2-----3-----4-----5-----6-----7-----8

1|
2|
3|
4|
5|
6|
7|
8|
9|
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|

We have made every effort to send this questionnaire only to companies that have a full time staff of information systems professionals developing custom software applications.

You indicated that this is not the case for your company. Therefore, it is not appropriate for you to continue with the main portion of this survey.

Even so, it would be a great help to us if you would still return the diskette to us using the postage-paid mailer after exiting the questionnaire.

You are also entitled to receive a summary of the results of this study and the "reports" mentioned in our cover letter, but you must return the diskette to receive them.

Thank you very much for your assistance.

To exit survey now, press any key.

-----1-----2-----3-----4-----5-----6-----7-----8

D-01

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3| We have made every effort to send this questionnaire only to people
4| who are: 1) information technology managers and 2) at least slightly
5| knowledgeable about software development activities. You have indicated
6| that this is not true in your case.
7|
8| It would be a great help to us if you would forward this diskette and
9| materials to another manager at your company. Please give it to
10| someone who is knowledgeable about application development activities
11| at your location or another location.
12|
13| If you do not feel comfortable forwarding the survey to another
14| manager, then please return the diskette directly to us using the
15| postage-paid mailer. You may still receive a summary of the results of
16| the study and the reports mentioned in our cover letter but the diskette
17| must be returned to us for you to receive them.
18|
19| Thank you for your assistance.
20|
21|
22|
23| To exit survey now, press any key.
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

E-01a

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Object-oriented programming languages
3|
4|
5|
6|
7| The next set of questions ask about activities at your location related
8| to object-oriented programming languages.
9|
10|
11|
12|
13|
14| Press any key to continue.
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

E-02

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Object-oriented programming
3|
4|
5| How would you describe your current level of familiarity with
6| object-oriented programming (OOP)?
7|
8| Please highlight the one item that best applies and press ENTER.
9|
10|
11| Not at all familiar with OOP
12|
13| Have heard of OOP, but not familiar with concepts
14|
15| Somewhat familiar with OOP concepts
16|
17| Very familiar with OOP concepts
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

E-03

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Object-oriented programming
3|
4|
5| Has there ever been any evaluation or use of an object-oriented
6| programming language at your location?
7|
8|
9| Please highlight the one item that best applies and press ENTER.
10|
11|
12| Yes, some activity involving OOP
13|
14| No OOP activity to my knowledge
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

E-03a

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
1|
2|           Object-oriented programming languages
3|
4|
5|   In the following section we are interested in finding out about the
6|   level of activity at your location with general purpose, fully object-
7|   oriented languages, such as C++, Smalltalk, Objective C, and Eiffel.
8|
9|   Other languages and tools that are not fully object-oriented,
10|  including Ada, PowerBuilder, and Visual Basic, will be asked about in
11|  a later section.
12|
13|
14|
15|           Press any key to continue.
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
```

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

E-04

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
1|
2|           Object-oriented programming languages
3|
4|   Which of the OOPs below have been acquired at your location for
5|   evaluation, trial development projects, or production development
6|   projects?
7|
8|   Using the arrow keys, move the highlight bar to each item
9|   you wish to select.  Then press ENTER.  Press F1 for help.
10|
11|           C++
12|           Smalltalk
13|           Objective C
14|           Eiffel
15|           Object Pascal, Quick Pascal, or Turbo Pascal 5x
16|           CLOS
17|           NONE/NO MORE/NEXT QUESTION
18|
19|
20|
21|
22|
23|   To deselect an item, use arrow keys to highlight and press ENTER.
24|
25|
```

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

E-05

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5|   Which of the OOPs acquired at your location is the primary language
6|   (i.e., the OOP you expect to have the most future activity)?
7|
8|
9|   Please highlight your primary OOP and press ENTER.
10|
11|           C++
12|           Smalltalk
13|           Objective C
14|           Eiffel
15|           Object Pascal, Quick Pascal, or Turbo Pascal 5x
16|           CLOS
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

E-06

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5|   You indicated that your primary OOP was
6|
7|
8|
9|
10|  When was it first installed?
11|
12|  Please type the year and press ENTER.
13|
14|
15|           19xx   Year installed
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

E-07

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5|   How many software applications development staff members at this location
6|   had OOP development experience prior to acquisition of your first OOPL?
7|
8|
9|           Please type a number from 0 to 999 and press ENTER.
10|
11|
12|           Number OOP experienced personnel prior to acquisition
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

E-08

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|   Has this location ever approved the development of a significant
5|   production application using a general purpose OOPL as the primary
6|   programming language?
7|
8|           Press a number to record your answer.
9|
10|  1  Yes, at least one significant OOPL production application
11|     has been approved
12|
13|  2  No significant OOPL production applications have been approved
14|     at this location
15|
16|   Significant means projects requiring at least one person-month of
17|   development effort.
18|
19|   Production application means applications to be delivered to users
20|   and operated and maintained on an ongoing basis. (We do not include
21|   experimental, evaluation and pilot applications.)
22|
23|   General purpose OOPL = C++, Smalltalk, Objective C, Eiffel,
24|   Object/Quick/Turbo Pascal, or CLOS.
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

E-09

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5| What describes the current status of the general purpose OOPLs at
6| your location?
7|
8| Please highlight the one item that best applies and press ENTER.
9|
10|
11|    Previously evaluated and rejected OOPLs
12|
13|    Planning to evaluate OOPLs
14|
15|    Currently evaluating an OOPL, but not doing any development work
16|
17|    Using OOPL on trial/pilot projects, but not significant production projects
18|
19|
20|    General purpose OOPLs: C++, Smalltalk, Objective C, Eiffel
21|                          Object/Quick/Turbo Pascal, CLOS
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

E-10

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4| How many significant production applications development projects using
5| a general purpose OOPL as the primary language have ever been approved?
6|
7| Please type the number of projects (from 0 to 999) in each category
8| provided and press ENTER. Categories are mutually exclusive.
9|
10|           Projects approved for development but not yet initiated
11|           Projects in progress
12|           Projects implemented
13|           Projects initiated but later canceled/not implemented
14|           Don't know or other
15|           All correct? (Y/N)
16| By significant we mean projects requiring at least one
17| person-month of development effort
18|
19| By production applications we mean applications intended to be
20| delivered to users and operated and maintained on an ongoing basis.
21|
22| By general purpose OOPL we mean C++, Smalltalk, Objective C,
23| Eiffel, Object/Quick/Turbo Pascal, or CLOS.
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

E-10a

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Object-oriented programming languages
3|
4|
5| You have indicated that at least one significant OOPL project has
6| been approved or initiated at your location. Is this correct?
7|
8| Please highlight your response and press ENTER.
9|
10| Yes
11| No
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

E-10b

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|
4|
5| How many such projects have been approved or initiated at any time?
6|
7|
8| Please enter a number from 1 to 999 and press ENTER.
9|
10| Projects
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8


```

E-11 -----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4| You have indicated that xxx projects have been initiated then implemented
5| at this location using an OOPL. How successful were they?
6|
7|
8| Please type the number of projects (from 0 to 999) in
9| each category provided and press ENTER.
10|
11|           Projects
12|
13|           Completely successful ..
14|           Mostly successful .....
15|           Partly successful .....
16|           Unsuccessful .....
17|           Don't know.....
18|
19|           Total implemented projects
20|
21|
22|
23|           All correct? (Y/N)
24|
25| -----1-----2-----3-----4-----5-----6-----7-----8

```

```

E-12 -----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5| You've indicated that while you have one or more OOPL projects
6| implemented or canceled, none are currently pending or in progress.
7|
8| Please highlight the item that best describes the current status of
9| OOPL use at your location and press ENTER.
10|
11|
12|           No projects currently in progress, but future use expected
13|
14|           Discontinued pending further evaluation
15|
16|           Discontinued with no reevaluation currently planned
17|
18|           Don't know/Other
19|
20|
21|           General purpose COPLs: C++, Smalltalk, Objective C, Eiffel
22|           Object/Quick/Turbo Pascal, CLOS
23|
24|
25| -----1-----2-----3-----4-----5-----6-----7-----8

```

F-01

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5| What is the current status of general purpose OOPs at your location?
6|
7| Please highlight the one item that best applies and press ENTER.
8|
9|
10| Not interested in OOP/no current plans to evaluate
11|
12| Paying attention to OOP, but with no plans to evaluate right now
13|
14| Have definite plans to evaluate an OOP within 12 months
15|
16| Currently evaluating one or more OOPs
17|
18|
19|
20|           General purpose OOPs: C++, Smalltalk, Objective C, Eiffel
21|                               Object/Quick/Turbo Pascal, CLOS
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

F-02

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5| Listed below are some of the common reasons for deferring evaluation of
6| new technologies such as object-oriented programming languages (OOPs).
7|
8| Please highlight each item that applies in your case and press ENTER.
9|
10| -> Technology too immature
11| -> Too early to invest
12| -> Prefer to wait until standards have emerged
13| -> Not convinced claimed benefits of OOPs are genuine
14| -> OOPs are too incompatible with our legacy systems
15| -> OOPs are too difficult to evaluate and trial
16| -> Lack of management support for OOP evaluation
17| -> We have no pressing need for OOPs in our environment
18| -> Other new technologies are more promising than OOPs
19| -> Other
20| -> NONE/NO MORE/NEXT QUESTION
21|
22| To deselect an item, use arrow keys to highlight and press ENTER.
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

```

G-01 -----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5|
6| How many months ago did your evaluation of general purpose OOPLs begin?
7|
8|
9|           Please type a number from 1 to 999 and press ENTER.
10|
11|           Months ago
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25| -----1-----2-----3-----4-----5-----6-----7-----8

```

```

G-02 -----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5|
6| When will your location make a decision about using a general purpose
7| OOPL for development of production applications?
8|
9|
10| Please highlight the one time-frame that best applies then press ENTER.
11|
12|           Within 3 months
13|           Within 3 to 6 months
14|           Within 6 to 12 months
15|           Within 12 to 24 months
16|           Longer than 24 months
17|
18|
19|
20|
21|
22|
23|
24|
25| -----1-----2-----3-----4-----5-----6-----7-----8

```

G-03

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object-oriented programming languages
3|
4|
5| How likely do you think it is that an OOPL will be approved for use
6| on production applications within 24 months?
7|
8|
9|   Please highlight the one item that best applies then press ENTER.
10|
11|   Certain: 100% chance of approval
12|   Probable: 75% chance of approval
13|   Toss up: 50% chance of approval
14|   Doubtful: 25% chance of approval
15|   Unlikely: 0% chance of approval
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

g-05

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Buying Considerations
3|
4| Listed below are considerations that might influence an organization
5| when evaluating software products.
6|
7|   1 = Product related considerations (e.g. capabilities, performance)
8|   2 = Vendor related considerations (e.g. stability, reputation)
9|   3 = Other (e.g., company policy considerations or management edict)
10|  4 = Don't know
11|
12| For each product category, type the number (1 to 4) of the consideration
13| most likely to influence your location's evaluations and press ENTER.
14|
15|   Development tools for mainframes
16|   Development tools for midrange systems
17|   Development tools for workstations/PCs
18|
19|
20|           All correct? (Y/N)
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

g-06

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Product Promotion Methods
3|
4|   Listed below are product promotion methods vendors use to market products.
5|
6|       1 = Advertisements in magazines and journals
7|       2 = Presentations/demonstrations at seminars/trade shows/conferences
8|       3 = On site presentations/demonstrations
9|       4 = Advertising literature (promotional brochures, white papers)
10|      5 = Reviews by independent press or industry analysts
11|      6 = Experiences of peers or vendor reference accounts
12|      7 = Direct selling by an account rep or telemarketer
13|      8 = Other
14|      9 = Don't know
15|
16|   For each product category, type the number (1 to 9) of the promotion method
17|   most likely to influence your location's buying decisions and press ENTER.
18|
19|       Development tools for mainframes
20|       Development tools for midrange systems
21|       Development tools for workstations/PCs
22|
23|           All correct? (Y/N)
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

G-07

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Distribution Channels
3|
4|   Listed below are channels used to distribute products to customers.
5|
6|       1 = Directly from the vendor who developed the software product
7|       2 = From a retail store (any kind of store)
8|       3 = From an external distributor (mail order house, catalog, etc.)
9|       4 = From some other provider (systems integrator, hardware vendor, etc.)
10|      5 = From another part of my own company
11|      6 = Other
12|      7 = Don't know
13|
14|   For each software product, type the number (1 to 7) of the channel
15|   your location is most likely to use to buy products and press ENTER.
16|
17|       Development tools for mainframes
18|       Development tools for midrange systems
19|       Development tools for workstations/PCs
20|
21|           All correct? (Y/N)
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```


I-00a

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| OOPL Project Experiences
3|
4|
5|
6|
7| Even though use of a general purpose OOPL has been discontinued at your
8| location, we are interested in your experiences prior to discontinuance.
9|
10|
11|
12| Press any key to continue.
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

I-00b

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| OOPL Project Experiences
3|
4|
5|
6|
7| In this section we will be asking several questions about your current
8| and expected use of the OOPL on production projects.
9|
10|
11| Press any key to continue.
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

I-00c

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest-ever approved OOPL project
3|
4|
5|
6|
7| Please consider the largest project (in terms of person months of effort)
8| that has ever been approved for development at your location using an OOPL.
9|
10|
11|
12|
13|           Press any key to continue.
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

I-01

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest-ever approved OOPL project
3|
4|
5| What is the current status of this project?
6|
7|   Please highlight the one item that best applies and press ENTER.
8|
9|     Approved for development but not yet initiated
10|    In progress
11|    Implemented
12|    Initiated but later canceled/not implemented
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```


I-02

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest-ever approved OOPL project
3|
4|
5| What analysis/design methodology was, or will be, used on this
6| project?
7|
8|   Please highlight the one item that best applies and press ENTER.
9|
10|      No formal methods
11|      Internally developed methods
12|      Traditional structured methods or information engineering
13|      Any form of OO analysis and design
14|      Other
15|      Don't know
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

I-03

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest-ever approved OOPL project
3|
4|
5| How would you classify the application this project did, or will,
6| develop?
7|
8|   Please highlight the one item that best applies and press ENTER.
9|
10|
11|      Primarily batch MIS/transaction processing
12|      Primarily on-line MIS/transaction processing
13|      Information retrieval / reporting / query / DSS
14|      Scientific / engineering / modeling / simulation
15|      Real-time or process control
16|      Office automation / personal productivity / groupware
17|      Don't know/not sure
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

I-04

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest-ever approved OOPL project
3|
4| Please provide a one line description of the application this project did,
5| or will, develop (e.g., "customer order processing", "managing
6| inventory").
7|
8|           Please type the description and press ENTER.
9|
10| _____
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

I-05 & I-06

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest-ever approved OOPL project
3|
4|
5| How large was, or is, this project in person-months of developer
6| time?
7|
8|           Please type a number from 0 to 9999 and press ENTER.
9|
10|          Person-months
11|
12|
13|
14| What is your estimate of the number of object classes for this application?
15|
16|
17|           Please type a number from 0 to 9999 and press ENTER.
18|           In you can't estimate, type 8888 and press ENTER.
19|
20|          Number of object classes
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

```

I-07
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest-ever approved OOPL project
3|
4|
5|   What is the total duration (in calendar months) of this project from
6|   initiation to implementation (actual or expected)?
7|
8|   Please type the number from 1 to 999 and press ENTER.
9|
10|          Duration in calendar months
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

```

I-08
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest-ever approved OOPL project
3|
4|   Which general purpose OOPL is, or was, used the most on this
5|   application?
6|
7|
8|   Please highlight the one item that best applies and press ENTER.
9|
10|    C++
11|    Smalltalk
12|    Objective C
13|    Eiffel
14|    Object/Quick/Turbo 5x Pascal
15|    CLOS
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

I-09 & I-10

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest Implemented OOPL application
3|
4| Please consider the largest application development effort implemented
5| at your location using a general purpose OOPL as the primary programming
6| language.
7|
8|
9| How large was this project in person-months of development time?
10|
11|           Please type a number from 1 to 9999 and press ENTER.
12|
13|           Person-months
14|
15|
16| What is your estimate of the number of in-house object classes
17| developed for this application?
18|
19|           Please type a number from 0 to 9999 and press ENTER.
20|           If you can't estimate, type 8888 and press ENTER.
21|
22|           Number of object classes
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

I-11

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Largest Implemented OOPL application
3|
4|
5|
6| Would you classify any OOPL applications developed and implemented at
7| your location as being mission critical?
8|
9|           Please highlight your answer and press ENTER.
10|
11|           Yes
12|           No
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```


I-15a & I-15b

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Frequency of Use of OOPLs
3|
4| How likely is it that your location will use OOPLs on at least 25% of
5| new applications development within the next five years?
6|
7| Please highlight the one item that best applies then press ENTER.
8|
9|     Unlikely: Near 0% chance
10|    Doubtful: 25% chance
11|    Toss up:  50% chance
12|    Probable: 75% chance
13|    Certain:  Near 100% chance
14|
15| In which year is this 25% use most likely to first occur?
16| Highlight a year and press ENTER.
17|
18|     1994
19|     1995
20|     1996
21|     1997
22|     1998
23|     1999 or later
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

I-16

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Frequency of Use of OOPLs
3|
4|
5| What percent of the new application development effort (programmer hours)
6| for 1993 was in an OOPL?
7|
8| Please type a percent from 0 to 100 and press ENTER.
9|
10|           % Percent 1993 projects using OOPL
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

I-16a

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Adoption Triggers
3|
4|
5| Which of the factors below were significant in triggering your location
6| to adopt an OOPL for use on production projects?
7|
8| Using the arrow keys, highlight each item that applies and press ENTER.
9|
10| -> Expectations of reduced development cycle times
11| -> Expectations of increased developer productivity/lower costs
12| -> Expectations of more reliable applications
13| -> To help integrate production applications and desktops
14| -> To support new applications that can't be done with current technologies
15| -> To help achieve high levels of software reuse
16| -> To help move to client-server applications development
17| -> Competitors already adopting OOPLs
18| -> Need OOPLs to use other purchased software, e.g., Windows
19| -> Merged with another group that was already using an OOPL
20| -> Senior management strongly encouraged adoption
21| -> An external organization strongly encouraged adoption
22| -> Other
23| -> NONE/NO MORE/NEXT QUESTION
24| To deselect an item, use arrow keys to highlight and press ENTER.
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

I-17

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Frequency of Use of OOPLs
3|
4|
5| We would like to know why your location has discontinued use of
6| general purpose OOPLs.
7|
8| Using the arrow keys, highlight each item that applies and press ENTER.
9|
10| -> Inadequate performance (e.g., response times) of implemented systems
11| -> OOPL did not work as advertised
12| -> Difficult to integrate OOP applications with legacy systems
13| -> OOPL technology too immature
14| -> Developers found OOPL difficult to use
15| -> Waiting for standards to emerge
16| -> Requirements for software development technologies changed
17| -> Found a better technology to achieve the same purposes as OOPL
18| -> Other
19| -> NONE/NO MORE/NEXT QUESTION
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

J-01

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           How OOP is being used and supported
3|
4|   You have previously noted that your primary OOPL is
5|
6|   This is a language that does not require the use of
7|   object-oriented features such as encapsulated object classes.
8|
9|   Many adopters find that
10|  is used most effectively as a "better"
11|  and do not choose to use the optional object-oriented features
12|  very heavily.
13|           * * *
14|           Where does your location stand on this issue?
15|
16|   Please highlight the item that best describes how you are
17|   currently using encapsulated object classes and press ENTER.
18|
19|       No use of encapsulated object classes
20|       Minimal use of encapsulated object classes
21|       Frequently use encapsulated object classes
22|       Usually use encapsulated object classes
23|       Always use encapsulated object classes
24|       Don't know
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

J-02

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           How OOP is being used and supported
3|
4|   Provided below are categories for rating the status of reuse mechanisms.
5|
6|       1 = Not in place
7|       2 = Partly in place
8|       3 = Fully in place
9|       4 = Don't know
10|
11|  Which of the mechanisms below are in place in your location for OO projects?
12|
13|  For each mechanism, type the number (1 to 4) of the applicable status
14|  category and press ENTER.
15|
16|       Formal reuse procedures for developers to follow on
17|       most projects
18|       Separation of roles on most projects between class designers and
19|       class users/assemblers
20|       Full time staff tasked with ensuring application developers reuse
21|       existing object classes
22|       Performance evaluations of individuals/teams considers
23|       reuse achieved
24|           All correct? (Y/N)
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```


J-03

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           How OOP is being used and supported
3|
4|
5| Of the ____ OOPL applications you have indicated are planned,
6| completed, or in progress, how many would you classify as single
7| user vs. shared departmental vs. shared organization wide?
8|
9| Please type a number from 0 to 999 for each item and press ENTER.
10|
11|
12|           Single user
13|           Shared departmental
14|           Shared organization wide
15|           Don't know
16|           ----
17|           Total OOPL applications completed or in progress
18|
19|
20|
21|
22|           All correct? (Y/N)
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

J-04

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           How OOP is being used and supported
3|
4| Where has object-oriented technology been used heavily (on projects using
5| an OOPL at all)?
6|
7| Highlight each item that applies and press ENTER.
8|
9|     -> Presentation/user interface
10|    -> Application logic/business rules
11|    -> Database access/management
12|    -> NONE/NO MORE/NEXT QUESTION
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

J-05

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           How OOP is being used and supported
3|
4|
5| Which of the following kinds of externally purchased class libraries
6| has your location used on any OOPL projects?
7|
8|
9| Please highlight each item that applies and press ENTER.
10|
11|  -> Graphical user interfaces support
12|  -> Basic data structures (lists, strings, collections, etc.)
13|  -> Database access
14|  -> Industry or domain specific
15|  -> NONE/NO MORE/NEXT QUESTION
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

J-06

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           How OOP is being used and supported
3|
4|
5|
6|
7|
8| Which class libraries do you expect to use within the next 2 years?
9|
10|
11|
12|
13| Please highlight each item that applies and press ENTER.
14|
15|  -> Graphical user interfaces support
16|  -> Basic data structures (lists, strings, collections, etc.)
17|  -> Database access
18|  -> Industry or domain specific
19|  -> NONE/NO MORE/NEXT QUESTION
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

J-07

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           How OOP is being used and supported
3|
4|
5| Which tools, techniques or strategies have been used on at least
6| one OOP production development project?
7|
8| Please highlight each item that applies and press ENTER.
9|
10| -> External OOP experts
11| -> Professional OOP training classes
12| -> Project staff with previous OOP experience
13| -> OO seminars and conferences
14| -> Purchased object class libraries
15| -> OO analysis and design methodology
16| -> NONE/NO MORE/NEXT QUESTION
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

J-09

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           How OOP is being used and supported
3|
4| Of the staff you currently have programming in
5| what percent had their first experience with OOP before joining
6| your company?
7|
8| Please type a number from 0 to 100 and press ENTER.
9|
10|      %   Hired from outside
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

```

K-01
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object technologies
3|
4|   Provided below are categories for rating the status of technology adoption.
5|
6|     1 = No current activity
7|     2 = Research/evaluation is planned
8|     3 = Research/evaluation is in progress
9|     4 = Research/evaluation is complete
10|    5 = Approved for production
11|    6 = Used for at least one deployed system
12|    7 = Used regularly by most people/projects
13|    8 = Don't know/other
14|
15|   For each OO technology, type the number (1 to 8) of the status category
16|   that applies and press ENTER. (Press F1 for examples of these technologies).
17|
18|
19|     Object Oriented 4GLs / generators / GUI builders
20|     Object Oriented database management systems
21|     Object Oriented analysis or design methodologies
22|     Object Oriented Upper CASE tools for analysis/design
23|
24|           All correct? (Y/N)
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

```

k-03
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Object Technologies
3|
4|   In what order has your location acquired or would it be most likely
5|   to acquire the technologies below?
6|
7|   Use arrow keys and press ENTER to select the most likely
8|   item, continue to select items in order.
9|
10|
11|     Object Oriented programming language
12|     Object Oriented database management systems
13|     Object Oriented Upper CASE tools for analysis/design
14|     Object Oriented 4GLs / generators / GUI builders
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

L-00a

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|
4|
5| In this section we will be asking a few questions about three software
6| development technologies that emerged in the 1980's:
7|
8|
9|     1) Relational databases
10|
11|     2) Fourth-generation languages
12|
13|     3) Upper CASE tools
14|
15|
16|             Press any key to proceed.
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

L-01 & L-02

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|             Relational Database Management Systems (RDBMS)
3|
4|
5| In the questions that follow, we are interested in RDBMSs capable of
6| developing medium to large multi-user applications, such as IBM DB2,
7| Oracle, Sybase, Digital RDB, Ingres, Informix, ADABAS, Datacom/db.
8|
9|             * * *
10|
11| Has your location ever installed a RDBMS for evaluation, trial or use?
12|
13|     Highlight your answer and then press ENTER.
14|
15|         Yes
16|         No
17|
18|
19| In which year was it first purchased?
20|
21| Type in your answer and then press ENTER.
22|
23|         19
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

L-03a

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Relational Database Management Systems (RDBMS)
3|
4| Which of the following has occurred at your location related to RDBMS?
5|
6|   Please highlight each item that has occurred and press ENTER:
7|
8|   Approval of RDBMSs for use on production applications
9|   Implementation of a multi-user application using RDBMS
10|  Implementation of a large, mission critical application using RDBMS
11|  Use of RDBMS on at least 25% of all new development in the same year
12|  NONE/NO MORE/NEXT QUESTION
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

L-03b

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Relational Database Management Systems (RDBMS)
3|
4|
5| In which year did your location first use an RDBMS on at least 25% of
6| new applications development?
7|
8|     19
9|
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

L-03c

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Relational Database Management Systems (RDBMS)
3|
4| How will your location's use of RDBMSs to develop production applications
5| change in the next three years?
6|
7| Please highlight the one item that best applies and press ENTER:
8|
9| Increase substantially
10| Increase moderately
11| Increase a small amount
12| Stay about the same as it is now
13| Decrease a small amount
14| Decrease moderately
15| Decrease substantially
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

M-01 & M-02

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Production Fourth Generation Languages (4GLs)
3|
4|
5|
6|
7| Has your location ever installed a 4GL capable of developing medium
8| to large multi-user applications (e.g., Natural, Ideal, Adds Online,
9| Ramis, Focus) for evaluation, trial or use?
10|
11| Yes
12| No
13|
14|
15|
16|
17| In which year was it first purchased?
18|
19| 19
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

M-03a

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Production Fourth Generation Languages (4GLs)
3|
4| Which of the following has occurred at your location related to 4GLs?
5|
6| Please highlight each item that has occurred and press ENTER:
7|
8| Approval of 4GLs for use on production applications
9| Implementation of a multi-user application using 4GL
10| Implementation of a large, mission critical application using 4GL
11| Use of 4GL on at least 25% of all new development in the same year
12| NONE/NO MORE/NEXT QUESTION
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

M-03b

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Production Fourth Generation Languages (4GLs)
3|
4| In which year did your location first use a production 4GL on at least 25%
5| of new applications development?
6|
7| 19
8|
9|
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

M-03c

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Production Fourth Generation Languages (4GLs)
3|
4| How will your location's use of 4GLs to develop production applications
5| change in the next three years?
6|
7| Please highlight the one item that best applies and press ENTER:
8|
9| Increase substantially
10| Increase moderately
11| Increase a small amount
12| Stay about the same as it is now
13| Decrease a small amount
14| Decrease moderately
15| Decrease substantially
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

N-01 & N-02

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Upper CASE Tools for Analysis and Design
3|
4|
5| In the questions that follow, we are interested only in traditional (non
6| object-oriented) Upper CASE tools appropriate for developing medium to
7| large multi-user applications, e.g., Texas Instruments IEF, KnowledgeWare
8| ADW, Excelerator, CADRE TeamWork, LDBMS.
9|
10| * * *
11|
12| Has your location ever installed an Upper CASE tool for evaluation,
13| trial or use?
14|
15| Yes
16| No
17|
18|
19|
20| In which year was it first purchased?
21|
22| 19
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

N-03a

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Upper CASE Tools for Analysis and Design
3|
4| Which of the following has occurred at your location related to
5| Upper CASE tools?
6|
7| Please highlight each item that has occurred and press ENTER:
8|
9| Approval of Upper CASE for use on production applications
10| Implementation of a multi-user application using Upper CASE
11| Implementation of a large, mission critical application using Upper CASE
12| Use of Upper CASE on at least 25% of all new development in the same year
13| NONE/NO MORE/NEXT QUESTION
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

N-03b

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Upper CASE Tools for Analysis and Design
3|
4| In which year did your location first use an Upper CASE tool on at least
5| 25% of new applications development?
6|
7| 19
8|
9|
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

N-03c

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Upper CASE Tools for Analysis and Design
3|
4| How will your location's use of Upper CASE tools to develop production
5| applications change in the next three years?
6|
7| Please highlight the one item that best applies and press ENTER:
8|
9| Increase substantially
10| Increase moderately
11| Increase a small amount
12| Stay about the same as it is now
13| Decrease a small amount
14| Decrease moderately
15| Decrease substantially
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

O-01

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Application Developers: Skills and Background
3|
4| In this section we would like to ask a few questions about the skills
5| of your applications development staff.
6|
7| * * *
8|
9| What percentage of the applications development staff at your location
10| fit in the categories listed below?
11|
12|
13| Please type a percent for each item and press ENTER.
14|
15| % Bachelors degree is highest obtained
16|
17| % Masters degree or higher
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

O-03

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Application Developers: Skills and Background
3|
4|   What percentage of the development staff at your location have had
5|   experience in the areas below?
6|
7|   Please type a percent for each item and press ENTER.
8|
9|           % Programming using the C language
10|
11|          % Developing client-server applications
12|
13|          % Developing PC or workstation-based applications
14|
15|          % Developing graphical user interfaces
16|
17|          % Developing data models using entity relationship diagrams
18|
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

O-04

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Application Developers: Skills and Background
3|
4|
5|   Prior to your acquisition of your first OOPL what percentage of the
6|   development staff at your location had experience in the areas below?
7|
8|   Please type a percent for each item and press ENTER.
9|
10|          % Programming using the C language
11|
12|          % Developing client-server applications
13|
14|          % Developing PC or workstation-based applications
15|
16|          % Developing graphical user interfaces
17|
18|          % Developing data models using entity relationship diagrams
19|
20|
21|
22|
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

P-01

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
1|
2|           Trends in new development
3|
4| Now we would like to learn more about the specific breakdown of
5| applications development-related activity at your location.
6|
7|           * * *
8|
9| Please divide your location's total applications development and
10| maintenance effort for 1994 into the categories listed below.
11|
12| Please type a percent for each item and press ENTER.  Make the total
13| equal 100%.
14|
15|     % Maintenance and support of existing systems
16|     ‡ Significant enhancements to existing systems
17|     % Development of new systems, or rewrites of existing systems
18|     -----
19|     % Total
20|
21|
22|           All correct? (Y/N)
23|
24|
25|-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

P-01a

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
1|
2|           Trends in new development
3|
4| Over the past three years did the volume of new applications development
5| increase, decrease, or stay about the same?
6|
7|
8| Please highlight the one item that best applies and press ENTER.
9|
10|     Increased substantially
11|     Increased moderately
12|     Increased a small amount
13|     Stayed about the same as it was before
14|     Decreased a small amount
15|     Decreased moderately
16|     Decreased substantially
17|
18|
19|
20|
21|
22|
23|
24|
25|-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

P-02

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Trends in new development
3|
4| Over the next three years, will the volume of new applications
5| development increase, decrease, or stay about the same?
6|
7|
8| Please highlight the one item that best applies and press ENTER.
9|
10| Increase substantially
11| Increase moderately
12| Increase a small amount
13| Stay about the same as it is now
14| Decrease a small amount
15| Decrease moderately
16| Decrease substantially
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

P-03 & P-04

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Trends in new development
3|
4|
5| What percentage of your major applications did you redevelop or
6| replace over the past three years?
7|
8| Please type a percent between 0 and 100 and press ENTER.
9|
10| % Will be replaced
11|
12|
13| What percentage of your major applications do you expect to redevelop
14| or replace over the next three years?
15|
16| Please type a percent between 0 and 100 and press ENTER.
17|
18| % Were replaced.
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

Q-00

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3| It should take about 8 more minutes to complete this survey.
4|
5|
6| Press any key to continue.
7|
8|
9|
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

Q-01

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| General Information about IT Department
3|
4| What is the primary industry of the company supported by your
5| location's application development staff?
6|
7| Highlight the one item that best applies and press ENTER.
8|
9| Manufacturing - Discrete (repetitive and job shop)
10| Manufacturing - Process (continuous)
11| Insurance and financial services
12| Banking
13| Healthcare/medical
14| Education
15| Government
16| Business services and other services
17| Transportation
18| Communications
19| Utilities
20| Energy (oil/gas/coal production and refining)
21| Wholesale trade
22| Retail trade
23| Other
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

Q-01a

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|   What other industries does your company operate in, and have
5|   application development staff (at this location) supporting?
6|
7|           Highlight each item that applies and press ENTER.
8|
9|           Manufacturing - Discrete (repetitive and job shop)
10|          Manufacturing - Process (continuous)
11|          Insurance and financial services
12|          Banking
13|          Healthcare/medical
14|          Education
15|          Government
16|          Business services and other services
17|          Transportation
18|          Communications
19|          Utilities
20|          Energy (oil/gas/coal production and refining)
21|          Wholesale trade
22|          Retail trade
23|          Other
24|          NONE/NO MORE/NEXT QUESTION
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

Q-02

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|
5|   What was the division of new applications development activities over
6|   the past three years across runtime platforms?
7|
8|
9|   Type a percentage for each item and press ENTER. Please make
10|  percentages total to 100%.
11|
12|          %   Traditional mainframe
13|          %   Traditional midrange (mini or supermini)
14|          %   Client-server with mainframe host
15|          %   Client-server with midrange host
16|          %   Client-server with desktop host
17|          %   Networked peer-to-peer workstations or PCs
18|          %   Standalone workstations or PCs
19|          %   Other
20|          -----
21|          %   Total
22|
23|           All correct? (Y/N)
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```


Q-03

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|
5|  What was the division of new applications development activities over
6|  the past three years across application types?
7|
8|
9|  Type a percentage for each item and press ENTER. Please make
10| percentages total to 100%.
11|
12|      %    Primarily batch MIS/transaction processing
13|      %    Primarily on-line MIS/transaction processing
14|      %    Information retrieval / reporting / query / DSS
15|      %    Scientific / engineering / modeling / simulation
16|      %    Real-time or process control
17|      %    Office automation / personal productivity / groupware
18|  -----
19|      %    Total
20|
21|           All correct? (Y/N)
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

Q-04

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|
5|  For which of the following specialized functions do you have a
6|  staff of one or more full time professionals?
7|
8|  Please highlight each item that applies and press ENTER.
9|
10|  -> Advanced technology evaluation
11|  -> Quality assurance
12|  -> Data administration
13|  -> Methods & tools development
14|  -> Metrics and measurement
15|  -> System testing
16|  -> NONE/NO MORE/NEXT QUESTION
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

Q-05a & Q-05b

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|
5|   What percent of your applications development staff were using each
6|   of the following languages in 1993? What do you expect for 1995?
7|   (Since some programmers use more than one language, columns can sum
8|   to more than 100%)
9|
10|  Type a percentage for each item and press ENTER.
11|
12|           | 1993 |   | 1995 |
13|   Assembly language ..... %      %
14|   COBOL ..... %      %
15|   C language ..... %      %
16|   Another 3GL (Fortran, PL/, Basic, RPG) %      %
17|   A traditional 4GL (Natural, Ideal) . %      %
18|   C++ ..... %      %
19|   Smalltalk ..... %      %
20|   Another CO language ..... %      %
21|
22|           All correct? (Y/N)
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

Q-06

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|   Which of the development environments/operating systems listed below serve
5|   as host (i.e., by supporting program compilation) to at least 5% of your
6|   location's development work?
7|
8|   Please highlight each item that applies and press ENTER.
9|
10|   -> Mainframe/midrange systems accessed by terminals
11|   -> Mainframe/midrange systems accessed by terminal emulation on PCs
12|   -> MS-DOS/Windows
13|   -> Windows NT
14|   -> OS/2
15|   -> Taligent
16|   -> NextStep
17|   -> Solaris
18|   -> IBM Aix
19|   -> HP-UX
20|   -> Other Unix
21|   -> Macintosh
22|   -> Other
23|   -> NONE/NO MORE/NEXT QUESTION
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

Q-07

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4| What is the primary development environment/operating system for the
5| programmers at your location (i.e., the host environment where most programs
6| are compiled)?
7|
8| Please highlight the one item that best applies and press ENTER.
9|
10|    -> Mainframe/midrange systems accessed by terminals
11|    -> Mainframe/midrange systems accessed by terminal emulation on PCs
12|    -> MS-DOS/Windows
13|    -> Windows NT
14|    -> OS/2
15|    -> Taligent
16|    -> NextStep
17|    -> Solaris
18|    -> IBM Aix
19|    -> HP-UX
20|    -> Other Unix
21|    -> Macintosh
22|    -> Other/Don't know
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

Q-08

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4| What will be the primary development environment/operating system for
5| programmers at your location by the end of 1995 (i.e., the host environment
6| where most programs are compiled)?
7|
8| Please highlight the one item that best applies and press ENTER.
9|
10|    -> Mainframe/midrange systems accessed by terminals
11|    -> Mainframe/midrange systems accessed by terminal emulation on PCs
12|    -> MS-DOS/Windows
13|    -> Windows NT
14|    -> OS/2
15|    -> Taligent
16|    -> NextStep
17|    -> Solaris
18|    -> IBM Aix
19|    -> HP-UX
20|    -> Other Unix
21|    -> Macintosh
22|    -> Other/Don't know
23|
24|
25|-----1-----2-----3-----4-----5-----6-----7-----8
```

Q-09 & Q-10

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           General Information about IT Department
3|
4|
5|   What percent of the total installed base of applications at your
6|   location are written in COBOL?
7|
8|           %   Written in COBOL
9|
10|
11|
12|   What percent of existing COBOL applications do you expect to
13|   begin redeveloping as client-server by the end of 1996?
14|
15|           %   Redeveloped for client-server
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

R-00a

```
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Size and Scope of IT activities
3|
4|
5|   How many years have custom software applications been developed
6|   at your location?
7|
8|
9|   Using the arrow keys, please highlight the one item that best applies
10|   and press ENTER.
11|
12|   -> Less than 1 year
13|   -> 1 to 2 years
14|   -> 3 to 4 years
15|   -> 5 to 10 years
16|   -> More than 10 years
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8
```

r-00b

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Size and Scope of IT activities
3|
4|
5| How many application developers, including analysts, programmers,
6| and project leaders are currently engaged in applications
7| development at this location?
8|
9| Please type a number from 0 to 9999 and press ENTER.
10|
11| Size of professional staff developing/maintaining applications
12|
13|
14|
15| Include all non-managerial applications personnel located at your
16| location whether in the IT department or decentralized in line
17| departments.
18|
19| Exclude systems programming, operations personnel, support personnel
20| and "power users"
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

R-01

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Size and Scope of IT activities
3|
4| What is the total 1994 budget for information technology (IT) at
5| your location, including development, operations and support functions?
6|
7| \$ 000 IT Budget
8|
9|
10| If you don't know the exact number, provide your best estimate.
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

R-01a

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Size and Scope of IT Activities
3|
4| What is the total external IT spending on hardware, software, networking
5| and contract services (excluding payroll and overhead) related to
6| object-oriented technologies for 1994?
7|
8| Please highlight the one range that best applies and press ENTER.
9|
10| Less than \$1,000
11| \$1,000 - \$9,999
12| \$10,000 - \$49,999
13| \$50,000 - \$99,999
14| \$100,000 - \$199,999
15| \$200,000 - \$299,999
16| \$300,000 - \$399,999
17| \$400,000 - \$499,999
18| \$500,000 - \$599,999
19| \$600,000 - \$699,999
20| \$700,000 - \$799,999
21| \$800,000 - \$899,999
22| \$900,000 - \$999,999
23| More than \$1,000,000
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

R-02, R-03 and R-04

-----1-----2-----3-----4-----5-----6-----7-----8
1|
2| Size and Scope of IT Activities
3|
4| What is the total number of IT related staff at your location, including
5| all development, operations and support personnel, managerial and non-
6| managerial?
7|
8| Total 1994 IS staff
9|
10| What was the total revenue for your company (division) in thousands
11| of dollars for the latest year figures are available?
12|
13| \$ 000 Total company revenues
14|
15| What is the total number of employees in your company (division)?
16|
17| Total number of employees
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

S-01
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Performance Objectives
3|
4| Provided below is a scale for rating the importance of performance objectives.
5|
6|   Very unimportant                               Extremely important
7|
8|           1.....2.....3.....4.....5.....6.....7
9|
10|
11| How important is each objective listed below for your location?
12|
13| Type the scale number (1 to 7) for each object and press ENTER.
14|
15|   Very rapid applications development
16|   Cost effective development
17|   Meeting project schedules & budgets
18|   Deploying very high performance applications
19|   Deploying very high reliability applications
20|   Deploying very easy to use applications
21|   Deploying very easy to change applications
22|
23|           All correct? (Y/N)
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

```

S-02
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Performance Objectives
3|
4| Listed below are options for rating the frequency of meeting objectives.
5|
6|   1 = Nearly always met
7|   2 = Met at least 75% of the time
8|   3 = Met at least 50% of the time
9|   4 = Met at least 25% of the time
10|  5 = Rarely met
11|
12| How often has each performance objective below been met for projects
13| completed by your location over the last three years?
14|
15| Type the number of the frequency (1 to 5) that applies and press ENTER.
16|
17|   Very rapid applications development
18|   Cost effective development
19|   Meeting project schedules & budgets
20|   Deploying very high performance applications
21|   Deploying very high reliability applications
22|   Deploying very easy to use applications
23|   Deploying very easy to change applications
24|           All correct? (Y/N)
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

```

dbmq1
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Experience with disk-based survey
3|
4|
5|
6|           Have you ever answered a survey using a computer before?
7|
8|
9|           Yes
10|          No
11|          Don't know
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

```

dbmq2
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|           Experience with disk-based survey
3|
4|
5|           Based on your experience in answering this computerized
6|           survey, would you be more or less willing to answer another
7|           survey using a computer?
8|
9|
10|          Much more willing
11|          Somewhat more willing
12|          About the same
13|          Somewhat less willing
14|          Much less willing
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```



```

comments
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|
4|      We're interested in any comments you may have.
5|
6|      Please type them in below.
7|
8|      When you are finished, press ENTER twice.
9|
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

```

End
-----1-----2-----3-----4-----5-----6-----7-----8
1|
2|
3|
4|
5|
6|
7|
8|
9|
10|
11|      Thank you for your participation.
12|
13|
14|      Please wait for the prompt to appear
15|
16|      and then return the disk in the mailer.
17|
18|
19|
20|
21|
22|
23|
24|
25|
-----1-----2-----3-----4-----5-----6-----7-----8

```

REFERENCES

- Abdel-Hamid, T. K. and Madnick, S. E. (1989) "Lessons Learned From Modeling the Dynamics of Software Development," *Communications of the ACM*, 32, 12, 1426-1438.
- Allison, P. D. (1982) "Discrete-Time Methods for the Analysis of Event Histories," in *Sociological Methodology*, Leinhardt, S. (ed.), Jossey-Bass: San Francisco, 61-98.
- Anonymous (1992) "Ci3 Update", *Sawtooth News*, 8, 2, Sawtooth Software, Evanston, Illinois.
- Argote, S.; Beckman, S. L. and Epple, D. (1990) "The Persistence and Transfer of Learning in Industrial Settings," *Management Science*, 36, 140-154.
- Arrow, K. (1962) "The Economic Implications of Learning by Doing," *Review of Economic Studies*, 29, 166-170.
- Arthur, W. B. (1988) "Competing Technologies: An Overview," in *Technical Change and Economic Theory*, Dosi, G. et al. (ed.), Pinter Publishers: London, 590-607.
- Attewell, P. (1992) "Technology Diffusion and Organizational Learning: The Case of Business Computing," *Organization Science*, 3, 1, 1-19.
- Badaracco, J. (1991) *The Knowledge Link*, Harvard Business School Press: Boston.
- Bagozzi, R. P. (1980) *Causal Modeling in Marketing*, John Wiley & Sons: New York.
- Bagozzi, R. P. and Phillips, L. W. (1982) "Representing and Testing Organizational Theories: A Holistic Construal," *Administrative Science Quarterly*, 27, 459-489.
- Bass, F. M. (1969) "A New Product Growth Model For Consumer Durables," *Management Science*, 15, 1, 215-227.
- Bayer, J. and Melone, N. (1989) "A Critique of Diffusion Theory as a Managerial Framework for Understanding the Adoption of Software Engineering Innovations," *Journal of Systems and Software*, 9, 161-166.
- Beatty, C. A. (1992) "Implementing Advanced Manufacturing Technologies: Rules of the Road," *Sloan Management Review*, 35, Summer, 49-60.
- Bernstein, S. (1993) Apple Computer, personal communication.

- Booch, G. (1994) *Object Oriented Analysis and Design with Applications*, 2nd edition, Benjamin/Cummings: Redwood City, CA.
- Brancheau, J. C. and Wetherbe, J. C. (1990) "The Adoption of Spreadsheet Software: Testing Innovation Diffusion Theory in the Context of End-User Computing," *Information Systems Research*, 1, 2, 115-143.
- Brandt, R. (1993) "A Tricky Tack for Borland," *Business Week*, August 2, 1993.
- Bretschneider, S. and Wittmer, D. (1993) "Organizational Adoption of Microcomputer Technology: The Role of Sector," *Information Systems Research*, 4, 1, 88-109.
- Brooks, F. P. (1987) "No Silver Bullet: Essence and Accidents of Software Engineering," *Computer*, 20, 4, 10-19.
- Brynjolfsson, E. and Kemerer, C. F. (1993) "Determinants of the Demand for Microcomputer Software: An Econometric Analysis", *Fourteenth International Conference on Information Systems*, Orlando, Florida, December.
- Bulkeley, W. M. (1990) "Bright Outlook for Artificial Intelligence Yields to Slow Growth and Big Cutbacks," *Wall Street Journal*, pp. B1, B3, July 5, 1990.
- Chew, B. W.; Leonard-Barton, D. and Bohn, R. E. (1991) "Beating Murphy's Law," *Sloan Management Review*, 32, 3, 5-16.
- Christensen, C. M. (1994) "Exploring the Limits of the Technology S-Curve," *Production and Operations Management*, 1, 4, 334-366.
- Clark, D. (1995) "Borland's Kahn Resigns as President and CEO; Wetsel to Lead Restructuring," *Wall Street Journal*, pp. B6, January 12, 1995.
- Coad, P. and Yourdon, E. (1991) *Object-Oriented Analysis*, 2nd edition, Prentice-Hall: Englewood Cliffs, NJ.
- Cohen, W. M. and Levinthal, D. A. (1990) "Absorptive Capacity: A New Perspective on Learning and Innovation," *Administrative Science Quarterly*, 35, March, 128-152.
- Coleman, D. and Hayes, F. (1991) "Getting the Best From Objects: The Experience of Hewlett-Packard," *Hotline on Object-Oriented Technology*, 2, 10.
- Cook, T. D. and Campbell, D. T. (1979) *Quasi-Experimentation*, Houghton Mifflin Company: Boston.

- Cooper, R. B. and Zmud, R. W. (1990) "Information Technology Implementation Research: A Technological Diffusion Approach," *Management Science*, 36, 2, 123-139.
- Cox, B. J. (1990) "Planning the Software Industrial Revolution," *IEEE Software*, November, 25-33.
- Cox, D. R. and Oakes, D. (1984) *Analysis of Survival Data*, Chapman and Hall: London.
- Cryer, J. D. and Miller, R. B. (1991) *Statistics for Business: Data Analysis and Modeling*, PWS-Kent: Boston.
- Damanpour, F. (1991) "Organizational Innovation: A Meta-Analysis of Effects of Determinants and Moderators," *Academy of Management Journal*, 34, 3, 555-590.
- David, P. A. (1985) "Clio and the Economics of QWERTY," *American Economic Review, Proceedings*, 75, 332-337.
- Dietrich, W. C.; Nackman, L. R. and Gracer, F. (1989) "Saving a Legacy with Objects", *Proceedings, OOPSLA '89*, New Orleans, LA, 71-75.
- Dillman, D. (1983) *Mail and Telephone Surveys*, John Wiley & Sons: New York.
- Dodgson, M. (1993) "Organizational Learning: A Review of Some Literatures," *Organization Studies*, 14, 3, 375-394.
- Downs, G. W. and Mohr, L. B. (1976) "Conceptual Issues in the Study of Innovation," *Administrative Science Quarterly*, 21, 700-714.
- Ettlie, J. E. (1980) "Adequacy of Stage Models for Decisions on Adoption of Innovation," *Psychological Reports*, 46, 991-995.
- Eveland, J. D. and Tornatzky, L. G. (1990) "The Deployment of Technology," in *The Processes of Technological Innovation*, Tornatzky, L. G. and M. Fleischer (ed.), Lexington Books: Lexington, Massachusetts, 117-148.
- Farrell, J. and Saloner, G. (1987) "Competition, Compatibility and Standards: The Economics of Horses, Penguins and Lemmings," in *Product Standardization and Competitive Strategy*, Gabel, H. L. (ed.), Elsevier Science: North-Holland, 940-955.
- Fichman, R. G. (1992) "Information Technology Diffusion: A Review of Empirical Research", *Proceedings of the Thirteenth International Conference on Information Systems*, Dallas, December, 195-206.

- Fichman, R. G. and Kemerer, C. F. (1992) "Object-Oriented and Conventional Analysis and Design Methodologies: Comparison and Critique," *IEEE Computer*, October, 22-39.
- Fichman, R. G. and Kemerer, C. F. (1993) "Adoption of Software Engineering Process Innovations: The Case of Object Orientation," *Sloan Management Review*, 34, 2, 7-22.
- Fiol, C. M. and Lyles, M. A. (1985) "Organizational Learning," *Academy of Management Review*, 10, 803-813.
- Foster, R. N. (1986) *Innovation: The Attacker's Advantage*, Summit Books: New York.
- Gatignon, H. and Robertson, T. S. (1989) "Technology Diffusion: An Empirical Test of Competitive Effects," *Journal of Marketing*, 53, January, 35-49.
- Granovetter, M. (1978) "Threshold Models of Collective Behavior," *American Journal of Sociology*, 83, 6, 1420-1443.
- Grover, V. and Goslar, M. D. (1993) "The Initiation, Adoption and Implementation of Telecommunications Technologies in U.S. Organizations," *Journal of Management Information Systems*, 10, 1, 141-163.
- Horton, L. (1990) "Disk-Based Surveys: A New Way to Pick Your Brain," *Software Magazine*, 76-77.
- Howard, G. S. and Rai, A. (1993) "Promise and Problems: CASE Usage in the US," *Journal of Information Technology*, 8, 65-73.
- Huber, G. P. (1991) "Organizational Learning: The Contributing Processes and the Literatures," *Organizational Science*, 2, 1, 88-115.
- Huber, G. P. and Power, D. J. (1985) "Retrospective Reports of Strategic-Level Managers: Guidelines for Increasing Their Accuracy," *Strategic Management Journal*, 6, 171-180.
- Huff, C. C. (1992) "Elements of a Realistic CASE Tool Adoption Budget," *Communications of the ACM*, 35, 4, 45-54.
- Jöreskog, K. G. (1971) "Statistical Analysis of Cognetic Tests," *Psychometrika*, 36, 108-133.
- Katz, M. L. and Shapiro, C. (1986) "Technology Adoption in the Presence of Network Externalities," *Journal of Political Economy*, 94, 4, 822-41.
- Keisler, S. and Sproull, L. S. (1986) "Response Effects in the Electronic Survey," *Public Opinion Quarterly*, 50, 402-413.

Kelly, P. and Kranzberg, M. (1978) "The Diffusion of Innovation," in *Technological Innovation: A Critical Review of Current Knowledge*, Kelly, P. and M. Kranzberg (ed.), San Francisco Press: San Francisco.

Kemerer, C. F. (1989) "An Agenda for Research on the Managerial Evaluation of Computer-Aided Software Engineering Tool Impacts", *Proceedings of the 22nd Hawaii International Conference on System Sciences*, Hawaii, 219-228.

Kemerer, C. F. (1992) "How the Learning Curve Affects CASE Tool Adoption," *IEEE Software*, 9, 3, 23-28.

Kimberley, J. R. and Evanisko, M. J. (1981) "Organizational Innovation: The Influence of Individual, Organizational, and Contextual Factors on Hospital Adoption of Technical and Administrative Innovations," *Academy of Management Journal*, 24, 689-713.

Kogut, B. (1988) "Joint Ventures: Theoretical and Empirical Perspectives," *Strategic Management Journal*, 9, 319-332.

Kwon, T. H. and Zmud, R. W. (1987) "Unifying the Fragmented Models of Information Systems Implementation," in *Critical Issues in Information Systems Research*, Boland, J. R. and R. Hirshheim (ed.), John Wiley: New York, 227-251.

Lawless, J. F. (1982) *Statistical Models for Lifetime Data*, John Wiley & Sons: New York.

Leathers, B. (1990) "Panel: OOP In the Real World", *Proceedings, ECOOP/OOPSLA '90*, , 299-302.

Lee, S. (1989) *Managing I/S Planning and Design: A Control Theory Perspective*, unpublished Massachusetts Institute of Technology Ph.D dissertation.

Leonard-Barton, D. (1988a) "Implementation as Mutual Adaptation of Technology and Organization," *Research Policy*, 17, 251-267.

Leonard-Barton, D. (1988b) "Implementation Characteristics of Organizational Innovations," *Communication Research*, 15, 5, 603-631.

Leonard-Barton, D. (1989) "Implementing New Production Technologies: Exercises in Corporate Learning," in *Managing Complexity in High Technology Industries: Systems and People*, Von Glinow, M. A. and S. Mohrman (ed.), Oxford Press.

Leonard-Barton, D. and Deschamps, I. (1988) "Managerial Influence in the Implementation of New Technology," *Management Science*, 34, 10, 1252-1265.

- Levitt, B. and March, J. G. (1988) "Organizational Learning," *Annual Review of Sociology*, 14, 319-340.
- Lewis, T. (1994) "The Dark Side of Objects," *IEEE Computer*, 27, 12, 6-7.
- Liker, J. K.; Fleischer, M. and Arnsdcrf, D. (1992) "Fulfilling the Promises of CAD," *Sloan Management Review*.
- Liu, C.; Goetze, S. and Glynn, B. (1992) "What Contributes to Successful Object-Oriented Learning?", *OOPSLA '92 Proceedings*.
- Mahajan, V. and Peterson, R. (1985) *Models for Innovation Diffusion*, Sage Publications: Beverly Hills, CA.
- Mahajan, V.; Sharma, S. and Bettis, R. A. (1988) "The Adoption of the M-form Organizational Structure: a Test of the Imitation Hypothesis," *Management Science*, 34, 10, 1188-1201.
- Mansfield, E. (1968) *Industrial Research and Technological Innovation: An Econometric Analysis*, W.W. Norton and Co.: New York.
- Mansfield, E. (1993) "The Diffusion of Flexible Manufacturing Systems in Japan, Europe and the United States," *Management Science*, 39, 2, 149-159.
- Markus, M. L. (1987) "Toward a 'Critical Mass' Theory of Interactive Media: Universal Access, Interdependence and Diffusion," *Communications Research*, 14, 5, 491-511.
- Meyer, A. D. and Goes, J. B. (1988) "Organizational Assimilation of Innovations: A Multilevel Contextual Analysis," *Academy of Management Journal*, 31, 4, 897-923.
- Meyer, B. (1987) "Reusability: The Case for Object-Oriented Design," *IEEE Software*, March 1987, 50-64.
- Mitchell, D. L. (1993) *A Multivariate Analysis of the Effects of Gender and Computer vs Paper/Pencil Mode of Administration on Survey Results*, unpublished Louisiana Tech University PhD. dissertation.
- Nasbeth, L. and Ray, G. F. (ed.) (1974) *The Diffusion of New Industrial Processes: An International Study*, Cambridge University Press, London.
- Nelson, R. and Winter, S. (1982) *An Evolutionary Theory of Economic Change*, Harvard University Press: Cambridge, MA.
- Nilakanta, S. and Scamell, R. W. (1990) "The Effect of Information Sources and Communication Channels on the Diffusion of an Innovation in a Data Base Environment," *Management Science*, 36, 1, 24-40.

Nolan, D. (1986) "Stone Age Programming Cripples 4GL Environment," *COMPUTERWORLD*, November 17, 1986.

Oliver, P.; Marwell, G. and Teixeira, R. (1985) "A Theory of the Critical Mass. I. Interdependence, Group Heterogeneity, and the Production of Collective Action," *American Journal of Sociology*, 91, 3, 522-556.

Oppenheim, A. N. (1992) *Questionnaire Design, Interviewing and Attitude Measurement*, Pinter Publishers: London.

Orlikowski, W. J. (1991) "Radical and Incremental Innovations in Systems Development: An Empirical Investigation of CASE Tools", Working Paper, Sloan School of Management, MIT, Cambridge, Massachusetts.

Page-Jones, M. and Weiss, S. (1989) "Synthesis: An Object-Oriented Analysis and Design Method," *American Programmer*, 2, 7-8, 64-67.

Pascoe, G. A. (1986) "Elements of Object-Oriented Programming," *Byte*, 11, August 1986, 139-144.

Patton, M. Q. (1990) *Qualitative Evaluation and Research Methods*, Sage: Newbury Park.

Pennings, J. M. and Harianto, F. (1992a) "The Diffusion of Technological Innovation in the Commercial Banking Industry," *Strategic Management Journal*, 13, 29-46.

Pennings, J. M. and Harianto, F. (1992b) "Technological Networking and Innovation Implementation," *Organization Science*, 3, 3, 356-382.

Peterson, G., E. (ed.) (1987) *Tutorial: Object-Oriented Computing, Volume II*, IEEE Computer Society Press, Washington D.C.

Pilon, T. L. and Norris, C. C. (1988) "Disk-by-Mail: A New Survey Modality", *Proceedings of the Sawtooth Software Conference on Perceptual Mapping, Conjoint Analysis and Computer Interviewing*, Sun Valley, Idaho, 387-396.

Raho, L. E.; Belohlav, J. A. and Fiedler, K. D. (1987) "Assimilating New Technology into the Organization: An Assessment of McFarlan and Mckenney's Model," *MIS Quarterly*, March 1987, 43-56.

Rai, A. (1993) "External Information Source and Channel Effectiveness and the Diffusion of CASE Innovations: An Empirical Study", *Diffusion Interest Group in Information Technology (DIGIT) Workshop*, Orlando, Florida, December.

Rice, R. and Rogers, E. (1980) "Reinvention in the Innovation Process," *Knowledge: Creation, Diffusion, Utilization*, 1, 4, 499-514.

Roberts, E. B. and Fusfeld, A. R. (1981) "Staffing the Technology-Based Organization," *Sloan Management Review*, 22, 3, 19-34.

Robertson, T. S. and Gatignon, H. (1986) "Competitive Effects on Technology Diffusion," *Journal of Marketing*, 50, July, 1-12.

Rogers, E. and Eveland, J. D. (1978) "Diffusion of Innovations Perspectives on National R&D Assessment: Communication and Innovation in Organizations," in *Technological Innovation: A Critical Review of Current Knowledge*, Kelly, P. and M. Kranzberg (ed.), San Francisco Press: San Francisco.

Rogers, E. M. (1962) *Diffusion of Innovations*, The Free Press: New York.

Rogers, E. M. (1983) *Diffusion of Innovations*, 3rd edition, The Free Press: New York.

Rogers, E. M. (1991) "The 'Critical Mass' in the Diffusion of Interactive Technologies in Organizations," in *The Information Systems Research Challenge: Survey Research Methods, Volume 3*, Kraemer, K. L., J. I. Cash and J. F. Nunamaker (ed.), Harvard Business School Research Colloquium: Boston.

Rosenberg, N. (1976) "On Technological Expectations," *The Economic Journal*, 86, September, 523-535.

Rosenberg, N. (1982) *Inside the Black Box: Technology and Economics*, Cambridge University Press: Cambridge.

Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F. and Lorenzen, W. (1991) *Object-Oriented Modeling and Design*, Prentice-Hall: Englewood Cliffs, NJ.

Russo, M. V. (1991) "The Multidivisional Structure as an Enabling Device: A Longitudinal Study of Discretionary Cash as a Strategic Resource," *Academy of Management Journal*, 34, 3, 718-733.

Saga, V. L. and Zmud, R. W. (1993) "The Nature and Determinants of IT Acceptance, Routinization, and Infusion", *International Federation for Information Processing Conference on Diffusion, Transfer & Implementation of Information Technology*, Champion, Pennsylvania, October, 67-86.

Sahal, D. (1985) "Technological Guideposts and Innovation Avenues," *Research Policy*, 14, April, 61-82.

Saltzman, A. (1993) "Improving Response Rates in Disk-by-Mail Surveys," *Marketing Research*, 5, 3, 32-39.

- Seidler, J. (1974) "On Using Informants: A Technique for Collecting Quantitative Data and Controlling for Measurement Error in Organizational Analysis," *American Sociological Review*, 816-831.
- Silk, A. and Kalwani, M. (1982) "Measuring Influence in Organizational Purchase Decisions," *Journal of Marketing Research*, 19, 165-181.
- Singer, J. D. and Willett, J. B. (1991) "Modeling the Days of Our Lives: Using Survival Analysis When Designing and Analyzing Longitudinal Studies of Duration and Timing of Events," *Psychological Bulletin*, 110, 2, 268-290.
- Slitz, J. and Mintz, J. (1993) "Object Technology 1993-2000: Trends and Expectations", Object Management Group Report.
- Smith, E. (1993) Intelliquest, personal communication.
- Srinivasan, B. and Mason, C. H. (1986) "Nonlinear Least Squares Estimation of New Product Diffusion Models," *Marketing Science*, 5, 2, 169-178.
- Stamps, D. (1989) "CASE vs. 4GLs," *Datamation*, 35, 16, 29-32.
- Stone, J. A. (1993) *Inside ADW and IEF: The Promise and Reality of CASE*, McGraw-Hill: New York.
- Sullivan, C. H. (1985) "Systems Planning in the Information Age," *Sloan Management Review*, 26, 2, 3-11.
- Swanson, E. B. (1994) "Information Systems Innovations Among Organizations," *Management Science*, 40, 9, 1069-92.
- Taylor, D. A. (1990) *Object-Oriented Technology: A Manager's Guide*, Servio Corporation: Alameda, CA.
- Teece, D. (1980) "The Diffusion of an Administrative Innovation," *Management Science*, 26, 5, 464-470.
- Thomas, D. (1989) "What's in an Object?," *Byte*, March 1989.
- Tornatzky, L. G. and Klein, K., J. (1982) "Innovation Characteristics and Innovation Adoption-Implementation: A Meta-Analysis of Findings," *IEEE Transactions on Engineering Management*, EM-29, 1, 28-45.
- Tornatzky, L. G.; Eveland, J. D.; Boylan, M. G.; Hetzner, W. A.; Johnson, E. C. et al. (ed.) (1983) *The Process of Technological Innovation: Reviewing the Literature*, National Science Foundation, Washington D. C.

- Trajtenberg, M. and Yitzhaki, S. (1989) "The Diffusion of Innovations: A Methodological Reappraisal," *Journal of Business and Economic Statistics*, 7, 1, 35-47.
- Tushman, M. L. and Anderson, P. (1986) "Technological Discontinuities and Organizational Environments," *Administrative Science Quarterly*, 31, 439-465.
- Vellante, D. P.; Picardi, A. and McClure, S. (1992) "Opportunities in Object-Oriented Technologies", International Data Corporation.
- Venkatraman, N. (1993) Boston University, personal communication.
- Venkatraman, N. and Ramanujam, V. (1987) "Measurement of Business Economic Performance: An Examination of Method Convergence," *Journal of Management*, 13, 1, 109-122.
- von Hippel, E. (1994) "'Sticky Information' and the Locus of Problem Solving: Implications for Innovation," *Management Science*, 40, 4, 429-439.
- Wirfs-Brock, R.; Wilkerson, B. and Wiener, L. (1990) *Designing Object-Oriented Software*, Prentice-Hall: Englewood Cliffs, NJ.
- Yourdon, E. (1986) "What Ever Happened to Structured Analysis?," *Datamation*, June 1, 1986, 133-138.
- Zaltman, G.; Duncan, R. and Nolbeck, J. (1973) *Innovations and Organizations*, John Wiley & Sons: New York.
- Zmud, R. W. (1982) "Diffusion of Modern Software Practices: Influence of Centralization and Formalization," *Management Science*, 28, 12, 1421-1431.
- Zmud, R. W. (1983) "The Effectiveness of External Information Channels in Facilitating Innovation Within Software Groups," *MIS Quarterly*, (June 1983), 43-56.
- Zmud, R. W. (1984) "An Examination of 'Push-Pull' Theory Applied to Process Innovation in Knowledge Work," *Management Science*, 30, 6, 727-738.
- Zmud, R. W. and Apple, L. E. (1992) "Measuring Technology Incorporation/Infusion," *Journal of Product Innovation Management*, 9, 148-155.