AD-A211 573

DTIC FILE COPY

Technical Report

CMU/SEI-89-TR-17
ESD-TR-89-25

Carnegie Mellon University
Software Engineering Institute

Adoption of Software Engineering Innovations in Organizations

Judy Bayer
Nancy Melone

April 1989

89 8 17 007

Technical Report
CMU/SEI-89-TR-17
ESD-TR-89-25
April 1989

Adoption of Software Engineering Innovations in Organizations

Judy Bayer, PhD
Nancy Melone, PhD

Transition Process Research Project

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
ESD/AVS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

Karl Shingler
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

Table of Contents

Part I: Executive Summary

Part II: Discussion

List of Figures

List of Tables

Part I: Executive Summary

Part II: Discussion

Acknowledgements

We extend our thanks to the organizations and individuals who were our subjects, to the National Security Industry Association, without whose help we might never have met our subjects, and to our colleagues and sponsors at Carnegie Mellon University in the Software Engineering Institute (SEI) and the Graduate School of Industrial Administration. We are indebted to our secretaries, Donna Erdner and Anna Spangler, and our student research assistants, Deana Erdner and Larry Levine. We especially thank Dick Martin at the SEI for his support of this research. This document was published at the SEI by Purvis M. Jackson.

Adoption of Software Engineering Innovations in Organizations

Abstract: Designing effective strategies to facilitate the adoption of new software engineering technologies is a complex endeavor. This document describes the experiences of organizations in the defense industry that have considered and in many cases adopted any one of five software engineering technologies: structured programming, program design languages, software cost models, complexity metrics, and Ada. In all, 296 respondents participated in the entire study. These respondents represented approximately 120 business units within approximately 75 defense contractor organizations. Data were collected using a structured survey instrument administered over the telephone.

This report examines the motivations behind technology acquisition and adoption decisions, the use of various technology transfer mechanisms during the stages of the adoption process, and the relationship between technology transfer mechanisms and the timing, pass through, and smoothness of adoption-process stages. Adoption is assumed to be a multi-stage process that may proceed in a linear or non-linear fashion. Also explored is the relationship between managerial level of the advocate (i.e., top management, middle management, technical management, and broad-based support) and the speed and smoothness of technology acquisition and adoption.

Analysis of data supports the notion that organizations and change agents (e.g., the Department of Defense (DoD)) should carefully tailor transition mechanisms and the choice of technology advocate to the specific stage of the adoption process, rather than adopt a single strategy for the entire process. Moreover, a single adoption strategy is not applicable to all technologies. These strategies must also be tailored depending on the subtleties of the particular technology.

Part I: Executive Summary

This report discusses the initial results of a field study focused on describing and understanding the experiences of over 75 firms in the defense industry that have considered and in many cases adopted any one of five software engineering innovations. Issues of interest in the study include the organizational level and efficacy of the technologies' advocates, the use and efficacy of technology transfer mechanisms, and the perceptions of relative advantage motivating adoption.

1. Introduction

While there is ample research documenting the substantial amount of time between the availability of a new tool and its adoption (e.g., Riddle 1984), considerably less attention has been devoted to evaluating the consequences of specific managerial actions on the realization of technology-transition objectives. As a result, the practitioner has little more to rely on than personal experience or that of colleagues. For the most part, these collective experiences are based on anecdotal evidence rather than systematic observation. While such case studies have value, they are of limited usefulness in generalizing beyond the specific technology or situation in which the observation was made. Hence, any manager who applies case-study findings about innovation-adoption behavior to a new situation is "shooting in the dark," since the (unspecified) factors that influenced an outcome in the single case study may not be operative in the new situation.

Our research on technology transition for the Software Engineering Institute begins to address this problem of generalizing beyond the experience of a single organization or a single software engineering technology by examining the behaviors of multiple firms and multiple software engineering technologies. While this approach is more complex, it enables practitioners to apply such findings to their own situations with greater confidence than can be obtained with case-study or single-innovation research designs.

The research summarized in this paper examines the adoption of five software engineering innovations of varying degrees of maturity, abstractness, and target users (i.e., Ada programming environment, program design languages, structured programming, cost models, and complexity metrics). Using data collected from structured telephone interviews, we examine two areas of concern to managers of technology transition efforts. First, we are particularly interested in how the choice of transfer mechanism (i.e., in-house training, outside training, written documentation, conferences and seminars, and site visits) and primary technology advocate (i.e., top management, middle management, technical staff, broad-based support) relates to three measures of adoption: time of entry into the adoption stage, movement through the stage, and the experienced ease or smoothness of passage. Second, we wish to understand the predominant motives underlying decisions to adopt, postpone, or reject software engineering technologies.

2. The Research Framework

The diffusion of an innovation is conceptualized as a process by which knowledge of an innovation spreads throughout a population, eventually to be adopted or not adopted by a decision-making unit in an organization. According to diffusion theory (Rogers 1983), the degree of acceptance is contingent upon the information about the innovation, characteristics of the adopters of the innovation, and the degree of similarity between technology advocates and potential adopters.

Depending on the innovation, either four or five adoption-process stages are identified in this research. The specific stages reflect our discussions with experienced software engineers and an examination of published materials about technology transition. The stages are:

Pre-acquisition, gathering information and approving/rejecting acquisition of capabilities.

Acquisition of physical capabilities through lease, rental or purchase (only for Ada).

Developing/acquiring human capabilities, training and/or additional hiring.

Trial, using the technology for a test project in order to assess the usefulness of the technology before finally committing the organization to it.

Production, using the technology in a software-production environment, that is, on a large scale. The listing does not presume an order of execution or that organizations go through all intermediate stages in order to finally arrive at production.

3. Research Method

3.1. Participants

Participants in the study were individuals responding on behalf of major software developers and consultants for the DoD. These individuals were knowledgeable about their organization's adoption, postponement, or rejection of the various technologies. In all, 296 interviews from over 120 business units representing 75 firms comprise data for the study. The number of business units represented for each technology are: structured programming (68); program design languages (60); software cost models (61); software complexity metrics (41); and Ada (66). Each business unit was permitted only one participant for each technology; hence a business unit could have a maximum of five participants and a minimum of one.

3.2. The Data-Collection Instrument

Data were collected using a structured survey instrument which was administered over the telephone. The eighteen-page survey posed questions on a broad range of issues related to the adoption of software engineering tools and methods. An overview of some of those data is reported in this paper. The survey contained questions requiring closed-form responses primarily. For example, participants were asked about the extent to which they used various transition mechanisms at a particular adoption stage. They responded using a 7-point scale representing a range from "1" (not at all) to "7" (to a very great extent). Similarly, participants were asked to supply dates when specific events took place. Participants were also given ample opportunity to comment on their responses.

3.3. Procedure

Data were collected using a telephone survey lasting approximately 35 minutes. Prior to data collection, participants were telephoned to verify qualifications, answer questions, and schedule the interview(s). Approximately one week before the telephone interview was to take place, the participant was sent a copy of the survey questions. Interviews were conducted by individuals who had undergone six hours of telephone-interview training. Interviewers were paid for their work. Following the interview, a thank-you letter was sent and the participant was told that he or she would receive an executive summary of the study's findings once the data had been collected and analyzed. The interviews were completed over a three-month period during the spring of 1988.

4. Highlights of the Results

Highlights of the results are presented. First, we examine the relationship between managerial level of the technology advocate (i.e., top management, middle management, technical staff, and broad-based support) and adoption outcomes (i.e., timing, movement, and smoothness). Next, we present results relating the choice of technology transfer mechanism and adoption outcomes. Finally, we examine the motives for adoption of software engineering innovations.

4.1. The Effects of the Technology Advocate on Adoption

Top management has often been viewed in the literature as the preferred advocate for facilitation of adoption. For our sample this was clearly not the case. Overall, across technologies and adoption stages, top management advocacy was not strongly correlated with timing, pass-through or smoothness of adoption with the exception of the trial stage of adoption, and even for trial the results are mixed. Top management advocacy was positively associated with earlier and smoother use of cost models during the trial stage, smoother use of structured programming in trial, earlier use of Ada in trial but failure to complete trial, and failure to complete trial for program design languages. Hence, our results contradict previous studies.

The effect of middle-management advocacy reveals a slightly different set of results as shown in Table 1. The table shows the association [positive (+), negative (-), or no relationship (0)] of

	Movement	Timing	Ease
SP	+ develop capabilities - production	- develop capabilities - trial - production	0
PDL	+ develop capabilities	0	0
SCM	+ develop capabilities	0	0
CM	+ develop capabilities + trial + production	+ develop capabilities	0
Ada	+ develop capabilities - trial - production	- trial + production	0

Table 1: Middle Management Advocate's Effect on Adoption of Technologies

middle-management advocacy with the adoption criteria (i.e., movement, timing and ease of adoption) during the various stages of adoption (e.g., trial). When these results are examined across the three adoption criteria, middle-management advocacy has a positive association with movement and a somewhat negative association with early entry into the stages. This level of management advocacy appears to have no association with smoothness of passage for any of the technologies. An interesting pattern emerges for this group to the extent that structured programming and program design languages can be considered more technically-oriented innovations and software cost models and complexity metrics can be considered more administrative innovations. As might be predicted, middle-management advocacy is positively associated with the adoption of administrative innovations but has a mixed record with regard to the adoption of technical innovations. Also, to the extent there is a stage of adoption at which middle management advocacy has benefit for all technologies, it is in getting the organization through the development of human capabilities.

Table 2 reveals strong significant positive associations between technical staff advocacy and completion of adoption stages (movement) and strong negative associations between technical staff advocacy and smoothness of passage (ease). Technical staff advocacy has either no effect or a negative effect on early entry into a stage.

	Movement	Timing	Ease
SP	+ develop capabilities	0	- trial - production
PDL	+ develop capabilities + trial	0	- develop capabilities
SCM	+ develop capabilities	0	0
CM	+ develop capabilities - trial	- develop capabilities	- develop capabilities - trial - production
Ada	+ compiler acquisition + develop capabilities + trial + production	- production	- develop capabilities + trial

Table 2: Technical Staff Advocate's Effect on Adoption of Technologies

As can be seen from Table 3, broad-based support has the most wide-spread, positive impact on the adoption of software engineering innovations than any other form of advocacy. In only one stage (trial) for one technology (Ada) is broad-based support negatively associated with an adoption criterion (ease). These results underscore the value of gaining such advocacy in promoting new software engineering technologies.

4.2. The Effects of Transfer Mechanism on Adoption

Organizations may use a variety of transition mechanisms to facilitate the adoption of software engineering innovations. These mechanisms may be more or less effective as an organization

	Movement	Timing	Ease
SP	+ develop capabilities + production	+ develop capabilities + trial + production	+ develop capabilities + trial + production
PDL	+ develop capabilities	0	+ develop capabilities + production
SCM	+ trial + production	+ develop capabilities	+ production
CM	+ develop capabilities	0	+ develop capabilities
Ada	+ trial + production	+ compiler acquisition + trial	- trial

Table 3: Broad-Based Advocates' Effect on Adoption of Technologies

passes through the various adoption-process stages. The transition mechanisms of interest in the current study are: (1) training prepared by in-house personnel; (2) training prepared by outside personnel; (3) written documentation and published technical materials; (4) attendance at conferences and seminars; and (5) site visits to other organizations. We briefly report the results for the five software engineering technologies of interest.

Overall, across technologies, adoption stages, and adoption criteria, extensive use of training prepared by in-house personnel has the greatest positive association with movement and timing of adoption. The details are displayed in Table 4. Interestingly, in-house training does not appear to be associated with making passage through adoption stages any easier or smoother.

In contrast to training prepared in-house, training prepared outside the organization does not have the same overall positive association with adoption criteria. These results are presented in Table 5.

Table 6 summarizes the effects of written documentation and published materials on the adoption of the five software engineering innovations. In many instances, the use of written materials bears no relationship to the criteria of adoption. In other cases, particularly the adoption of Ada, there are predominantly positive associations with adoption. For complexity metrics, the extensive use of written documentation appears to delay entry into and movement through production. Curiously, it is also associated with a smoother production process.

Site visits are infrequently used. In the cases in which use is positively associated with adoption, visits are more typically employed for innovations that are methodologically-oriented (e.g., structured programming). Site visits are also associated with facilitating trial.

In general, more detailed analysis of the data suggests that when extensive use of a transition mechanism is effective, it should be initiated early and continued through trial.

	Movement	Timing	Ease
SP	0	+ production	0
PDL	+ develop capabilities + production	+ develop capabilities	+ production
SCM	+ develop capabilities + trial	- develop capabilities	0
CM	+ develop capabilities	+ production	0
Ada	+ trial + production	+ compiler acquisition + develop capabilities + trial + production	+ develop capabilities

Table 4: Effects of Extensive Use of Training Prepared by In-House Staff Across Technologies

	Movement	Timing	Ease
SP	+ trial	- develop capabilities - trial	- trial - production
PDL	+ trial - production	0	+ production
SCM	0	+ develop capabilities + trial	0
CM	0	0	- develop capabilities
Ada	+ trial	- develop capabilities - compiler acquisition - production	- compiler acquisition + trial

Table 5: Effect of Extensive Use of Training Prepared by Outside Personnel Across Technologies

4.3. Perceived Relative Advantages of Adoption

Overall, beliefs about the economic advantages of adopting innovations, such as increasing likelihood of obtaining government contracts, or disadvantages clearly have significant positive and negative associations with adoption across all of the technologies studied. Perceptions about economic incentives have their most extensive impact on the use of program design languages and structured programming. Beliefs about training difficulties or the resistance of technical staff appear to have more impact on ease or smoothness rather than timing or movement

	Movement	Timing	Ease
SP	0	+ develop capabilities + production	0
PDL	0	0	- trial + production
SCM	+ trial	0	0
CM	+ develop capabilities - production	- production	+ production
Ada	+ trial + production	+ compiler acquisition + develop capabilities + trial	- production

Table 6: Effect of Extensive Use of Written Documentation Across Technologies

of adoption. Such impacts are strongest for structured programming and use of program design languages.

Other factors which were significantly associated with adoption and should be taken into consideration during technology transition are: (1) the prestige associated with adoption leading to perceptions of leadership or innovativeness of the firm, (2) the compatibility of the technology with either the mission or the technical culture of the organization, and (3) the nature of interpersonal communication among software engineers within and outside the organization.

4.4. Cautionary Note

Because of the nature of the data (i.e., cross-sectional), we can make no strong causal assertions with regard to the effect of choice of transition mechanism and advocate on dependent measures of adoption. However, strong association suggests possible causal hypotheses which should be explored.

5. Conclusion

An objective of this study was to enable practitioners to more fully understand the factors and processes that influence adoption, postponement, or rejection of a variety of software engineering innovations across a large number of organizations. The analysis examined the effect of the level in the organization of the primary advocate on the adoption process. Broad-based support was found to result in a number of positive associations with adoption. The authors also found that different factors are often related to adoption of the innovations at different stages. Transition mechanisms and perceived relative advantages of the innovation which facilitate adoption at one stage do not necessarily have the same effect at other stages.

Part II: Discussion

1. Introduction

The excessive amount of time between the availability of software engineering innovations and their adoption is well established (Riddle 1984). This lag is a serious concern for promoters of new software engineering tools and methods (e.g., DoD, Software Productivity Consortium) and managers in software development firms who are responsible for facilitating the use of these innovations. Both groups perceive a need to understand the adoption process so that strategies can be designed to yield optimal rates and levels of adoption.

Despite the urgency of this problem, the practitioner has little more to rely on than personal experience and the experience of others. These collective experiences are largely based on anecdotal evidence rather than systematic observation. Although such case studies are of some value to the practitioner, they are of limited usefulness in generalizing beyond the specific situation in which the observation is made. Thus, any manager or organization that applies case-study findings about innovation adoption behavior to a new situation is "shooting in the dark" since the (unspecified) factors that influenced an outcome in the case study may not be operative in the new situation.

Our research addresses this problem of generalizability by examining the process by which a large number of organizations make decisions to reject or to integrate "new" software engineering innovations into their operations. In this report, we focus on understanding the factors and processes that influence adoption, postponement, or rejection of innovations, and the smoothness of the organizational process. Because a specific software engineering innovation may have characteristics that make it easier to integrate than another, we examine five innovations of various levels of maturity and abstraction. Hence, our findings are more likely to generalize across technologies as well as organizations. We believe that this approach will enable practitioners to apply our findings to their own situations with greater confidence than they could the findings from previous case-study and single-innovation research.

The five innovations of interest in this study are:

- Structured programming techniques
- Program design languages
- Software cost models
- Software complexity metrics
- Ada

This report presents an analysis of 296 telephone surveys representing approximately 75 government contractors and 120 business units. The report gives details of analyses which focus on the adoption of the five software engineering innovations across multiple stages of adoption. Depending on the innovation, either four or five adoption stages are identified.

The specific stages reflect our discussions with experienced software engineers and an examination of published materials. The stages are:

- **Pre-acquisition**, gathering information and approving/rejecting acquisition of physical capabilities.
- **Acquisition of physical capabilities** through lease, rental or purchase.
- **Developing/acquiring human capabilities**, training and/or additional hiring.
- **Trial**, using the technology for a test project in order to assess the usefulness of the technology before finally committing the organization to it.
- **Production**, using the technology in a software-production environment, that is, on a large scale.

Acquisition of physical and human capabilities is broken up into separate stages for only one of the technologies: Ada. In this report we focus on the association of the primary advocate, as well as the effectiveness of various transition mechanisms an organization may use, and the adoption process with respect to the five innovations.

2. Overview

2.1. The Research Framework

The diffusion of a software engineering innovation is conceptualized as the process by which knowledge of an innovation spreads throughout a population, eventually to be adopted or not adopted by an individual or other decision-making unit in an organization. According to diffusion theory (Rogers, 1983), the degree of acceptance and the rate at which this process takes place is contingent upon the characteristics of the innovation, networks used to communicate the information about the innovation, characteristics of the adoptees of the innovation, and the degree of similarity between change agents and potential adoptees. This concept of innovation diffusion has been applied to technologies ranging from new ideas to new machines (Teece, 1980; Zmud, 1982, Zmud and Apple, 1986). Since this framework is used in the collection and analysis of data for this study, we discuss these components (i.e, innovation, communication, timing, and social system) in more detail before presenting the analysis for each technology innovation.

2.2. Innovation

An innovation is anything that appears to be new to the individuals or the organizations within a social system. Thus, an innovation can be a new idea, such as data hiding, a new of way of doing things, such as using structured programming, or a new hardware technology. Theories of innovation diffusion assert that characteristics of an innovation either facilitate or inhibit its adoption. Characteristics of the innovation which have received empirical support (Tornatzky and Klein, 1982) include:

- The relative advantage of the innovation over adoption of alternative technologies or non-adoption (e.g., advantage derived from economic, social prestige, convenience, or satisfaction aspects of the innovation) — the compatibility of the innovation with existing values, past experiences, or needs of individuals or organizations;

- The perceived complexity of the innovation — innovations that are easy to understand are adopted more rapidly than those that are difficult to understand;

- The trialability of the innovation — the ability to use an innovation on a trial or partial basis lowers the risk of adoption and, thus, tends to encourage adoption;

- The observability of the innovation or its outcomes — intangible innovations such as new software development philosophies tend to be adopted more slowly than more visible innovations such as hardware/software-based innovations.

Each of the five technologies is examined from these perspectives in Chapters 4 through 8.

2.3. Communication

Communication is the creation and sharing of information about innovations. Information moves from a source that knows about the innovation, through one or more communication channels (e.g., mass media such as technical journals, or interpersonal or other informal channels such as vendors, consultants, or electronic bulletin boards), to an individual who or organization that does not yet have knowledge of the innovation. These communication channels can be enhanced when the source of the communication (e.g., a trainer) is more similar to the target of the communication (e.g., a user). The likelihood exists that at different stages in the organization's adoption of the innovation, different sources, channels and targets of information may be appropriate.

2.4. Timing

Rate of adoption, the relative speed with which the innovation is adopted, has been shown empirically to follow an s-shaped curve. The curve receives strong empirical support in the literature. Rather than focus on this aspect of the timing issue which has already been well documented in the literature, the current research emphasizes the timing of the movement through the various adoption stages and the use of different transition mechanisms (e.g., in-house training, visits to other sites) throughout those stages. For purposes of this study, we identify four process stages through which an organization may move prior to the adoption of an innovation. These stages are the period of information gathering before acquisition of the technology (pre-acquisition), the period after acquisition of the technology in which capability is developed (learning), the period in which the technology is tried out on a small scale (pilot), and the period in which the technology is fully employed on a large scale (production). In the case of Ada, we identify a fifth stage: acquisition of physical capabilities. In addition, we inquire about the relative amount of the firm's resources used in employing these mechanisms.

2.5. The Social System

The social system is a group of interrelated individuals or organizations using collective problem-solving methods to achieve a common purpose. Software engineers are members of a number of overlapping social systems that may influence adoption behavior. The employing organization is a social system that may authorize (or inhibit) adoption of an innovation. Other organizations outside the adopting organization (e.g., DoD) may also have an influence on the adoption behavior of the employing organization. Professional peer groups or the cadre team to which the software engineer belongs may emphasize relative advantages (or disadvantages) associated with adopting the innovation. The rate and form of adoption is likely to be influenced by the existence of opinion leaders within the organization and efforts of external change agents interacting with the social system in the target organization.

This version of the preliminary report does not analyze data pertaining to social systems. However, it does present preliminary analyses of the efficacy of various sources of support (e.g., top management, technical staff, middle management) on the various stages of the adoption process.

3. Research Method

3.1. Subjects

Participants in this phase of the research program were individuals responding on behalf of major software developers and consultants for the government. These individuals were knowledgeable about their organization's adoption, postponement, or rejection of the various technologies of interest to this study. Participants for this study were informed of the study through a letter sent by the two principal investigators to NSIA members (see Appendix B). Approximately one month later an announcement describing the study appeared in the NSIA newsletter. Response was such that no additional follow-up letters were sent. Members of the NSIA represent major defense contractors in the U.S. as well as small consulting firms and developers.

In the initial solicitation letter, the study was described and individuals were asked to identify people in their firms (i.e., business unit) who had knowledge of the adopt, reject, or postpone decisions related to any of the five technologies of interest. The initial contacts returned a form indicating who in their unit would participate and for which technologies. In some cases, the participant was the addressee; in most cases, they were other people in the organization. Each business unit was permitted only one participant for each technology; hence, a business unit could have a maximum of five participants. In most cases, a single participant was knowledgeable about more than one technology, so a business unit usually had fewer than five people represented in the study. In the cases in which the participant was not the initial contact, a short letter was sent describing the study and who in the organization had given his or her name as a participant.

3.2. Procedure

Data were collected using a telephone survey. Prior to data collection, participants were telephoned to verify information, answer questions that the participant might have about the study, and to schedule the telephone interview(s). In cases where a single participant was responding to questions about more than one innovation, multiple appointments were made. In most cases, data were collected in a single interview lasting approximately one-half hour. In the remaining cases, generally no more than two appointments were needed.

Approximately one week before the telephone interview was to take place, the participant was sent a copy of the survey questions. Following the interview, the participant was sent a thank-you letter. The participant was told he or she would receive an executive summary of the study's findings once the data had been collected and analyzed.

The telephone interviews were conducted by advanced graduate students in management who were interested in the problems of technology innovation. Each interviewer followed a written script that corresponded to the questionnaire sent to participants. Prior to conducting any interviews, each interviewer underwent six hours of telephone-interview training. The interviews were completed over a three-month period during the spring of 1988.

3.3. Overview of the Analysis

The remaining portions of this report present the analysis of each technology. The order of presentation is:

- Structured programming (SP)
- Program design languages (PDL)
- Software cost models (SCM)
- Complexity metrics (CM)
- Ada

The reader should note that the technologies were chosen so that two (SP and PDLs) are targeted to individual software engineers, two (SCM and CM) are administrative aids, two (PDLs and SCM) are tools, and two (SP and CM) are primarily intangible methods. Ada cuts across each dimension. The analysis for each technology is presented as a self-contained unit to allow readers to look at specific technologies of interest. Unfortunately, this method of presentation creates redundancy in the text. We apologize for this in advance. Also, in this report we do not make explicit comparisons of innovations, although we summarize the findings across technologies in Chapter 9.

4. Structured Programming

Sixty-eight participants responded to questions about their organization's (business unit) use of structured programming techniques. This innovation differs from other innovations studied in that: 1) it is more mature than other software engineering innovations, 2) it is a methodology rather than a tool, and 3) the primary user of the technology is the individual software engineer.

Table 4-1 shows the percentage of our sample population of organizational units that have passed through each stage of the adoption process for structured programming. For this technology, the stages were:

1. Pre-acquisition, in other words going through an approval process for using structured programming within the organization.
2. Developing structured programming capabilities, that is, those tasks which enable the organization to use structured programming, such as training and/or hiring personnel.
3. Using structured programming for a pilot or test project in order to assess the usefulness of the technology before finally committing the organization to it.
4. Using structured programming in a production environment, for any complete software-development projects, rather than on a trial basis.

Stage	Percentage
Pre-acquisition	100%
Develop capabilities	85%
Trial	43%
Production	81%

Table 4-1: Percentage of Organizations That Have Passed Through Each Stage of the Adoption Process for Structured Programming.

The table should be read as follows: of the total sample of participants, 43% of the organizations passed through trial.

Clearly, for structured programming, a methodological innovation, organizations often do not try out the technology in a limited-use situation before using it in a full-production environment. The reader should note that 100% of the organizations will always have passed through the pre-acquisition stage (they will have considered adopting the innovation), since this was used as a pre-screening criterion.

Table 4-2 shows both average time and the range of time at which organizational units passed through each stage of the adoption process for structured programming. Participants were asked for the year in which structured programming was first adopted (used or capabilities developed) in the organization, by stage.

Stage	Average Time	Range
Develop capabilities	early 1980	1969-1987
Trial	early 1980	1970-1987
Production	mid 1980	1970-1988

Table 4-2: Time of Adoption by Stage

Note that, overall, there is not much spread found between the time organizations first decided to develop capabilities for using structured programming and the time it was first used in a production environment (an average of slightly more than a year elapsed between developing capabilities and use in production). This may occur because using structured programming initially in a production environment involves relatively low amounts of risk versus many other innovations. Adoption involves no capital expenditure, and, since it is a mature technology, late adopters may already be well informed as to the benefits of using structured programming.

4.1. Primary Advocate for Developing Structured Programming Capabilities

One of the objectives of this study was to determine the extent to which different levels in the organization served as the primary advocate for developing capabilities for using structured programming. Later we analyze the effects of support from different organization levels on the adoption process. Figure 4-1 shows the percentage of organizational respondents who indicated that the primary advocate for using structured programming was 1) top management, 2) middle management, 3) technical staff, or 4) broad-based support of technical management or staff.

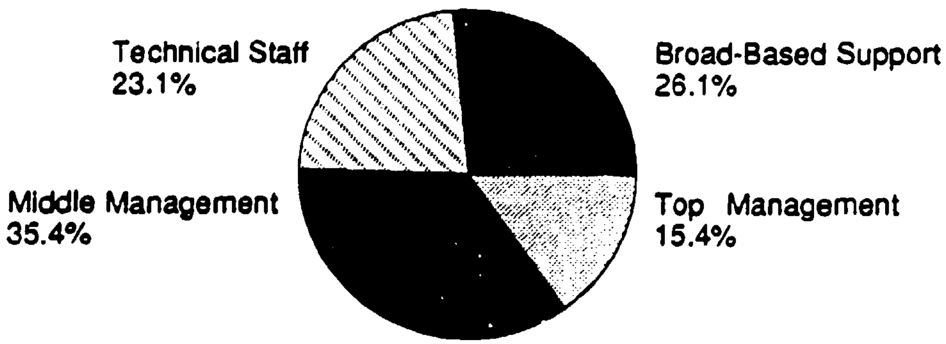

Figure 4-1: Primary Advocate for Developing Structured Programming Capabilities

4.2. Organizations' Use of Different Transition Mechanisms

Another aspect of this research was the investigation of organizations' use of various transition mechanisms to aid in the transference of structured programming within the organization. The transition mechanisms are:

1. Structured programming training prepared by in-house personnel.
2. Structured programming training prepared by outside personnel.
3. Sending personnel to seminars or conferences.
4. Providing written documentation about structured programming or articles about structured programming from technical or scholarly journals.
5. Visits to other organizations where structured programming is used.
6. Tools to aid transition.

Table 4-3 shows, overall, the percentage of organizations' resources allocated to the different transition mechanisms during the structured programming adoption process, as well as the range found across organizations.

Transition Mechanism	Mean %	Range
Training prepared by in-house personnel	42.5	0-100
Training prepared by outside personnel	9.0	0-70
Seminars & conferences	10.4	0-40
Written documentation	22.8	0-80
Visits to other organizations	2.9	0-25
Tools to aid transition	12.9	0-70

Table 4-3: Percentage of Organizations' Resources Allocated to Different Transition Mechanisms During Structured Programming Adoption Process

Table 4-3 shows the percentage of organizational resources used by the different transition mechanisms. Because of the difference in cost and effort to the organization involved in using these mechanisms, it does not tell us the extent to which each mechanism was used during the adoption process. We therefore asked participants to tell us to what extent their own organization provided each of these several types of transition mechanisms to users of structured programming, at each stage in the adoption process. Participants answered by responding with any number from 1 to 7, where 1 means "the mechanism was not at all provided", and 7 means "the mechanism was provided to a very great extent."

Figure 4-2 shows the mean responses for each transition mechanism at each stage.

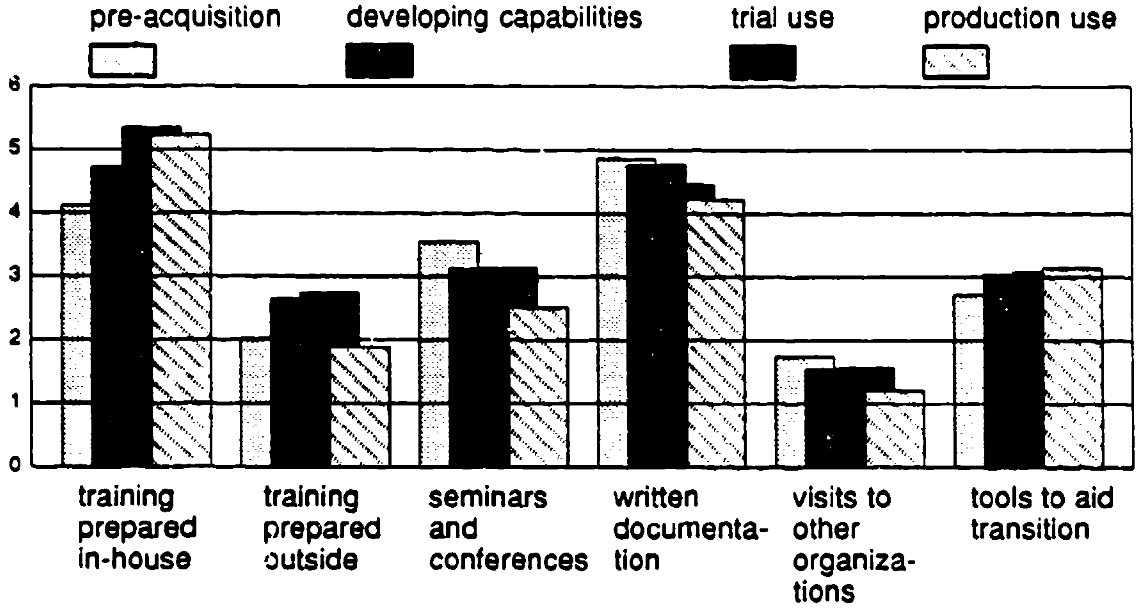

Figure 4-2: Extent to Which Various Transition Mechanisms Are Used at Different Stages in the Adoption Process for Structured Programming

Within each stage, comparing the different transition mechanisms, note that there are significant differences in the extent to which different transition mechanisms are used within each stage. During pre-acquisition and developing capability stages, written documentation is the most used transition mechanism. During trial and production stages, training prepared by in-house personnel is the most used transition mechanism. Visits to other organizations are used to the least extent of any transition mechanisms, at each stage.

Comparing each transition mechanism across stages, we can see that:

- Training prepared by in-house personnel are used most during trial and least during the pre-acquisition stage.
- Training prepared by outside personnel is used most during trial and least during production stage.
- Seminars and conferences are used most during pre-acquisition and least during production ($p < .01$).
- Written documentation is used most during pre-acquisition and least during production ($p < .05$).
- Visits to other organizations are used most during pre-acquisition and least during production ($p < .05$).

- Tools to aid transition are used most during production and least during pre-acquisition ($p < .1$).

4.3. The Adoption Process and the Determinants of Adoption of Structured Programming

Analyses were done to enable us to better understand the determinants of three types of adoption phenomena related to the adoption of structured programming:

- Whether or not the organization has passed through an adoption stage (the adopt or not adopt decision for each stage in the adoption process).
- The smoothness of the adoption process, at each stage.
- The time at which the organization initially enters each stage in the adoption process (early versus later adoption behavior).

4.3.1. Pass Through Adoption Stage

The diffusion literature has often modeled and empirically tested the organization's adopt or not-adopt decision for the innovation. In this study, we expand the empirical analysis by separating the adoption decision by stage in the process. As described previously, the stages used in the structured programming portion of the study are 1) pre-acquisition, 2) developing capabilities, 3) trial use of structured programming, and 4) use of structured programming in a production environment. Whether or not an organization has passed through one of these adoption phases is conceptualized as being a function of four classes of variables:

1. The organizational-level support of the primary advocate for the development of structured programming capabilities.
2. The organization's beliefs about the relative advantages of using structured programming.
3. Whether the organization has passed through an earlier stage in the adoption process. For example, trial is not a necessary precondition for using structured programming in full production. However, we would hypothesize that going through trial would increase the probability of going through production.
4. The time at which the organization passed through an earlier stage. For a mature technology such as structured programming, we would predict that those organizations that passed through the prior stages at an earlier time would be more likely to have passed through later stages.

As Table 4-1 shows, most of the organizations in our sample population have already developed capabilities for using structured programming. Therefore, we limited the analysis to 1) adopting structured programming in a limited-use, trial situation and 2) adopting structured programming for a regular production project.

4.3.1.1. Adopting Structured Programming in a Trial Situation

Primary Advocate

When the primary advocate for developing structured programming capabilities was top management, the organization was more likely to pass through trial ($p < .1$). When the primary advocate for developing structured programming capabilities was middle management, the organization was less likely to pass through trial ($p < .05$).

Perceived Relative Advantages of Structured Programming

a. In those organizations where colleagues in other organizations told them about the advantages of using structured programming, the organization was more likely to pass through trial ($p = .05$).

b. When competitors were developing structured programming capabilities, organizations were more likely to pass through trial ($p < .05$).

c. Belief that use of structured programming is appropriate for software engineering tasks is associated with an organization being less likely to have used structured programming for trial ($p < .05$).

d. Belief that structured programming is more appropriate for military than commercial applications is associated with an organization being more likely to have used structured programming for trial ($p < .05$).

e. Belief that use of structured programming does not yield sufficient economic benefits is associated with an organization being more likely to have used structured programming for trial ($p < .01$).

Note that the above beliefs support the notion that trial use of an innovation may be thought of as a device for reducing risk and uncertainty.

Pass Through Earlier Stage

This was not applicable since most of the organizations had developed structured programming capabilities.

Timing of Passing Through Earlier Stage

Timing was not statistically significant in this analysis.

4.3.1.2. Adopting Structured Programming in a Production Situation

Primary Advocate

The organizational level of the primary advocate for developing structured programming capabilities was only marginally significant in this analysis. When the primary advocate for developing structured programming capabilities was from middle management, the organization was less likely to use structured programming in a production environment. When there was broad-based support, the organization was more likely to use structured programming for production ($p < .1$).

Perceived Relative Advantages of Structured Programming

a. In those organizations where software engineers working in the organization told people in the organization about the desirability of using structured programming, the organization was more likely to pass through production ($p < .05$).

b. When it was believed that use of structured programming is compatible with software engineering practice in the organization, the organization was more likely to use structured programming for production ($p < .05$).

c. Belief that organizations that develop structured programming capabilities within the next year will be perceived as being leaders in software development is associated with an organization being less likely to have used structured programming for production ($p < .05$). This apparent anomaly can probably be explained by the maturity of the technology.

d. Belief that structured programming may have technical problems which should be ironed out is associated with an organization being less likely to have used structured programming for production ($p < .01$).

e. Belief that maintenance costs of software developed using structured programming is unacceptably high is associated with an organization being less likely to have used SP for production ($p < .01$).

f. Belief that structured programming does not yield sufficient economic benefits for the company is associated with an organization being less likely to have used structured programming for production ($p < .05$).

g. Belief that technical staff are heavily committed to old software development methods which they feel work very well for them is associated with an organization being less likely to have used structured programming for production ($p < .01$).

h. Belief that technical staff are skeptical about the technical value of using structured programming is associated with an organization being less likely to have used structured programming for production ($p < .01$).

i. Belief that technical staff have no motivation to adopt structured programming since benefits would be realized only at corporate level is associated with an organization being less likely to have used structured programming for production ($p < .01$).

j. When there is a belief that technical staff feel that they are being used as "guinea pigs in a management or government experiment," then the organization is less likely to have used structured programming for production ($p < .01$).

Pass Through Earlier Stage

Having passed through trial was not a contributing factor.

Timing of Passing Through Earlier Stage

The earlier the organization developed structured programming capabilities, the more likely it was to pass through production ($p < .01$).

4.3.2. Smoothness of the Adoption Process

The second dependent adoption measure analyzed for structured programming is the "smoothness" of the adoption process at each of the previously described stages. The smoothness of the adoption process at a specific stage is conceptualized as a function of the following:

a. The extent to which the organization has used various transition mechanisms, both at this stage, and at earlier stages in the adoption process.

b. The organizational-level support of the primary advocate for the development of structured programming capabilities.

c. The smoothness of the process of passing through earlier stages; a smoother adoption process at earlier stages (e.g., when structured programming capabilities are being developed) should lead to a smoother adoption at later stages.

d. The time at which the organization is passing through this stage in the adoption process.

e. The time at which the organization passed through earlier stages in the adoption process.

An analysis of smoothness of adopting structured programming at different phases was done for three phases: 1) developing structured programming capabilities, 2) trial, and 3) production.

4.3.2.1. Smoothness of Developing Structured Programming Capabilities

Use of Different Transition Mechanisms

During the pre-acquisition stage, use of visits to organizations where structured programming is used is associated with a smoother adoption process ($p < .05$). While structured programming capabilities are being developed, more extensive use of written documentation is associated with a less smooth adoption process ($p = .05$).

Primary Advocate

When the primary advocate for developing structured programming capabilities is made up of broad-based support, then developing structured programming capabilities tends to be smoother ($p < .01$).

Smoothness at Earlier Stage

Not applicable to this analysis.

Time Organization Passed Through This Stage

Timing of developing structured programming capabilities did not significantly affect smoothness.

Time Organization Passed Through Earlier Stages

Not applicable to this analysis.

4.3.2.2. Smoothness of Using Structured Programming in Trial

Use of Different Transition Mechanisms

During the pre-acquisition stage, more extensive use of visits to organizations where structured programming is used is associated with a smoother adoption process ($p < .05$). While structured programming capabilities are being developed, more extensive use of structured programming training prepared by outside personnel is associated with a less smooth adoption process ($p < .01$). Sending personnel to seminars and conferences during this stage, however, is associated with a smoother adoption process ($p < .01$). Sending personnel to seminars and conferences is associated with smoother use of structured programming in trial ($p < .05$).

Primary Advocate

When the primary advocate for developing structured programming capabilities is top management ($p < .05$) or broad-based support ($p < .1$) then using structured programming in a trial situation tends to be smoother. When the primary advocate is someone from technical staff, then the adoption process tends to be less smooth ($p < .05$).

Smoothness at Earlier Stage

Smoother development of structured programming capabilities is associated with smoother use of structured programming in trial situations ($p < .01$).

Time Organization Passed Through This Stage

Timing of using structured programming in trial situations did not affect smoothness.

Time Organization Passed Through Earlier Stages

Timing of developing structured programming capabilities did not affect smoothness.

4.3.2.3. Smoothness of Using Structured Programming for Production

Use of Different Transition Mechanisms

The only transition mechanism which affected smoothness of using structured programming for production tasks was extensive use of training prepared by outside personnel. When used extensively during the trial or production stage, this is associated with a less smooth adoption process ($p < .05$).

Primary Advocate

When the primary advocate was a member of technical staff, use of structured programming for production tended to be less smooth ($p < .05$). Broad-based support is associated with smoother use of structured programming for production ($p < .05$).

Smoothness at Earlier Stage

Smoother development of structured programming capabilities ($p < .01$) and smoother use of structured programming in trial ($p < .01$) is associated with smoother use of structured programming in a production environment.

Time Organization Passed Through This Stage

Timing of using structured programming in production situations did not affect smoothness.

Time Organization Passed Through Earlier Stages

Timing of passing through earlier phases did not affect smoothness.

4.3.3. Timing of Initial Entry Into Stage

The third type of dependent measure analyzed in this study is the organization's timing of adoption of structured programming at each phase in the adoption process. Time of adoption is conceptualized as being a function of the following sets of variables:

1. the organizational-level support of the primary advocate for the development of structured programming capabilities;
2. the organization's beliefs about the relative advantages of using structured programming;
3. the time at which the organization passed through earlier stages in the adoption process; and
4. the smoothness of the process of passing through earlier stages—a smoother adoption process at earlier stages should lead to earlier entry into later stages.

4.3.3.1. Time Structured Programming Capabilities Were Developed

Primary Advocate

Organizations in which the primary advocate for developing capabilities is middle management tend to develop structured programming capabilities later than other organizations ($p < .01$). Broad-based support is associated with earlier development of structured programming capabilities ($p < .05$).

Perceived Relative Advantages of Structured Programming

a. Belief that organizations that develop structured programming capabilities within the next year will be perceived as being leaders in software development is associated with an organization developing structured programming capabilities later ($p < .05$).

b. Belief that structured programming may have technical problems which should be ironed out is associated with later development of structured programming capabilities ($p < .01$).

c. Belief that training costs to instruct users of structured programming are steep is associated with later development of structured programming capabilities ($p < .01$).

d. Belief that the cost-to-benefit ratio of adopting structured programming is less favorable to the adopting company than outside developers realize is associated with later development of structured programming capabilities ($p < .01$).

e. Belief that maintenance costs of software developed using structured programming is unacceptably high is associated with an organization developing structured programming capabilities later ($p < .01$).

f. Belief that production pressures are such that technical personnel cannot easily take time to learn structured programming methods is associated with later development of structured programming capabilities ($p < .001$).

g. When there is a belief that developing capabilities for using structured programming interferes with ongoing development processes then the organization is likely to develop structured programming capabilities later ($p < .01$).

Time Organization Passed Through Earlier Stages

Not applicable to this analysis.

Smoothness of Passing Through Earlier Stages

Not applicable to this analysis.

4.3.3.2. Time Structured Programming Was Used for Trial

Primary Advocate

When the primary advocate for developing structured programming capabilities was from middle management, the organization tended to use structured programming for trial later ($p < .05$). When there was broad-based support, the organization tended to use structured programming for trial earlier ($p < .05$).

Perceived Relative Advantages of Structured Programming

a. When software engineers working in the organization told people about the desirability of using structured programming, then the organization tended to use structured programming for a trial project earlier ($p < .01$).

b. Belief that organizations that develop structured programming capabilities within the next year will be perceived as being leaders in software development is associated with an organization using structured programming for trial later ($p < .05$).

c. Belief that structured programming may have technical problems which should be ironed out is associated with later use of structured programming for trial ($p < .001$).

d. Belief that training costs to instruct users of structured programming are steep is associated with later use of structured programming for trial ($p < .001$).

e. Belief that the cost-to-benefit ratio of adopting structured programming is less favorable to the adopting company than outside developers realize is associated with later use of structured programming for trial ($p < .05$).

f. Belief that maintenance costs of software developed using structured programming is unacceptably high is associated with an organization using structured programming for trial later ($p < .01$).

g. Belief that technical staff are skeptical about the technical value of structured programming is associated with later use of structured programming for trial ($p < .05$).

h. Belief that production pressures are such that technical personnel cannot easily take time to learn structured programming methods is associated with later use of structured programming for trial ($p < .01$).

i. When there is a belief that developing capabilities for using structured programming interferes with ongoing development processes then the organization is likely to use structured programming for trial later ($p < .01$).

Time Organization Passed Through Earlier Stages

The earlier an organization develops structured programming capabilities, the earlier it uses structured programming in a trial situation ($p < .001$).

Smoothness of Passing Through Earlier Stages

Smoothness of developing structured programming capabilities did not affect time of trial.

4.3.3.3. Time Structured Programming Was Used in Production

Primary Advocate

Organizations in which the primary advocate for developing structured programming capabilities is middle management tend to use structured programming for a production project later than other organizations ($p < .01$). When there is broad-based support, the organization tends to use structured programming for production somewhat earlier ($p < .1$). It is interesting to note, however, that a middle management "champion" for developing structured programming capabilities is associated with later passage through each phase; it is also associated with less elapsed time between developing capabilities and using structured programming in production ($p < .05$).

Perceived Relative Advantages of Structured Programming

a. Belief that structured programming may have technical problems which should be ironed out is associated with later use of structured programming for production ($p < .001$).

b. Belief that training costs to instruct users of structured programming are steep is associated with later use of structured programming for production ($p < .01$).

c. Belief that the cost-to-benefit ratio of adopting structured programming is less favorable to the adopting company than outside developers realize is associated with later use of structured programming for production ($p < .01$).

d. Belief that maintenance costs of software developed using structured programming is unacceptably high is associated with an organization using structured programming for production later ($p < .01$).

e. Belief that production pressures are such that technical personnel cannot easily take time to learn structured programming methods is associated with later use of structured programming for production ($p < .01$).

f. When there is a belief that developing capabilities for using structured programming interferes with ongoing development processes then the organization is likely to use structured programming for production later ($p < .05$).

Time Organization Passed Through Earlier Stages

The earlier an organization develops structured programming capabilities and uses structured programming in a trial situation, the earlier it uses structured programming for production jobs ($p < .01$).

Smoothness of Passing Through Earlier Stages

Smoothness of developing structured programming capabilities and using structured programming in a trial situation did not affect time of production.

4.4. Time of Adoption and Transition Mechanisms Used

Finally, we note that extensive use of various transition mechanisms are associated with early or late adoption of structured programming at different stages in the adoption process. We report these findings without comment.

4.4.1. Developing Structured Programming Capabilities

1. Extensive use of written documentation during pre-acquisition is associated with earlier development of structured programming capabilities ($p < .05$).

2. Extensive use of seminars and conferences while structured programming capabilities are developed is associated with earlier development of structured programming capabilities ($p < .05$).

3. More extensive use of training developed by outside personnel while structured programming capabilities are developed is associated with later development of structured programming capabilities ($p < .05$).

4.4.2. Using Structured Programming in a Trial Situation

Extensive use of training developed by outside personnel while structured programming capabilities are developed is associated with later use of structured programming for trial ($p < .05$).

4.4.3. Using Structured Programming in Production

1. More extensive use of seminars and conferences during pre-acquisition ($p < .05$) and while structured programming capabilities are developed ($p < .01$) is associated with earlier use of structured programming for production.
2. More extensive use of written documentation during pre-acquisition is associated with earlier use of structured programming for production ($p < .05$).
3. More extensive use of training prepared by in-house personnel during production is associated with earlier use of structured programming for production.
4. More extensive use of training prepared by outside personnel during production is associated with later use of structured programming for production ($p < .05$).
5. More extensive use of seminars and conferences and visits to other organizations during production is associated with earlier use of structured programming for production ($p < .05$).

4.5. Summary

Table 4-4 summarizes the preceding analysis with respect to the overall effect of the organizational level of the primary advocate on the adoption of structured programming. The table shows the impact (positive, negative, or no effect) of top, middle, and technical management as well as broad-based support on the movement, timing, and ease of adoption of structured programming during each stage of adoption.

Table 4-4 can be read as follows: The primary advocacy of top management was significantly associated with ease of using structured programming for a trial project. Primary advocacy of middle management had no significant influence on ease of moving through adoption stages.

	Movement	Timing	Ease
Top	0	0	+ trial
Middle	+ develop capabilities - production	- develop capabilities - trial - production	0
Technical	+ develop capabilities	0	- trial - production
Broad-based	+ develop capabilities + production	+ develop capabilities + trial + production	+ develop capabilities + trial + production

Table 4-4: Level of Advocate's Effect on Adoption of Structured Programming

Table 4-5 summarizes the preceding analysis with respect to the overall effect of the extensive use of the various transition mechanisms on the adoption of structured programming. The table shows the significant association (positive, negative, no relationship) of training (in-house and outside) seminars, written documentation, site visits, and tools on the movement, timing, and ease of adoption of structured programming during each stage of adoption.

	Movement	Timing	Ease
Training prepared in-house	0	+ production	0
Training prepared outside	+ trial	- develop capabilities - trial	- trial - production
Seminars & Conferences	+ trial	+ develop capabilities + production	+ trial
Written Documents	0	+ develop capabilities + production	0
Site Visits	+ trial	+ production	+ develop capabilities + trial
Tools	+ develop capabilities + production	0	0

Table 4-5: Relationship Between Transition Mechanisms and Structured Programming Adoption Criteria

Table 4-5 can be read as follows: extensive use of site visits is associated with earlier use of structured programming for production projects. The reader should note that examining the data in greater depth suggests that extensive use of tools seems most effective when used during pre-acquisition, while developing capabilities and during trial stages.

5. Program Design Languages

Sixty participants responded to questions about their organization's (business unit) use of program design languages (PDLs). This innovation differs from other innovations studied in that: 1) it is more mature than some other software engineering innovations, 2) it is a tool rather than a methodology, and 3) the primary user of the technology is the individual software engineer.

Table 5-1 shows the percentage of our sample population of organizational units that have passed through each stage of the adoption process for program design languages. For this technology, the stages were:

1. Pre-acquisition, in other words going through an approval process for using program design languages within the organization.
2. Developing program design language capability, that is those tasks which enable the organization to use program design languages, such as training and/or hiring personnel.
3. Using program design languages for a pilot or test project in order to asses the usefulness of the technology before finally committing the organization to it.
4. Using program design languages in a production environment for any complete software-development projects, rather than on a trial basis.

Stage	Percentage
Pre-acquisition	100%
Develop capabilities	85%
Trial	43%
Production	78%

Table 5-1: Percentage of Organizations That Have Passed Through Each Stage of the Adoption Process for Program Design Languages.

The table should be read as follows: of the total sample of participants, 43% of the organizations passed through trial.

For program design languages, organizations often do not try out the technology in a limited-use situation before using it in a full-production environment. The reader should note that 100% of the organizations will always have passed through the pre-acquisition stage (they will have considered adopting the innovation), since this was used as a pre-screening criterion.

Table 5-2 shows both average time and the range of time at which organizational units passed through each stage of the adoption process for program design languages. Participants were asked for the year in which program design languages were first adopted (used or capabilities developed) in the organization, by stage.

Stage	Average Time	Range
Develop capabilities	early-mid 1981	1960-1986
Trial	early 1982	1970-1987
Production	mid 1982	1970-1988

Table 5-2: Time of Adoption by Stage

Note that, overall, there is not much spread between the time the organization first decided to develop capabilities for using program design languages and the time program design languages were first used in a production environment (on average, approximately one year). This may occur because using program design languages initially in a production environment involves relatively low amounts of risk versus many other innovations. Adoption involves low levels of capital expenditure, and, since it is a mature technology, late adopters may already be well informed as to the benefits of using program design languages.

5.1. Primary Advocate for Developing Program Design Language Capabilities

One of the objectives of this study was to determine the extent to which different levels in the organization served as the primary advocate for developing capabilities for using program design languages. Later we analyze the effects of support from different organization levels on the adoption process. Figure 5-1 shows the percentage of organizational respondents who indicated that the primary advocate for using program design languages was 1) top management, 2) middle management, 3) technical staff, or 4) broad-based support of technical management or staff.

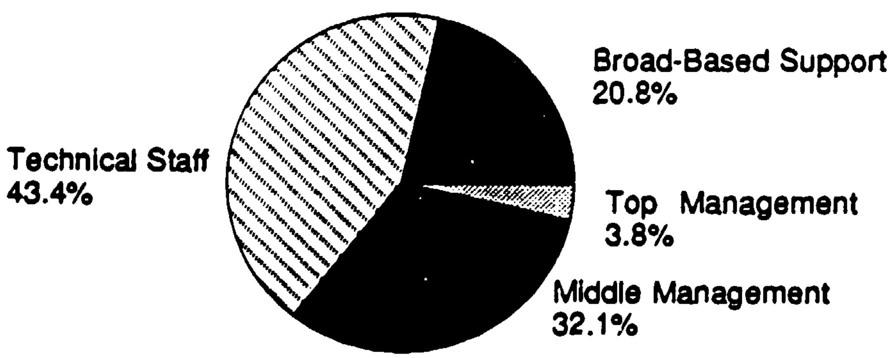

Figure 5-1: Primary Advocate for Developing Program Design Language Capabilities

The reader should note that, unlike some of our other technologies, the primary advocate for developing program design language capabilities is almost never top management. Technical staff are the dominant influencers for this technology.

5.2. Organizations' Use of Different Transition Mechanisms

Another aspect of this research was the investigation of organizations' use of various transition mechanisms to aid in the transference of program design languages within the organization. The transition mechanisms are:

1. Training prepared by in-house personnel.
2. Training prepared by outside personnel.
3. Sending personnel to seminars or conferences to learn about program design languages.
4. Providing written documentation about program design languages or articles about program design languages from technical or scholarly journals.
5. Visits to other organizations where program design languages are used.
6. Tools to aid transition to program design languages.

Table 5-3 shows, overall, the percentage of organizations' resources allocated to the different transition mechanisms during the program design languages adoption process, as well as the range found across organizations.

Transition Mechanism	Mean %	Range
Training prepared by in-house personnel	35.0	0-100
Training prepared by outside personnel	8.1	0-45
Seminars & conferences	10.4	0-100
Written documentation	29.2	0-100
Visits to other organizations	2.9	0-60
Tools to aid transition	14.4	0-75

Table 5-3: Percentage of Organizations' Resources Allocated to Different Transition Mechanisms During Program Design Language Adoption Process

Table 5-3 shows the percentage of organizational resources used by the different transition mechanisms. Because of the difference in cost and effort to the organization involved in using these mechanisms it does not, therefore, tell us the extent to which each mechanism was used during the adoption process.

We asked participants to tell us to what extent their own organization provided each of these several types of transition mechanisms to users of program design languages at each stage in the adoption process. Participants answered by responding with any number from 1 to 7, where 1 means "the mechanism was not at all provided", and 7 means "the mechanism was provided to a very great extent."

Figure 5-2 shows the mean responses for each transition mechanism, at each stage.

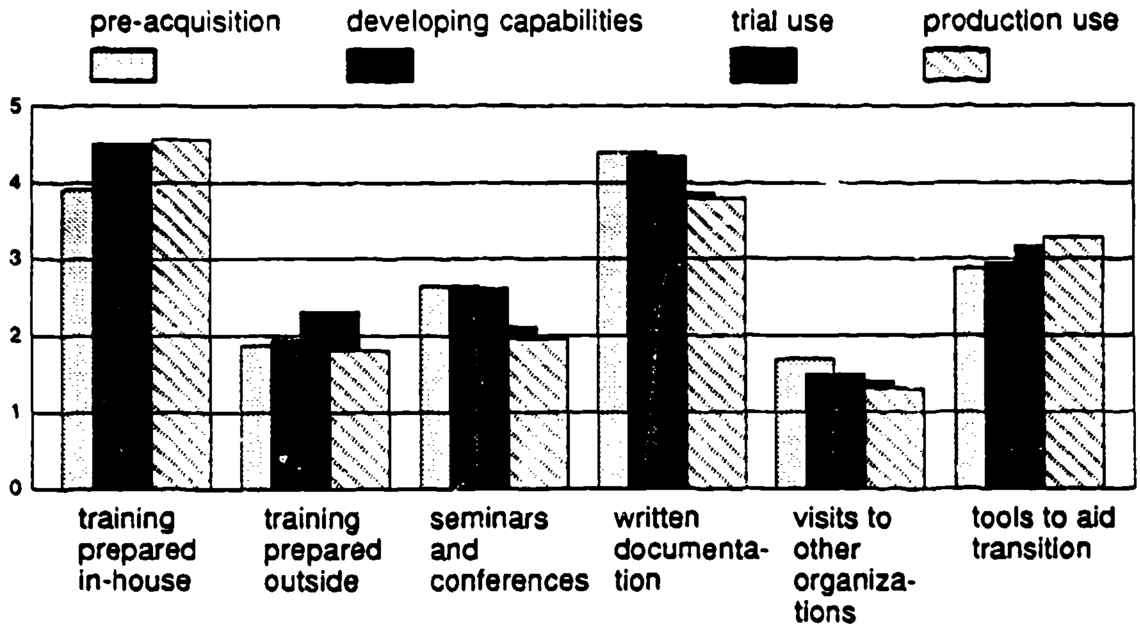

Figure 5-2: Extent to Which Various Transition Mechanisms Are Used at Different Stages in the Adoption Process for Program Design Languages

Within each stage, comparing the different transition mechanisms, note that there are significant differences in the extent to which different transition mechanisms are used within each stage. During the pre-acquisition stage written documentation is the most used transition mechanism. While developing capabilities for using program design languages, both written documentation and training prepared by in-house personnel are used equally. During trial and production stages, training prepared by in-house personnel is the most used transition mechanism. Visits to other organizations are used to the least extent of any transition mechanisms, at each stage.

Comparing each transition mechanism across stages, we can see that:

- Training prepared by in-house personnel is used most during production and least during the pre-acquisition stage.
- Training prepared by outside personnel is used most during trial and least during production stage.
- Seminars and conferences are used most during pre-acquisition and while developing capabilities and least during production ($p < .01$).
- Written documentation is used most during pre-acquisition and when capabilities are being developed, and least during trial and production ($p < .05$).

- Visits to other organizations are used most during pre-acquisition and least during production ($p < .1$).
- Tools to aid transition are used most during production and least during pre-acquisition.

5.3. The Adoption Process and The Determinants of Adoption of Program Design Languages

Analyses were done to enable us to better understand the determinants of three types of adoption phenomena related to the adoption of program design languages. These are:

1. Whether or not the organization has passed through an adoption stage (the adopt or not adopt decision for each stage in the adoption process).
2. The smoothness of the adoption process, at each stage.
3. The time at which the organization initially enters each stage in the adoption process (early versus later adoption behavior).

5.3.1. Pass Through Adoption Stage

The organization's decision to adopt or not adopt the innovation has been empirically studied previously, although not for technologies such as program design languages. In this study we extend the empirical analysis by separating out the adoption decision by stage in the process. As described previously, the stages used in the program design languages portion of the study are 1) pre-acquisition, 2) developing capabilities, 3) trial use of program design languages, and 4) use of program design languages in a production environment. Whether or not an organization has passed through one of these adoption phases is conceptualized as being a function of five classes of variables:

1. The organizational-level support of the primary advocate for the development of program design language capability.
2. The organization's beliefs about the relative advantages of using program design languages.
3. Whether the organization has passed through an earlier stage in the adoption process. For example, trial is not a necessary pre-condition for using program design languages in full production.
4. The time at which the organization passed through an earlier stage. For a mature technology like program design languages, we would predict that those organizations that passed through the prior stages at an earlier time would be more likely to have passed through later stages.

As shown in Table 5-1, most of the organizations in our sample population have already developed capabilities for using program design languages. Therefore, we limited this analysis to 1) adopting program design languages in a limited-used, trial, situation and 2) adopting program design languages for a regular production project.

5.3.1.1. Adopting Program Design Languages in a Trial Situation

Primary Advocate

When the primary advocate for developing program design language capability was technical staff, the organization was more likely to go through trial ($p < .05$). When the primary advocate was a member of top management, the organization was somewhat less likely to pass through trial ($p < .1$).

Perceived Relative Advantages of Program Design Languages

a. In those organizations where software engineers in the organization told others about the desirability of having program design language capability, the organization was more likely to pass through trial ($p < .05$).

b. When competitors were developing program design language capability, organization were more likely to pass through trial ($p < .05$).

c. The belief that training costs for the introduction of program design languages are steep is associated with passing through trial ($p < .05$). Note that trial reduces risk in this situation.

d. When there is a belief that production pressures are such that technical personnel cannot easily take time to learn program design languages then the organization is more likely to pass through trial ($p < .05$).

e. Belief that technical staff feel that they are being used as "guinea pigs in a management or government experiment" is associated with greater likelihood of trial ($p < .05$).

f. Belief that developing program design language capability interferes with on-going development processes is associated with the organization being more likely to use program design languages for trial ($p < .01$).

Pass Through Earlier Stage

This was not applicable since most of the organizations had developed program design language capability.

Timing of Passing Through Earlier Stage

Timing was not statistically significant in this analysis.

5.3.1.2. Adopting Program Design Languages In A Production Situation

Primary Advocate

The organizational level of the primary advocate for developing program design language capability was not significant in this analysis.

Perceived Relative Advantages of Program Design Languages

a. In those organizations where software engineers working in the organization told people in the organization about the desirability of using program design languages, the organization was more likely to pass through production ($p < .05$).

b. When it was believed that use of program design languages is compatible with software engineering practice in the organization, the organization was more likely to have used program design languages for production ($p < .05$).

c. When it was believed that developing program design language capability will cause an organization to be more likely to be granted government contracts, the organization was more likely to have used program design languages for production ($p < .05$).

d. Belief that program design languages are appropriate tools for software engineering tasks is associated with increased likelihood of passing through the production stage ($p < .05$).

e. Those organizations that believe that personnel familiar with other software design methods can easily be trained to use program design languages were more likely to have passed through production($p < .05$).

f. Organizations that believe the cost-to-benefit ratio of adopting program design languages is less favorable to the adopting company than outside developers realize are less likely to pass through production ($p < .01$).

g. Belief that performance quality of program design languages is too low to justify developing a program design languages capability is associated with organizations being less likely to pass through production ($p < .001$).

h. Belief that program design languages does not yield sufficient economic benefits for the company is associated with an organization being less likely to have used program design languages for production ($p < .001$).

i. Belief that return on investment for program design languages is too long term is associated with an organization being less likely to have used program design languages for production ($p < .001$).

j. When there is a belief that technical staff feel that they are being used as guinea pigs in a management or government experiment, then the organization is less likely to have used program design languages for production ($p < .01$).

Pass Through Earlier Stage

Having passed through trial was not a contributing factor.

Timing of Passing Through Earlier Stage

Timing of passing through earlier stages was not a contributing factor.

5.3.2. Smoothness of the Adoption Process

The second dependent adoption measure analyzed for program design languages is the "smoothness" of the adoption process at each of the previously described stages. The smoothness of the adoption process at a specific stage is conceptualized as a function of:

1. The extent to which the organization has used various transition mechanisms, both at this stage, and at earlier stages in the adoption process.
2. The organizational-level support of the primary advocate for the development of program design language capability.
3. The smoothness of the process of passing through earlier stages; a smoother adoption process at earlier stages.
4. The time at which the organization is passing through this stage in the adoption process.
5. The time at which the organization passed through earlier stages in the adoption process.

An analysis of smoothness of adopting program design languages at different phases was done for three phases: 1) developing program design language capability, 2) trial, and 3) production.

5.3.2.1. Smoothness of Developing Program Design Language Capability

Use of Different Transition Mechanisms

Use of transition mechanisms was not associated with smoothness of developing program design language capability at a sufficient level of significance.

Primary Advocate

When the primary advocate for developing program design language capability is made up of broad-based support, then developing program design language capability tends to be smoother($p < .01$). When the primary advocate for developing program design language capability is a member of the technical staff, developing program design language capability tends to be less smooth ($p < .01$).

Smoothness at Earlier Stage

Not applicable.

Time Organization Passed Through This Stage

Timing of developing program design language capability did not affect smoothness.

Time Organization Passed Through Earlier Stages

Not applicable.

5.3.2.2. Smoothness of Using Program Design Languages in Trial

Use of Different Transition Mechanisms

During the pre-acquisition stage, more extensive use of visits to organizations where program design languages are used is associated with a smoother adoption process ($p < .05$). While program design language capability are being developed, more extensive use of visits to other organizations is associated with a less smooth adoption process ($p < .05$). The transition mechanism used during trial that affects the smoothness of using program design languages in trial is written documentation ($p < .05$). More extensive use leads to less smooth trial adoption.

Primary Advocate

No statistically significant relationship was found.

Smoothness at Earlier Stage

Smoother development of program design language capability is associated with smoother use of program design languages in trial situations ($p < .05$).

Time Organization Passed Through This Stage

Timing of using program design languages in trial situations did not affect smoothness.

Time Organization Passed Through Earlier Stages

Timing of developing program design language capability did not affect smoothness.

5.3.2.3. Smoothness of Using Program Design Languages for Production

Use of Different Transition Mechanisms

More extensive use of training prepared by in-house personnel during all stages in the adoption process is associated with smoother adoption of program design languages in a production environment. More extensive use of training prepared by outside personnel during the production stage ($p < .05$) and extensive use of written documentation during the production stage ($p < .05$) are associated with smoother use of program design languages in production.

Primary Advocate

Broad-based support for the development of program design language capability is associated with increased smoothness of using program design languages in production ($p < .05$).

Smoothness at Earlier Stage

Smoother development of program design language capability ($p < .01$) and trial use ($p < .05$) is associated with smoother use of program design languages in a production environment.

Time Organization Passed Through This Stage

Timing of using program design languages in earlier stages did not affect smoothness.

Time Organization Passed Through Earlier Stages

Timing of passing through earlier phases did not affect smoothness.

5.3.3. Timing of Initial Entry into Stage

The third type of dependent measure analyzed in this study is the organization's timing of adoption of program design languages at each phase in the adoption process. Time of adoption is conceptualized as being a function of the following sets of variables:

1. The organizational level support of the primary advocate for the development of program design language capability.
2. The organization's beliefs about the relative advantages of using program design languages.
3. The time at which the organization passed through earlier stages in the adoption process.
4. The smoothness of the process of passing through earlier stages—a smoother adoption process at earlier stages should lead to earlier entry into later stages.

5.3.3.1. Time Program Design Language Capabilities Were Developed

Primary Advocate

The level in the organization of the primary advocate for developing program design language capability did not affect timing of developing capabilities.

Perceived Relative Advantages of Program Design Languages

a. In those organizations where belief that program design languages will be mandated for future government contracts was relevant to the decision making process, program design language capabilities were developed later ($p < .05$).

b. In those organizations where software engineers working in the organization told people in the organization about the desirability of using program design languages, the organization was more likely to develop program design language capability earlier($p < .01$).

c. In those situation where colleagues in other organizations told people about the advantages of using program design languages, organizations tended to develop program design language capability earlier than at other organizations ($p < .05$).

d. When it was believed that use of program design languages is compatible with software engineering practice in the organization, the organization was more likely to have developed program design language capability earlier ($p < .05$).

e. Belief that program design languages may have technical problems which should be ironed out is associated with later development of program design language capability ($p < .05$).

f. Belief that training costs to instruct users of program design languages are steep is associated with later development of program design language capability ($p < .01$).

g. Belief that the cost-to-benefit ratio of adopting program design languages is less favorable to the adopting company than outside developers realize is associated with later development of program design language capability ($p < .01$).

h. Belief that performance quality of program design languages is too low to justify developing a program design languages capability is associated with later development of program design language capability ($p < .01$).

i. Belief that use of program design languages does not yield sufficient economic benefits for the company is associated with later development of program design language capability ($p < .01$).

j. Belief that return on investment for program design languages is too long term is associated with later development of program design language capability ($p < .05$).

k. Belief that technical staff are skeptical about the technical value of using program design languages is associated later development of program design language capability ($p < .05$).

l. Belief that production pressures are such that technical personnel cannot easily take time to learn program design languages methods is associated with later development of program design language capability ($p < .05$).

Time Organization Passed Through Earlier Stages

Not applicable.

Smoothness of Passing Through Earlier Stages

Not applicable.

5.3.3.2. Time Program Design Languages were Used for Trial

Primary Advocate

The position of the primary advocate for developing program design language capability did not affect time of trial for program design languages.

Perceived Relative Advantages of Program Design Languages

a. When software engineers working in the organization told people thereabout the desirability of using program design languages, then the organization tended to use program design languages for a trial project earlier ($p < .01$).

b. When colleagues in other organizations inform the organization about advantages of using program design languages, the organization tended to use program design languages for a trial project earlier ($p < .01$).

c. Belief that training costs to instruct users of program design languages are steep is associated with later use of program design languages for trial ($p < .05$).

d. Belief that the cost-to-benefit ratio of adopting program design languages is less favorable to the adopting company than outside developers realize is associated with later use of program design languages for trial ($p < .05$).

e. Belief that technical staff are skeptical about the technical value of program design languages is associated with later use of program design languages for trial ($p < .05$).

f. Belief that production pressures are such that technical personnel cannot easily take time to learn program design language methods is associated with later use of program design languages for trial ($p < .01$).

g. When there is a belief that developing capabilities for using program design languages interferes with on-going development processes then the organization is likely to use program design languages for trial later ($p < .05$).

Time Organization Passed Through Earlier Stages

The earlier an organization develops program design language capability, the earlier it uses program design languages in a trial situation ($p < .01$).

Smoothness of Passing Through Earlier Stages

Smoothness of developing program design language capability was associated with earlier trial ($p < .1$).

5.3.3.3. Time Program Design Languages were Used in Production

Primary Advocate

Organizational level of the primary advocate for developing program design language capability did not affect the timing of using program design languages in production.

Perceived Relative Advantages of Program Design Languages

a. When software engineers working in the organization told people thereabout the desirability of using program design languages, then the organization tended to use program design languages for a production project earlier ($p < .01$).

b. Belief that use of program design languages is compatible with software engineering practice in the organization is associated with earlier use of program design languages for production ($p < .05$).

c. Belief that program design languages are appropriate tools for software engineering tasks is associated with earlier use of program design languages for production ($p < .05$).

d. Belief that personnel familiar with other software design methods can easily be trained to use program design languages is associated with earlier use of program design languages for production ($p < .05$).

e. Belief that program design languages may have technical problems which should be ironed out is associated with later use of program design languages for production ($p < .05$).

f. Belief that training costs for the introduction of program design languages is steep is associated with later use of program design languages for production ($p < .05$).

g. Belief that the cost-to-benefit ratio of adopting program design languages is less favorable to the adopting company than outside developers realize is associated with later use of program design languages for production ($p < .001$).

h. Belief that performance quality of program design languages is too low to justify developing a program design languages capability is associated with later use of program design languages for production ($p < .01$).

i. Belief that use of program design languages does not yield sufficient economic benefits is associated with later use of program design languages for production ($p < .001$).

j. Belief that return on investment for program design languages is too long term is associated with later use of program design languages for production ($p < .05$)..

k. Belief that technical staff feel that they are being used as "guineapigs in a management or government experiment" is associated with later use of program design languages for production ($p < .05$).

l. Belief that production pressures are such that technical personnel cannot easily take time to learn program design languages methods is associated with later use of program design languages for production ($p < .01$).

m. When there is a belief that developing capabilities for using program design languages interferes with on-going development processes then the organization is likely to use program design languages for production later ($p < .05$).

Time Organization Passed Through Earlier Stages

The earlier an organization develops program design language capability and uses program design languages in a trial situation, the earlier it uses program design languages for production jobs ($p < .01$).

Smoothness of Passing Through Earlier Stages

Smoothness of developing program design language capability and using program design languages in a trial situation did not affect time of using program design languages in production.

5.4. Time of Adoption and Transition Mechanisms Used

Finally, we note that extensive use of various transition mechanisms are associated with early or late adoption of program design languages at different stages in the adoption process. We report these findings without comment.

5.4.1. Developing Program Design Language Capability

Extensive use of in-house training while developing program design language capabilities was associated with earlier development of program design language capabilities ($p < .05$).

5.4.2. Using Program Design Languages in a Trial Situation

Specific transition mechanisms were not significantly associated with timing of using program design languages for trial.

5.4.3. Using Program Design Languages in Production

Specific transition mechanisms were not significantly associated with timing of using program design languages in production.

5.5. Summary

Table 5-4 summarizes the preceding analyses with respect to the overall effect of the organizational level of the primary advocate on the adoption of program design languages. The table shows the impact (positive, negative, or no effect) of top, middle, and technical management as well as broad-based support on the movement, timing, and ease of adoption of program design languages during each stage of adoption.

	Movement	Timing	Ease
Top	- trial	0	0
Middle	+ develop capabilities	0	0
Technical	+ develop capabilities + trial	0	- develop capabilities
Broad-based	+ develop capabilities	0	+ develop capabilities + production

Table 5-4: Level Of Advocate's Effect on Adoption of Program Design Languages

Table 5-4 can be read as follows: the primary advocacy of top management was significantly associated with not using program design languages for a pilot project; primary advocacy of top management had no significant influence on ease of moving through adoption stages.

Table 5-5 summarizes the preceding analyses with respect to the overall effect of the extensive use of various transition mechanisms on the adoption of program design languages. The table shows the impact (positive, negative, or no effect) of training (in-house and outside) seminars, written documentation, site visits, and tools on the movement, timing, and ease of adoption of program design languages during each stage of adoption.

	Movement	Timing	Ease
Training prepared in-house	+ develop capabilities + production	+ develop capabilities	+ production
Training prepared outside	+ trial - production	0	+ production
Seminars and conferences	0	0	0
Written documents	0	0	+ production - trial
Site visits	- production	0	- trial
Tools	0	0	0

Table 5-5: Relationship Between Transition Mechanisms and PDL Adoption Criteria

Table 5-5 can be read as follows: extensive use of training prepared by outside personnel was significantly associated with an organization being more likely to use PDLs for trial and less likely to use PDLs for production. The reader should note that examining the data in greater depth suggests that in-house training seems most effective when used during pre-acquisition, while developing capabilities, and during trial.

6. Software Cost Models

Sixty-one participants responded to questions about their organization's (business unit) use of software cost models. This innovation differs from other innovations studied in that: 1) it is less mature than other software engineering innovations, 2) it is a software package rather than a methodology, and 3) the primary user of the technology is the software project administrator. Table 6-1 shows the percentage of our sample population of organizational units that have passed through each stage of the adoption process for software cost models. For this technology, the stages were:

1. Pre-acquisition, in other words going through an approval process for using software cost models within the organization.
2. Development of capabilities to use software cost models, that is perform those tasks which enable the organization to use software cost models, such as training.
3. Using software cost models for a pilot or test project in order to assess the usefulness of the technology before finally committing the organization to it.
4. Using software cost models in a production environment, that is for production-development projects, rather than on a trial basis.

Stage	Percentage
Pre-acquisition	100%
Develop capabilities	89%
Trial	59%
Production	67%

Table 6-1: Percentage of Organizations That Have Passed Through Each Stage of the Adoption Process for Software Cost Models

The table should be read as follows: of the total sample of participants, 59% of the organizations passed through trial.

Clearly, for software cost models, organizations do not always try out the technology in a limited-use (trial) situation before using it in a full-production environment. The reader should note that 100% of the organizations will always have passed through the pre-acquisition stage (they will have considered adopting the innovation), since this was used as a pre-screening criterion. Table 6-2 shows both average time and the range of time at which organizational units passed through each stage of the adoption process for software cost models. Participants were asked for the year in which software cost models were first adopted (used or capabilities developed) in the organization, by stage.

Note that, overall, there is not much spread found between the time organizations, as a group, first decided to develop capabilities for using software cost models and the time they were first used in a production environment (less than one and a half years, on average). This may occur

Stage	Average Time	Range
Develop capabilities	mid 1982	1965-1987
Trial	mid 1982	1972-1987
Production	early 1983	1965-1987

Table 6-2: Time of Adoption by Stage

because using software cost models initially in a production environment involves relatively low amounts of risk versus many other innovations. Similarly, adoption of the innovation does not involve heavy financial investment.

6.1. Primary Advocate for Developing Software Cost Modeling Capabilities

One of the objectives of this study was to determine the extent to which different levels in the organization served as the primary advocate for developing capabilities for using software cost models. Later we analyze the effects of support from different organization levels on the adoption process. Figure 6-1 shows the percentage of organizational respondents who indicated that the primary advocate for using software cost models was 1) top management, 2) middle management, 3) technical staff, or 4) broad-based support of technical management or staff.

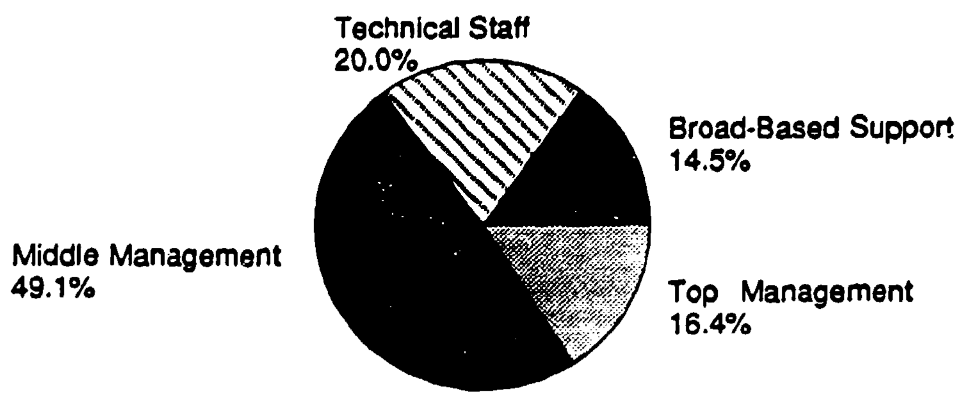

Figure 6-1: Primary Advocate for Developing Software Cost Modeling Capabilities

The reader should note that in the largest percentage of cases, the primary advocate for developing software cost modeling capabilities was a middle management person, not someone from top management. The reader may want to compare these results to those of the other technologies.

6.2. Organizations' Use of Different Transition Mechanisms

Another aspect of this research was the investigation of organizations' use of various transition mechanisms to aid in the transference of software cost modeling capability within the organization. The transition mechanisms are:

1. Training in software cost modeling prepared by in-house personnel.
2. Training in software cost modeling prepared by outside personnel.
3. During providing written documentation about software cost models or articles about software cost modeling from technical or scholarly journals.
4. Visits to other organizations where software cost models are used. Table 6-3 shows, overall, the percentage of organizations' resources allocated to the different transition mechanisms during the software cost model adoption process, as well as the range found across organizations.

Transition Mechanism	Mean %	Range
Training prepared by in-house personnel	46.9	0-100
Training prepared by outside personnel	16.4	0-75
Written documentation	32.2	0-90
Visits to other organizations	4.5	0-25

Table 6-3: Percentage of Organizations' Resources Allocated to Different Transition Mechanisms During the Adoption Process for Software Cost Modeling

Table 6-3 presents the percentage of organizational resources allocated to the different transition mechanisms for software cost models. Because of the difference in resource allocation and effort to the organization involved in using these mechanisms it does not, therefore, tell us the extent to which each mechanism was used during the adoption process.

We therefore asked participants to tell us to what extent their own organization provided each of these several types of transition mechanisms to users of software cost models at each stage in the adoption process. Participants answered by responding with any number from 1 to 7, where 1 means "the mechanism was not at all provided," and 7 means "the mechanism was provided to a very great extent."

Figure 6-2 shows the mean responses for each transition mechanism, at each stage.

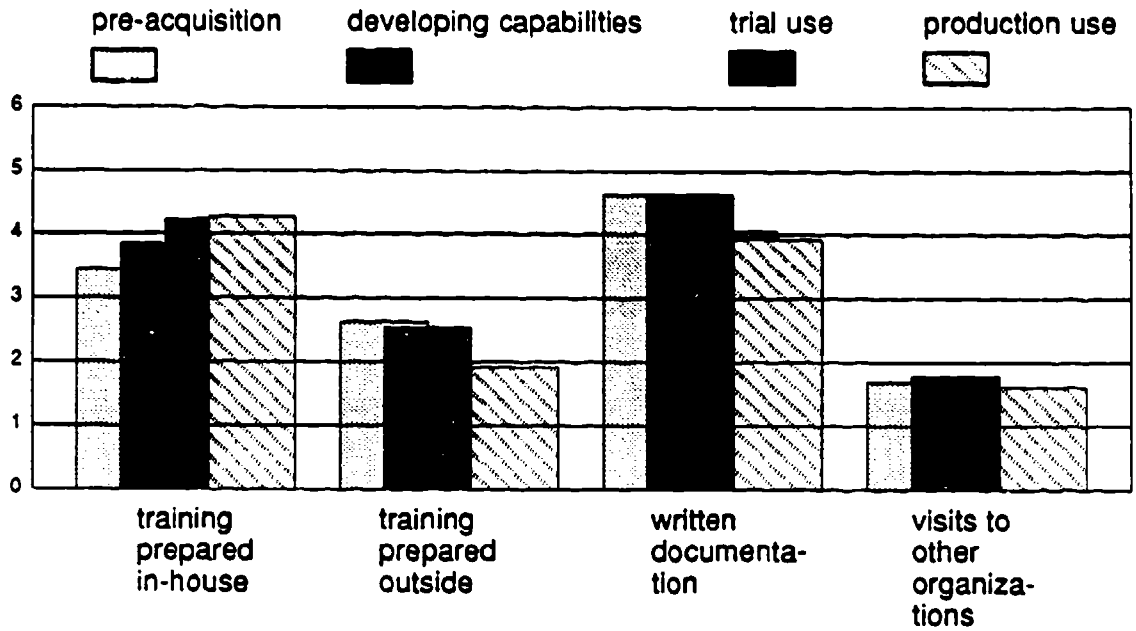

Figure 6-2: Extent to Which Various Transition Mechanisms Are Used at Different Stages in the Adoption Process for Software Cost Modeling

Within each stage, comparing the different transition mechanisms, note that there are significant differences in the extent to which different transition mechanisms are used within each stage. During pre-acquisition and developing capability stages, written documentation is the most used transition mechanism. During trial and production stages, training prepared by in-house personnel is the most used transition mechanism. Visits to other organizations are used to the least extent of any transition mechanism at each stage. Training prepared by outside personnel was not a mechanism that was relied upon to any extent in any of the stages.

Comparing each transition mechanism across stages, we can see that:

- Training prepared by in-house personnel is used to the greatest extent during production and least during pre-acquisition, however this difference is not statistically significant.

- Training prepared by outside personnel is used to the greatest extent at pre-acquisition and least during trial ($p < .01$).

- Written documentation is used most during pre-acquisition and capability development and least during production ($p < .05$).

- Visits to other organizations are used most during capability development and least during trial, however this difference is not statistically significant.

6.3. The Adoption Process and the Determinants of Adoption of Software Cost Modeling

Analyses were done to better understand the determinants of three types of adoption phenomena related to software cost modeling. These are:

1. Whether the organization has passed through an adoption stage (the adopt or not adopt decision for each stage in the adoption process).
2. The smoothness of the adoption process, at each stage.
3. The time at which the organization initially enters each stage in the adoption process (early versus later adoption behavior).

6.3.1. Pass Through Adoption Stage

This study empirically analyzes the adopt or not decision for the innovation by stage in the process. As described previously, the stages used in the software cost modeling portion of the study are 1) pre-acquisition, 2) developing capabilities, 3) trial use of software cost modeling , and 4) use of software cost modeling in a production environment. Whether the organization has passed through one of these adoption phases is conceptualized as being a function of five classes of variables:

1. The organizational-level support of the primary advocate for the development of software cost modeling capabilities.
2. The organization's beliefs about the relative advantages of using software cost modeling.
3. Whether or not the organization has passed through an earlier stage in the adoption process. For example, trial is not a necessary pre-condition for using software cost modeling in full-production. However, we would hypothesize that going through trial would increase the probability of going through production.
4. The time at which the organization passed through an earlier stage. For a technology like software cost modeling, we would predict that those organizations that passed through the prior stages at an earlier time would be more likely to have passed through later stages.

As shown in Table 6-1, most of the organizations in our sample population have already developed capabilities for using software cost models. Therefore, we limited the analysis to 1) adopting software cost models a limited-use, trial, situation and 2) adopting software cost models for a regular production project.

6.3.1.1. Adopting Software Cost Modeling in a Trial Situation

Primary Advocate

Organizations in which advocacy for developing software cost model capabilities was made up of broad-based support were somewhat more likely to pass through trial ($p < .1$).

Perceived Relative Advantages of Software Cost Modeling

Those organizations that develop software cost modeling software are more likely to pass through trial ($p < .05$).

Pass Through Earlier Stage

This was not applicable since most of the organizations had developed software cost modeling capabilities.

Timing of Passing Through Earlier Stage

The timing of adoption was significant in this analysis. Specifically, the earlier the organization had developed software cost model capabilities, the more likely they were to go through trial ($p < .05$).

6.3.1.2. Adopting Software Cost Modeling in a Production Situation

Primary Advocate

The organizational level of the primary advocate for developing software cost models in the production situation is significant. When the primary advocate from the organization is drawn from broad-based support, software cost modeling is more likely to pass through the production stage ($p < .05$).

Perceived Relative Advantages of Software Cost Modeling

a. Organizations that believe the earlier an organization develops capabilities for using software cost models, the more likely they will receive government contracts were more likely to have used cost models in a production situation ($p < .01$).

b. Those organizations that believe that personnel familiar with other software cost estimation techniques can easily be trained to use software cost models were more likely to have passed through production ($p < .05$).

c. Organizations that believe the costs to train people to use software cost models are steep are less likely to use the technology in a production environment ($p < .05$).

d. Organizations that believe that developing capabilities for using software cost models interferes with on-going development processes are less likely to use software cost models in a production environment ($p < .05$).

Pass Through Earlier Stage

No statistically significant relationship was found.

Timing of Passing Through Earlier Stage

The earlier the organization passed through trial, the more likely it was to pass through production ($p < .05$).

6.3.2. Smoothness of the Adoption Process

The second dependent adoption measure analyzed for software cost modeling is the "smoothness" of the adoption process at each of the previously described stages. The smoothness of the adoption process at a specific stage is conceptualized as a function of:

1. The extent to which the organization has used various transition mechanisms, both at this stage, and at earlier stages in the adoption process.
2. The organizational-level support of the primary advocate for the development of software cost modeling capabilities.
3. The smoothness of the process of passing through earlier stages—a smoother adoption process at earlier stages (e.g., when software cost modeling capabilities are being developed) should lead to a smoother adoption at later stages.
4. The time at which the organization is passing through this stage in the adoption process.
5. The time at which the organization passed through earlier stages in the adoption process.

An analysis of smoothness of adopting software cost models at different phases was done for three phases: 1) developing software cost modeling capabilities, 2) trial, and 3) production.

6.3.2.1. Smoothness of Developing Software Cost Model Capabilities

Use of Different Transition Mechanisms

Transition mechanisms were not associated at a statistically significant level with smoothness.

Primary Advocate

The organizational level of the primary advocate for developing software cost modeling capability was not statistically significant in this analysis.

Smoothness at Earlier Stage

This item was not applicable.

Time Organization Passed Through This Stage

Timing of developing software cost model capabilities did not affect smoothness.

Time Organization Passed Through Earlier Stages

Not applicable.

6.3.2.2. Smoothness of Using Software Cost Models in Trial

Use of Different Transition Mechanisms

Transition mechanisms were not associated at a statistically significant level with smoothness.

Primary Advocate

When the primary advocate for developing software cost model capabilities is top management then using software cost models in a trial situation tends to be smoother ($p < .1$).

Smoothness at Earlier Stage

Smoother development of software cost model capabilities is associated with smoother use of software cost models in trial situations ($p < .01$).

Time Organization Passed Through This Stage

Timing of using software cost models in trial situations did not affect smoothness.

Time Organization Passed Through Earlier Stages

Timing of developing software cost models capabilities did not affect smoothness.

6.3.2.3. Smoothness of Using Software Cost Models For Production

Use of Different Transition Mechanisms

Transition mechanisms were not associated at a statistically significant level with smoothness.

Primary Advocate

When the primary advocate for developing software cost modeling capabilities was made up of broad-based support, the process of using software cost models in production tended to be smoother ($p < .1$).

Smoothness at Earlier Stage

Smoother development of software cost models capabilities ($p < .001$) is associated with smoother use of software cost models in a production environment.

Time Organization Passed Through This Stage

Timing of using software cost models in production situations did not affect smoothness.

Time Organization Passed Through Earlier Stages

Timing of passing through earlier phases did not affect smoothness.

6.3.3. Timing of Initial Entry Into Stage

The third type of dependent measure analyzed in this study is the organization's timing of adoption of software cost models at each phase in the adoption process. Time of adoption is conceptualized as being a function of the following sets of variables: 1) the organizational level support of the primary advocate for the development of software cost model capabilities; 2) the organization's beliefs about the relative advantages of using software cost models; 3) the time at which the organization passed through earlier stages in the adoption process; and 4) the smoothness of the process of passing through earlier stages; a smoother adoption process at earlier stages should lead to earlier entry into later stages.

6.3.3.1. Time Software Cost Modeling Capabilities were Developed

Primary Advocate

When support for developing capabilities was broad-based, the organization tended to develop software cost model capabilities earlier ($p < .05$).

Perceived Relative Advantages of Structured Programming

a. When it was believed that developing capabilities for using software cost models would make the organization more competitive in getting government contracts, the organization tended to develop software cost modeling capabilities earlier ($p < .05$).

b. When software engineers working in the organization told people there about the desirability of using software cost models, then the organization tended to develop software cost model capabilities earlier ($p < .05$).

c. In those situation where colleagues in other organizations told people about the advantages of using software cost models, organizations tended to develop software cost model capabilities earlier than at other organizations ($p < .05$).

d. When competitors were developing software cost model capabilities, the organization tended to develop capabilities earlier ($p < .05$).

e. Belief that organizations that currently have capabilities for using software cost models are more innovative than those that do not is associated with earlier development of software cost modeling capability ($p < .01$).

f. Belief that technical staff have no motivation to adopt software cost models since benefits would be realized only at corporate level is associated with later development of software cost models capabilities ($p < .05$).

g. Belief that technical staff are skeptical about the technical value of software cost models is associated with later development of capabilities ($p < .05$).

Time Organization Passed Through Earlier Stages

Not applicable.

Smoothness of Passing Through Earlier Stages

Not applicable.

6.3.3.2. Time Software Cost Models were Used for Trial

Primary Advocate

A top-management primary advocate is associated with earlier trial ($p < .1$).

Perceived Relative Advantages of Software Cost Models

Not significant at the $p < .05$ level of significance.

Time Organization Passed Through Earlier Stages

The earlier an organization develops software cost models capabilities, the earlier it uses software cost models in a trial situation ($p < .01$).

Smoothness of Passing Through Earlier Stages

Smoother development of software cost model capabilities was associated with earlier trial ($p < .1$).

6.3.3.3. Time Software Cost Models were Used in Production

Primary Advocate

The organizational level of the primary advocate for developing software cost modeling capabilities did not affect timing of using software cost models in production.

Perceived Relative Advantages of Structured Programming

Not significant at the $p < .05$ level.

Time Organization Passed Through Earlier Stages

The earlier an organization develops software cost models capabilities and uses software cost models in a trial situation, the earlier it uses software cost models for production jobs ($p < .01$).

Smoothness of Passing Through Earlier Stages

Smoother development of software cost models capabilities is associated with earlier use of software cost models for production ($p < .1$).

6.4. Time of Adoption and Transition Mechanisms Used

Finally, we note that extensive use of various transition mechanisms are associated with early or late adoption of software cost models at different stages in the adoption process. We report these findings without comment.

6.4.1. Developing Software Cost Modeling Capabilities

1. Extensive use of training prepared by in-house personnel during pre-acquisition is associated with later development of software cost modeling capabilities ($p < .05$).
2. Extensive use of training prepared by outside personnel while Software cost models capabilities are developed is associated with earlier development of software cost modeling capabilities ($p < .05$).

6.4.2. Using Software Cost Models in a Trial Situation

1. Extensive use of training prepared by outside personnel during pre-acquisition is associated with earlier use of software cost models for trial ($p < .1$).

2. More extensive use of training prepared by outside personnel while software cost modeling capabilities are developed is associated with earlier use of software cost models for trial ($p < .05$).

3. More extensive use of visits to other organizations while software cost models are used for trial is associated with earlier trial ($p < .05$).

6.4.3. Using Software Cost Models in Production

Not significant at $p < .05$ level.

6.5. Summary

Table 6-4 summarizes the preceding analyses with respect to the overall effect of the organizational level of the primary advocate on the adoption of software cost modeling. The table shows the impact (positive, negative, or no effect) of top, middle, and technical management as well as broad-based support on the movement, timing, and ease of adoption of software cost models during each stage of adoption.

	Movement	Timing	Ease
Top	0	+ trial	+ trial
Middle	+ develop capabilities	0	0
Technical	+ develop capabilities	0	0
Broad-based	+ trial + production	+ develop capabilities	+ production

Table 6-4: Level of Advocate's Effect on Adoption of Software Cost Modeling

Table 6-4 can be read as follows: the primary advocacy of top management was significantly associated with using software cost models earlier for trial projects; primary advocacy of middle management had no significant influence on ease of moving through adoption stages.

Table 6-5 summarizes the preceding analyses with respect to the overall effect of the extensive use of transition mechanisms on the adoption of software cost modeling. The table shows the significant association (positive, negative, or no effect) of training (in-house and outside), written documentation, and site visits on the movement, timing, and ease of adoption of software cost models during each stage of adoption.

	Movement	Timing	Ease
Training prepared in-house	+ develop capabilities + trial	- develop capabilities	0
Training prepared outside	0	+ develop capabilities + trial	0
Written Documents	+ trial	0	0
Site Visits	+ production	+ trial	0

Table 6-5: Relationship Between Transition Mechanisms and Software Cost Models Adoption Criteria

Table 6-5 can be read as follows: extensive use of written documentation is associated with organizations being more likely to use software cost models for trial projects. The reader should note that examining the data in greater depth suggests that site visits seem more effective when used during pre-acquisition, while developing capabilities, and during trial.

7. Complexity Metrics

Forty-one participants responded to questions about their organization's (business unit) use of complexity metrics. This innovation differs from other innovations studied in this research in that: 1) it is less mature than other software engineering innovations, 2) it is a methodology rather than a tool, and 3) the primary user of the technology is the project administrator. Table 7-1 shows the percentage of our sample population of organizational units that have passed through each stage of the adoption process for complexity metrics. For this technology, the stages were:

1. Pre-acquisition, in other words going through an approval process for using complexity metrics within the organization.

2. Developing complexity metrics capabilities, that is those tasks which enable the organization to use complexity metrics, such as training and/or hiring personnel.

3. Using complexity metrics for a pilot or test project in order to asses the usefulness of the technology before finally committing the organization to it.

4. Using complexity metrics in a production environment, that is for any complete software-development projects, rather than on a trial basis.

Stage	Percentage
Pre-acquisition	100%
Develop capabilities	59%
Trial	20%
Production	37%

Table 7-1: Percentage of Organizations That Have Passed Through Each Stage of the Adoption Process for Complexity Metrics

The table should be read as follows: of the total sample of participants, 20% of the organizations passed through trial.

Many organizations have not yet adopted complexity metrics. The reader should note that 100% of the organizations will always have passed through the pre-acquisition stage (they will have considered adopting the innovation), since this was used as a pre-screening criterion. Table 7-2 shows both average time and the range of time at which organizational units passed through each stage of the adoption process for complexity metrics. Participants were asked for the year in which complexity metrics was first adopted (used or capabilities developed) in the organization, by stage.

Overall, there is, on average, a longer spread of time found between the time organizations first decided to develop capabilities for using complexity metrics and the time it was first used in a production environment. The sample size, however, is small.

Stage	Average Time	Range
Develop capabilities	early 1984	1974-1987
Trial	early-mid 1985	1983-1987
Production	mid-late 1985	1983-1987

Table 7-2: Time of Adoption by Stage

7.1. Primary Advocate for Developing Complexity Metrics Capabilities

One of the objectives of this study was to determine the extent to which different levels in the organization served as the primary advocates for developing capabilities for using complexity metrics. Later we analyze the effects of support from different organization levels on the adoption process. Figure 7-1 shows the percentage of organizational respondents who indicated that the primary advocate for using complexity metrics was 1) top management, 2) middle management, 3) technical staff, or 4) broad-based support of technical management or staff.

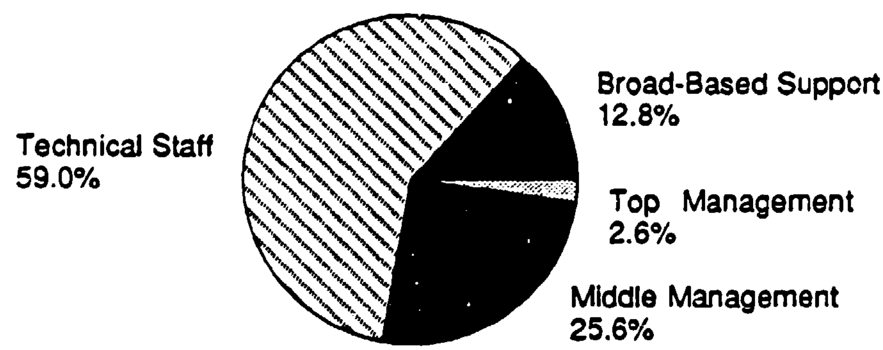

Figure 7-1: Primary Advocate for Developing Complexity Metrics Capabilities

The reader should note that in the largest percentage of cases, the primary advocate for developing complexity metrics capabilities was a technical staff person.

7.2. Organizations' Use of Different Transition Mechanisms

Another aspect of this research was the investigation of organizations' use of various transition mechanisms to aid in the transference of complexity metrics within the organization. The transition mechanisms are:

1. Complexity metrics training prepared by in-house personnel.

2. Complexity metrics training prepared by outside personnel.

3. Providing written documentation about complexity metrics or articles about complexity metrics from technical or scholarly journals.

4. Visits to other organizations where complexity metrics are used.

Table 7-3 shows, overall, the percentage of organizations' resources allocated to the different transition mechanisms during the complexity metrics adoption process, as well as the range found across organizations.

Transition Mechanism	Mean %	Range
Training prepared by in-house personnel	41.9	0-90
Training prepared by outside personnel	20.5	0-90
Written documentation	31.9	0-100
Visits to other organizations	5.6	0-40

Table 7-3: Percentage of Organizations' Resources Allocated to Different Transition Mechanisms During the Adoption Process for Complexity Metrics

Table 7-3 shows the percentage of organizational resources used by the different transition mechanisms. Because of the difference in cost and effort to the organization involved in using these mechanisms, it does not tell us the extent to which each mechanism was used during the adoption process.

We asked participants to tell us to what extent their own organization provided these transition mechanisms to users of complexity metrics, at each stage in the adoption process. Participants answered by responding with any number from 1 to 7, where 1 means "the mechanism was not at all provided", and 7 means "the mechanism was provided to a very great extent."

Figure 7-2 shows the mean responses for each transition mechanism, at each stage. Within each stage, comparing the different transition mechanisms, note that there are significant differences in the extent to which different transition mechanisms are used within each stage. During all stages, written documentation is the most used transition mechanism. Visits to other organizations are used to the least extent of any transition mechanisms during all stages.

Comparing each transition mechanism across stages, we can see that:

- Training prepared by in-house personnel is used most while capabilities are being developed and least during the pre-acquisition and trial stages.
- Training prepared by outside personnel is used most during trial and least while capabilities are developed and during the production stage.
- Written documentation is used most while capabilities are developed and least during production.

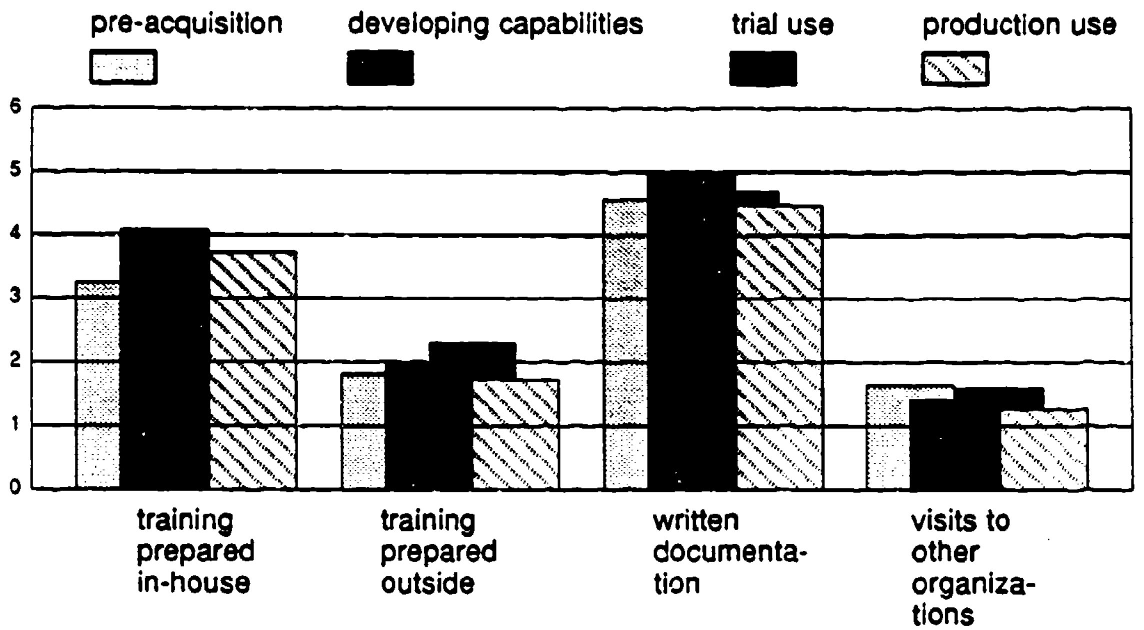

Figure 7-2: Extent to Which Various Transition Mechanisms are used at Different Stages in the Adoption Process for Complexity Metrics

- Visits to other organizations are used most during pre-acquisition and least during production.

7.3. The Adoption Process and the Determinants of Adoption of Complexity Metrics

Analyses were done to enable us to better understand the determinants of three types of adoption phenomena related to the adoption of complexity metrics. These are:

1. Whether or not the organization has passed through an adoption stage (the adopt or not adopt decision for each stage in the adoption process).

2. The smoothness of the adoption process, at each stage.

3. The time at which the organization initially enters each stage in the adoption process (early versus later adoption behavior).

7.3.1. Pass Through Adoption Stage

An empirical analysis was done on the complexity metrics data which breaks down the adopt or not-adopt decision for the innovation by stage in the process. As described previously, the stages used in the complexity metrics portion of the study are 1) pre-acquisition, 2) developing capabilities, 3) trial use of complexity metrics, and 4) use of complexity metrics in a production environment. Whether or not an organization has passed through one of these adoption phases is conceptualized as being a function of four classes of variables:

1. The organizational level of the primary advocate for the development of complexity metrics capabilities.
2. The organization's beliefs about the relative advantages of using complexity metrics.
3. Whether the organization has passed through an earlier stage in the adoption process. For example, trial is not a necessary pre-condition for using complexity metrics in full production. However, we would hypothesize that going through trial would increase the probability of going through production.
4. The time at which the organization passed through an earlier stage.

We limited this analysis to 1) adopting complexity metrics in a limited-use, trial situation and 2) adopting complexity metrics for a regular production project.

7.3.1.1. Adopting Complexity Metrics in a Trial Situation

Primary Advocate

When the primary advocate for developing complexity metrics capabilities was middle management, the organization was more likely to go through trial ($p < .01$). When the primary advocate for developing complexity metrics capabilities was technical staff, the organization was less likely to pass through trial ($p < .05$).

Perceived Relative Advantages of Complexity Metrics

a. Those organizations that believe that developing complexity metrics capabilities will make their organization more competitive in getting consulting projects with government contractors were more likely to pass through trial ($p < .01$).
b. When upper management believe that having capabilities for using complexity metrics would benefit the organization, the organization is more likely to pass through trial ($p < .01$).
c. Belief that technical staff are skeptical about the technical value of using complexity metrics is associated with an organization being less likely to have used complexity metrics for production ($p < .01$).

Pass Through Earlier Stage

This was not used in this analysis.

Timing of Passing Through Earlier Stage

Organizations that developed complexity metrics capabilities earlier were more likely to use complexity metrics for trial ($p < .05$).

7.3.1.2. Adopting Complexity metrics in a Production Situation

Primary Advocate

Organizations in which the primary advocate for developing complexity metrics capabilities was from middle management were more likely to use complexity metrics for production.

Perceived Relative Advantages of Complexity Metrics

a. When colleagues in other organizations told others of the advantages of using complexity metrics, the organization was less likely to pass through production.

b. Belief that costs to train people to use complexity metrics are steep is associated with an organization being less likely to have used complexity metrics for production ($p < .05$).

Pass Through Earlier Stage

Organizations that passed through trial were more likely to pass through production ($p < .05$).

Timing of Passing Through Earlier Stage

No significant relationship.

7.3.2. Smoothness of the Adoption Process

The second dependent adoption measure analyzed for complexity metrics is the "smoothness" of the adoption process at each of the previously described stages. The smoothness of the adoption process at a specific stage is conceptualized as a function of:

1. The extent to which the organization has used various transition mechanisms, both at this stage, and at earlier stages in the adoption process.
2. The organizational-level support of the primary advocate for the development of complexity metrics capabilities.
3. The smoothness of the process of passing through earlier stages—a smoother adoption process at earlier stages (e.g., when complexity metrics capabilities are being developed) should lead to a smoother adoption at later stages.
4. The time at which the organization is passing through this stage in the adoption process.
5. The time at which the organization passed through earlier stages in the adoption process.

An analysis of smoothness of adopting complexity metrics at different phases was done for three phases: 1) developing complexity metrics capabilities, 2) trial, and 3) production.

7.3.2.1. Smoothness of Developing Complexity Metrics Capabilities

Use of Different Transition Mechanisms

During both the pre-acquisition stage and while complexity metric capabilities are developed, more extensive use of visits to organizations where complexity metrics are used is associated with a less smooth adoption process ($p < .01$ during pre-acquisition and $p < .05$ while capabilities are developed). Also, during pre-acquisition, more extensive use of training prepared by outside personnel is associated with a less smooth process ($p < .05$).

Primary Advocate

When the primary advocate for developing complexity metrics capabilities is made up of broad-based support, then developing complexity metrics capabilities tends to be smoother ($p < .01$). When the primary advocate is a member of the technical staff, then developing complexity metrics capabilities tends to be less smooth ($p < .01$).

Smoothness at Earlier Stage

Not applicable.

Time Organization Passed Through This Stage

Timing of developing complexity metrics capabilities did not affect smoothness.

Time Organization Passed Through Earlier Stages

Not applicable.

7.3.2.2. Smoothness of Using Complexity Metrics in Trial

Use of Different Transition Mechanisms

Use of visits during pre-acquisition is associated with a less smooth trial.

Primary Advocate

When the primary advocate for developing complexity metrics capabilities is from technical staff, then using complexity metrics in a trial situation tends to be less smooth ($p < .05$).

Smoothness at Earlier Stage

Smoother development of complexity metrics capabilities is associated with smoother use of complexity metrics in trial situations ($p < .01$).

Time Organization Passed Through This Stage

Timing of using complexity metrics in trial situations did not affect smoothness.

Time Organization Passed Through Earlier Stages

Timing of developing complexity metrics capabilities did not affect smoothness.

7.3.2.3. Smoothness of Using Complexity Metrics for Production

Use of Different Transition Mechanisms

Extensive use of written documentation during production is associated with smoother production ($p < .05$).

Primary Advocate

When the primary advocate for developing complexity metrics is a member of the technical staff, then using complexity metrics for production tends to be less smooth ($p < .05$).

Smoothness at Earlier Stage

Smoother development of complexity metrics capabilities ($p < .01$) and smoother use of complexity metrics in trial ($p < .01$) is associated with smoother use of complexity metrics in a production environment.

Time Organization Passed Through This Stage

Timing of using complexity metrics in production situations did not affect smoothness.

Time Organization Passed Through Earlier Stages

Timing of passing through earlier phases did not affect smoothness.

7.3.2.4. Timing of Initial Entry into Stage

The third type of dependent measure analyzed in this study is the organization's timing of adoption of complexity metrics at each phase in the adoption process. Time of adoption is conceptualized as being a function of the following sets of variables:

1. The organizational level support of the primary advocate for the development of complexity metrics capabilities.
2. The organization's beliefs about the relative advantages of using complexity metrics.
3. The time at which the organization passed through earlier stages in the adoption process.
4. The smoothness of the process of passing through earlier stages—a smoother adoption process at earlier stages should lead to earlier entry into later stages.

7.3.2.5. Time Complexity Metrics Capabilities were Developed

Primary Advocate

Organizations in which the primary advocate for developing complexity metrics capabilities is middle management tend to develop complexity metrics capabilities earlier than other organizations ($p < .01$). When the primary advocate is a member of the technical staff, the organization tends to develop complexity metrics later ($p < .1$).

Perceived Relative Advantages of Complexity Metrics

a. When upper management believed that having capabilities for using complexity metrics would benefit the organization the organization tended to develop complexity metrics capabilities earlier ($p < .01$).

b. When there is a belief that use of complexity metrics will be mandated for future government contracts, then the organization is more likely to develop complexity metrics capabilities later ($p < .05$).

Time Organization Passed Through Earlier Stages

Not applicable.

Smoothness of Passing Through Earlier Stages

Not applicable.

7.3.2.6. Time Complexity Metrics was Used for Trial

Primary Advocate

The position of the primary advocate for developing complexity metrics capabilities did not affect time of trial for complexity metrics.

Perceived Relative Advantages of Complexity Metrics

No relationships found at a high enough level of significance.

Time Organization Passed Through Earlier Stages

The timing of developing complexity metrics capabilities had no effect on timing of trial.

Smoothness of Passing Through Earlier Stages

Smoothness of developing complexity metrics capabilities did not affect time of trial.

7.3.2.7. Time Complexity Metrics was Used in Production

Primary Advocate

The position of the primary advocate for developing complexity metrics capabilities did not affect time of using complexity metrics for production projects.

Perceived Relative Advantages of Complexity Metrics

No relationships found.

Time Organization Passed Through Earlier Stages

The earlier an organization uses complexity metrics in a trial situation, the earlier it uses complexity metrics for production jobs ($p < .01$).

Smoothness of Passing Through Earlier Stages

Not statistically significant.

7.4. Time of Adoption and Transition Mechanisms Used

Finally, we note that extensive use of various transition mechanisms are associated with early or late adoption of complexity metrics at different stages in the adoption process. We report these findings without comment.

7.4.1. Using Complexity Metrics in Production

1. More extensive use of written documentation during the pre- acquisition stage is associated with later use of complexity metrics for production ($p < .05$).
2. More extensive use of training prepared by in-house personnel while complexity metrics capabilities are developed is associated with earlier use of complexity metrics for production ($p < .01$).
3. More extensive use of training prepared by in-house personnel during production is associated with earlier use of complexity metrics for production ($p < .05$).

7.5. Summary

Table 7-4 summarizes the preceding analysis with respect to the overall effect of the organizational level of the primary advocate on the adoption of complexity metrics. The table shows the impact (positive, negative, or no effect) of top, middle, and technical management as well as broad-based support on the movement, timing, and ease of adoption of complexity metrics during each stage of adoption.

	Movement	Timing	Ease
Top	0	0	0
Middle	+ develop capabilities + trial + production	+ develop capabilities	0
Technical	+ develop capabilities - trial	- develop capabilities	- develop capabilities - trial - production
Broad-based	+ develop capabilities	0	+ develop capabilities + production

Table 7-4: Level of Advocate's Effect on Adoption of Complexity Metrics

Table 7-4 can be read as follows: the primary advocacy of middle management was significantly associated with developing capabilities and for using complexity metrics for both trial and production projects. Primary advocacy of top management had no significant influence on ease of moving through adoption stages.

Table 7-5 summarizes the preceding analysis with respect to the overall effect of the extensive use of transition mechanisms on the adoption of complexity metrics. The table shows the significant association (positive, negative, or no effect) of training (in-house and outside), written documentation, and site visits on the movement, timing, and ease of adoption of complexity metrics during each stage of adoption.

	Movement	Timing	Ease
Training prepared in-house	+ develop capabilities	+ production	0
Training prepared outside	0	0	- develop capabilities
Written Documents	+ develop capabilities - production	- production	+ production
Site Visits	+ trial	0	- develop capabilities - trial

Table 7-5: Relationship Between Transition Mechanisms and Complexity Metrics Adoption Criteria

Table 7-5 can be read as follows: extensive use of training prepared by in-house staff is associated with organizations being more likely to develop capabilities for using complexity metrics.

8. Ada

Sixty-six participants responded to questions about their organization's (business unit) use of Ada. This innovation differs from other innovations studied in this research in that: 1) it is less mature than many other software engineering innovations, 2) it is both a methodology and a tool, and 3) the primary user of the technology is the individual software engineer or a team of software engineers.

Table 8-1 shows the percentage of our sample population of organizational units that have passed through each stage of the adoption process for Ada. For this technology, the stages were:

1. Pre-acquisition, in other words going through an approval process for using Ada within the organization.
2. Acquiring and installing an Ada compiler.
3. Developing Ada capabilities, that is those tasks which enable the organization to use Ada, such as training and/or hiring personnel.
4. Using Ada for a pilot or test project in order to asses the usefulness of the technology before finally committing the organization to it.
5. Using Ada in a production environment, that is for any complete software-development projects, rather than on a trial basis.

Stage	Percentage
Pre-acquisition	100%
Compiler acquisition	86%
Develop capabilities	96%
Trial	68%
Production	64%

Table 8-1: Percentage of Organizations That Have Passed Through Each Stage of the Adoption Process for Ada

The table should be read as follows: of the total sample of participants, 68% of the organizations passed through trial.

Note that developing Ada capabilities (training or hiring personnel) and acquiring an Ada compiler are not always done together. Some organizations may have capabilities for developing systems in Ada, but no compiler. Others may have the compiler, but no manpower capabilities. Note also that use of Ada on a trial basis is more common than for some of the other technologies described in this report. Note that 100% of the organizations will always have passed through the pre-acquisition stage (they will have considered adopting the innovation), since this was used as a pre-screening criterion.

Table 8-2 shows both average time and the range of time at which organizational units passed through each stage of the adoption process for Ada. Participants were asked for the year in which Ada was first adopted (used, capabilities developed, and compiler acquired) in the organization, by stage.

Stage	Average Time		Range
Acquire compiler	early	1985	1981-1987
Develop capabilities	early	1984	1977-1988
Trial	early-mid	1985	1981-1987
Production	early	1986	1983-1987

Table 8-2: Time of Adoption by Stage

Note that, overall, there is more spread found between the time the organization first decided to develop capabilities for using Ada and the time it was first used in a production environment than for the other technologies discussed in this report.

8.1. Primary Advocate for Developing Ada Capabilities

One of the objectives of this study was to determine the extent to which different levels in the organization served as the primary advocates for developing capabilities for using Ada. Later we analyze the effects of support from different organization levels on the adoption process. Figure 8-1 shows the percentage of organizational respondents who indicated that the primary advocate for Ada was 1) top management, 2) middle management, 3) technical staff, or 4) broad-based support of technical management or staff.

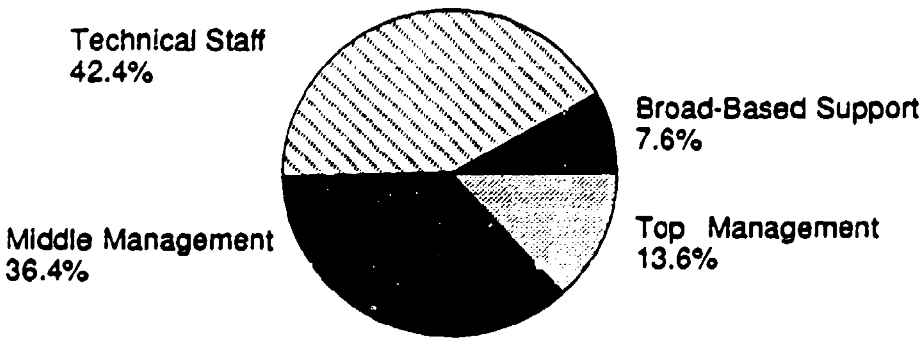

Figure 8-1: Primary Advocate for Developing Ada Capabilities

Based on this data, the Ada "champion" seems to come primarily from middle management or technical staff. However, a primary advocate in one position may not always be as effective in facilitating adoption as one in another position.

8.2. Organizations' Use of Different Transition Mechanisms

Another aspect of this research was the investigation of organizations' use of various transition mechanisms to aid in the transference of Ada within the organization. The transition mechanisms are:

1. Ada training prepared by in-house personnel.
2. Ada training prepared by outside personnel.
3. Sending personnel to seminars or conferences, for example, to SIGAda.
4. Providing written documentation about Ada or articles about Ada from technical or scholarly journals.
5. Visits to other organizations where Ada is used.

Table 8-3 shows, overall, the percentage of organizations' resources allocated to the different transition mechanisms during the Ada adoption process, as well as the range found across organizations.

Transition Mechanism	Mean %	Range
Training prepared by in-house personnel	36.5	0-100
Training prepared by outside personnel	20.4	0-80
Seminars & conferences	16.4	0-60
Written documentation	20.5	0-70
Visits to other organizations	5.7	0-50

Table 8-3: Percentage of Organizations' Resources Allocated to Different Transition Mechanisms During the Adoption Process for Ada

Table 8-3 shows, for Ada, the percentage of organizational resources used by the different transition mechanisms. Because of the difference in cost and effort to the organization involved in using these mechanisms it does not tell us the extent to which each mechanism was used during the adoption process.

We asked participants to tell us to what extent their own organization provided each of these transition mechanisms to users of Ada at each stage in the adoption process. Participants answered by responding with any number from 1 to 7, where 1 means "the mechanism was not at all provided," and 7 means "the mechanism was provided to a very great extent."

Figure 8-2 shows the mean responses for each transition mechanism, at each stage.

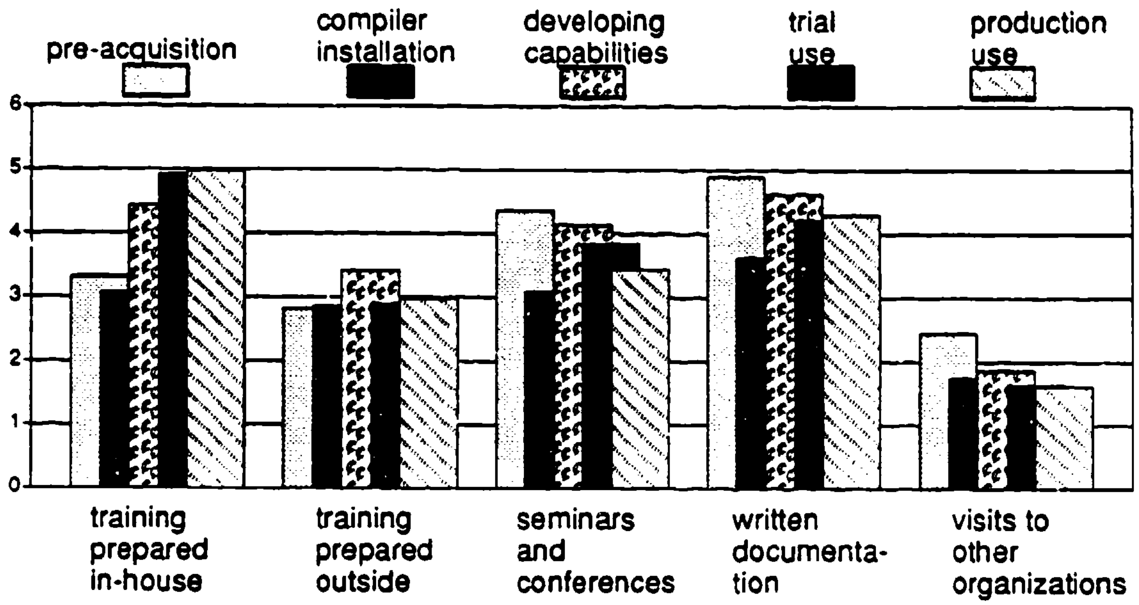

Figure 8-2: Extent to Which Various Transition Mechanisms Are Used at Different Stages in the Adoption Process for Ada

Within each stage, comparing the different transition mechanisms, note that there are significant differences in the extent to which different transition mechanisms are used within each stage. During pre-acquisition, compiler installation, and developing capability stages, written documentation is the most usod transition mechanism. During trial and production stages, training prepared by in-house personnel is the most used transition mechanism. Visits to other organizations are used to the least extent of any transition mechanisms, at each stage.

Comparing each transition mechanism across stages, we can see that:

- Training prepared by in-house personnel is used most during production and least during the compiler installation stage ($p < .01$).
- Training prepared by outside personnel is used most while Ada capabilities are developed and least during compiler installation and pre-acquisition stages ($p < .05$).
- Seminars and conferences are used most during pre-acquisition and least during compiler installation ($p < .01$).
- Written documentation is used most during pre-acquisition and least during compiler installation ($p < .01$).
- Visits to other organizations is used most during pre-acquisition and least during production ($p < .05$).

8.3. The Adoption Process and the Determinants of Adoption of Ada

Analyses were done to enable us to better understand the determinants of three types of adoption phenomena related to the adoption of Ada. These are:

1. Whether the organization has passed through an adoption stage (the adopt or not adopt decision for each stage in the adoption process).
2. The smoothness of the adoption process, at each stage.
3. The time at which the organization initially enters each stage in the adoption process (early versus later adoption behavior).

8.3.1. Pass Through Adoption Stage

In this study we separate out and empirically analyze the adoption decision by stage in the process. As described previously, the stages used in the Ada portion of the study are 1) pre-acquisition, 2) compiler installation, 3) developing capabilities, 4) trial use of Ada, and 5) use of Ada in a production environment. Whether or not an organization has passed through one of these adoption phases is conceptualized as being a function of four classes of variables:

1. The organizational level support of the primary advocate for the development of Ada capabilities.
2. The organization's beliefs about the relative advantages of using Ada.
3. Whether or not the organization has passed through an earlier stage in the adoption process.
4. The time at which the organization passed through an earlier stage.

We limited this analysis to 1) adopting Ada in a limited-use, trial situation and 2) adopting Ada for a regular production project.

8.3.1.1. Adopting Ada in a Trial Situation

Primary Advocate

When the primary advocate for developing Ada capabilities was technical staff or was made up of broad-based support, the organization was more likely to go through trial ($p < .1$). When the primary advocate for developing Ada capabilities was from middle management or from top management, the organization was less likely to pass through trial ($p < .1$).

Perceived Relative Advantages of Ada

a. Belief that there are sufficient Ada tools available to justify developing an Ada capability at this time is associated with organizations being more likely to use Ada in a trial situation ($p < .05$).

b. Belief that organizations that develop Ada capabilities within the next year will be perceived as leaders in software development is associated with organizations being less likely to use Ada in a trial situation ($p < .05$).

Pass Through Earlier Stage

Not used in this analysis.

Timing of passing through Earlier Stage

Timing was not significant in this analysis.

8.3.1.2. Adopting Ada in a Production Situation

Primary Advocate

When the primary advocate for developing Ada capabilities was technical staff or was made up of broad-based support, the organization was more likely to go through production ($p < .05$). When the primary advocate for developing Ada capabilities was middle management, the organization was less likely to pass through production ($p < .01$).

Perceived Relative Advantages of Ada

a. Belief that developing Ada capabilities will make the organization more likely to get government contracts is associated with greater likelihood of the organization having used Ada in production ($p < .01$).

b. Belief that there are sufficient Ada tools available to justify developing an Ada capability at this time is associated with organizations being more likely to use Ada in a production situation ($p < .05$).

c. Belief that organizations that develop Ada capabilities within the next year will be perceived as leaders in software development is associated with organizations being less likely to have used Ada for production ($p < .05$).

d. Belief that performance quality of Ada compilers is too low to justify developing an Ada capability at this time is associated with organizations being less likely to use Ada in a production situation ($p < .05$).

e. Belief that Ada does not yield sufficient economic benefits is associated with decreased likelihood of having used Ada in production ($p < .05$).

f. When there is a belief that production pressures are such that technical personnel cannot take time to learn Ada, the organization is less likely to have used Ada for production ($p < .05$).

Pass Through Earlier Stage

Organizations that passed through trial were more likely to have passed through production stage ($p < .01$).

Timing of Passing Through Earlier Stage

The earlier the organization developed Ada capabilities, acquired a compiler, and went through a trial project, the more likely it was to pass through production ($p < .01$ for all).

8.3.1.3. Smoothness of the Adoption Process

The second dependent adoption measure analyzed for Ada is the "smoothness" of the adoption process at each of the previously described stages. The smoothness of the adoption process at a specific stage is conceptualized as a function of:

1. The extent to which the organization has used various transition mechanisms, both at this stage, and at earlier stages in the adoption process.
2. The organizational-level support of the primary advocate for the development of Ada capabilities.
3. The smoothness of the process of passing through earlier stages—a smoother adoption process at earlier stages (e.g., when Ada capabilities are being developed) should lead to a smoother adoption at later stages.
4. The time at which the organization is passing through this stage in the adoption process.
5. The time at which the organization passed through earlier stages in the adoption process.

An analysis of smoothness of adopting Ada at different phases was done for three phases: 1) acquiring an Ada compiler, 2) developing Ada capabilities, 3) trial, and 4) production.

8.3.1.4. Smoothness of Acquiring an Ada Compiler

Use of Different Transition Mechanisms

During the pre-acquisition stage, more extensive use of Ada training prepared by outside personnel is associated with a less smooth adoption process ($p < .05$), and extensive use of visits to other organizations is associated with a smoother adoption process ($p < .05$).

Primary Advocate

Not statistically significant.

Smoothness at Earlier Stage

Not applicable.

Time Organization Passed Through This Stage

Timing of acquiring an Ada compiler did not affect smoothness.

Time Organization Passed Through Earlier Stages

Not applicable.

8.3.1.5. Smoothness of Developing Ada Capabilities

Use of Different Transition Mechanisms

While Ada capabilities are being developed during pre-acquisition and during compiler acquisition, more extensive use of Ada training prepared by in-house personnel is associated with a smoother adoption process ($p < .05$).

Primary Advocate

When the primary advocate for the technology was technical staff, the organization experienced increased difficulty in developing human capability ($p < .1$).

Smoothness at Earlier Stage

Not applicable.

Time Organization Passed Through This Stage

Those organizations that acquired a compiler or developed Ada capabilities earlier tended to have a smoother process of developing Ada capabilities ($p < .01$).

Time Organization Passed Through Earlier Stages

Not applicable.

8.3.1.6. Smoothness of Using Ada in Trial

Use of Different Transition Mechanisms

During both compiler installation and while Ada capabilities are being developed, more extensive use of Ada training prepared by outside personnel and seminars or conferences is associated with a smoother adoption process ($p < .05$ for both).

Primary Advocate

Broad-based support for the development of Ada capabilities is associated with a less smooth trial ($p < .05$). A primary advocate from technical staff is associated with smoother trial ($p < .1$).

Smoothness at Earlier Stage

Not statistically significant.

Time Organization Passed Through This Stage

Timing of using Ada in trial situations did not affect smoothness.

Time Organization Passed Through Earlier Stages

Not statistically significant.

8.3.1.7. Smoothness of Using Ada for Production

Use of Different Transition Mechanisms

Extensive use of written documentation during trial is associated with a less smooth use of Ada for production tasks ($p < .05$).

Primary Advocate

Organizational level of the primary advocate for developing Ada capabilities had no effect on the smoothness of using Ada in production.

Smoothness at Earlier Stage

Smoother development of Ada capabilities ($p < .01$) and smoother use of Ada in trial ($p < .01$) are associated with smoother use of Ada in a production environment.

Time Organization Passed Through This Stage

Timing of using Ada in production situations did not affect smoothness.

Time Organization Passed Through Earlier Stages

Timing of passing through earlier phases did not affect smoothness.

8.3.2. Timing of Initial Entry into Stage

The third type of dependent measure analyzed in this study is the organization's timing of adoption of Ada at each phase in the adoption process. Time of adoption is conceptualized as being a function of the following sets of variables:

1. The organizational level support of the primary advocate for the development of Ada capabilities.

2. The organization's beliefs about the relative advantages of using Ada.

3. The time at which the organization passed through earlier stages in the adoption process.

4. the smoothness of the process of passing through earlier stages—a smoother adoption process at earlier stages should lead to earlier entry into later stages.

8.3.2.1. Timing of Acquiring an Ada Compiler

Primary Advocate

Organizations in which there is a broad-based support tend to acquire an Ada compiler earlier ($p < .05$).

Perceived Relative Advantages of Ada

a. Those organizations that believe that using Ada is compatible with software engineering practices in their organization were more likely to have acquired an Ada compiler earlier ($p < .05$).

b. When there is a belief that organizations that currently have Ada capabilities are more innovative than those that do not, then the organization is more likely to have acquired an Ada compiler earlier ($p < .05$).

c. Belief that performance quality of Ada compilers is too low to justify developing an Ada capability at this time is associated with organizations acquiring an Ada compiler later ($p < .01$).

d. Belief that Ada does not offer sufficient economic benefits is associated with organizations acquiring an Ada compiler later ($p < .05$).

Time Organization Passed Through Earlier Stages

Not applicable.

Smoothness of Passing Through Earlier Stages

Not applicable.

8.3.2.2. Timing of Developing Ada Capabilities

Primary Advocate

No significant relationship was found.

Perceived Relative Advantages of Ada

a. When it was believed that use of Ada is compatible with software engineering practice in the organization and is appropriate for software engineering tasks, the organization was more likely to have developed Ada capabilities earlier ($p < .05$).

b. Belief that use of Ada is more appropriate for military applications than for commercial applications is associated with an organization developing Ada capabilities later ($p < .05$).

c. Belief that organizations should have a "wait and see" attitude regarding the Ada mandate before developing Ada capability is associated with an organization developing Ada capabilities later ($p < .01$).

d. Belief that Ada may have technical problems which should be ironed out is associated with later development of Ada capabilities ($p < .01$).

e. Belief that training costs to instruct users of Ada are steep is associated with later development of Ada capabilities ($p < .01$).

f. Belief that performance quality of Ada compilers is too low is associated with later development of Ada capabilities ($p < .05$).

Time Organization Passed Through Earlier Stages

Not applicable.

Smoothness of Passing Through Earlier Stages

Not applicable.

8.3.2.3. Time Ada was Used for Trial

Primary Advocate

When the primary advocate for developing Ada capabilities was top management ($p < .1$) or was made up of broad-based support ($p < .01$), the organization was more likely to use Ada for a trial project earlier. When the primary advocate was middle management the organization was likely to pass through trial later ($p < .05$).

Perceived Relative Advantages of Ada

a. Belief that organizations that develop Ada capabilities within the next year will be perceived as leaders in software development is associated with organizations trying Ada later ($p < .05$).

b. Belief that performance quality of Ada compilers is too low to justify developing an Ada capability is associated with organizations trying Ada later ($p < .05$).

c. Belief that Ada does not yield sufficient economic benefits is associated with organizations trying Ada later ($p < .01$).

d. Belief that technical staff are skeptical about the technical value of Ada is associated with organizations trying Ada later ($p < .05$).

Time Organization Passed Through Earlier Stages

The earlier an organization develops Ada capabilities and acquires an Ada compiler, the earlier it tends to use Ada in a trial situation ($p < .01$).

Smoothness of Passing Through Earlier Stages

A smoother process of developing Ada capabilities was associated with earlier trial ($p < .05$).

8.3.2.4. Time Ada was Used in Production

Primary Advocate

Organizations in which the primary advocate for developing Ada capabilities is a member of technical staff tend to use Ada for a production project later than other organizations ($p < .01$). When the primary advocate is a member of middle management, organizations tend to use Ada in production earlier.

Perceived Relative Advantages of Ada

a. In those situation where there is a belief that developing Ada capabilities early will make the organization more likely to get government contracts, organizations tended to use Ada for production jobs earlier than at other organizations ($p < .05$).

b. Belief that organizations should have a "wait and see" attitude regarding the Ada mandate is associated with later use of Ada for production ($p < .05$).

c. When there is a belief that organizations that currently have Ada capabilities are more innovative than those that do not, then the organization is more likely to have used Ada in production earlier ($p < .05$).

d. Belief that the cost-to-benefit ratio of adopting Ada is less favorable to the adopting company than outside developers realize is associated with later use of Ada for production ($p < .05$).

e. Belief that performance quality of Ada compilers is too low to justify developing an Ada capability at this time is associated with organizations using Ada for production later ($p < .05$).

f. Belief that Ada does not yield sufficient economic benefits is associated with later use of Ada in production ($p < .05$).

g. Belief that technical staff are heavily committed to old programming languages which they feel work very well for them is associated with an organization being likely to have used Ada for production later ($p < .01$).

h. Belief that technical staff have no motivation to adopt Ada since benefits would be realized only at corporate level is associated with an organization using Ada for production later ($p < .05$).

Time Organization Passed Through Earlier Stages

The earlier an organization acquires an Ada compiler ($p < .01$), the earlier it uses Ada for production jobs.

Smoothness of Passing Through Earlier Stages

Smoother development of Ada capabilities is associated with earlier use of Ada for production ($p < .05$).

8.4. Time of Adoption and Transition Mechanisms Used

Finally, we note that extensive use of various transition mechanisms are associated with early or late adoption of Ada at different stages in the adoption process. We report these findings without comment.

8.4.1. Acquiring an Ada Compiler

1. Extensive use of training prepared by in-house personnel during pre-acquisition is associated with earlier acquisition of an Ada compiler ($p < .01$).

2. Extensive use of training prepared by in-house personnel ($p < .01$), seminars and conferences ($p < .01$), and written documentation ($p < .05$) during compiler acquisition is associated with earlier compiler acquisition.

3. Extensive use of training prepared by in-house personnel ($p < .01$) and seminars and conferences ($p < .05$) while Ada capabilities are developed is associated with earlier compiler acquisition. Extensive use of training prepared by outside personnel during this stage is associated with later compiler acquisition ($p < .05$).

8.4.2. Developing Ada Capabilities

1. Extensive use of training prepared by outside personnel while Ada capabilities are developed is associated with later development of Ada capabilities ($p < .05$);

2. More extensive use of written documentation ($p < .01$) and training prepared by in-house personnel during compiler acquisition is associated with earlier development of Ada capabilities ($p < .05$).

8.4.3. Using Ada in a Trial Situation

1. Extensive use of training prepared by in-house personnel while acquiring an Ada compiler is associated with earlier use of Ada for trial ($p < .01$).

2. More extensive use of written documentation during pre- acquisition and while acquiring an Ada compiler is associated with earlier use of Ada for trial ($p < .05$).

3. Extensive use of training prepared by in-house personnel while Ada capabilities are being developed is associated with earlier trial ($p < .01$). 4. More extensive use of written documentation while Ada capabilities are developed is associated with earlier trial ($p < .01$).

5. More extensive use of written documentation during trial is associated with earlier trial ($p < .01$).

8.4.4. Using Ada in Production

1. More extensive use of training prepared by in-house personnel during pre-acquisition is associated with earlier use of Ada for production ($p < .01$).

2. More extensive use of training prepared by in-house personnel while acquiring an Ada compiler is associated with earlier use of Ada for production ($p < .01$).

3. More extensive use of training prepared by in-house personnel while Ada capabilities are developed is associated with earlier use of Ada for production ($p < .05$).

4. More extensive use of training prepared by outside personnel when Ada is used for trial is associated with later use of Ada for production ($p < .05$).

5. More extensive use of training prepared by in-house personnel during trial is associated with earlier use of Ada for production ($p < .05$).

8.5. Summary

Table 8-4 summarizes the preceding analysis with respect to the overall effect of the organizational level of the primary advocate on the adoption of Ada. The table shows the impact (positive, negative, or no effect) of top, middle, and technical management as well as broad-based support on the movement, timing, and ease of adoption of Ada during each stage of adoption.

	Movement	Timing	Ease
Top	- trial	+ trial	0
Middle	+ develop capabilities - trial - production	- trial + production	0
Technical	+ compiler acquisition + develop capabilities + trial + production	- production	- develop capabilities + trial
Broad-based	+ trial + production	+ compiler acquisition + trial	- trial

Table 8-4: Level of Advocate's Effect On Adoption of Ada

Table 8-4 can be read as follows: the primary advocacy of top management was significantly associated with using Ada earlier for pilot projects. Primary advocacy of top management had no significant influence on ease of moving through adoption stages.

Table 8-5 summarizes the preceding analysis with respect to the overall effect of the extensive use of transition mechanisms on the adoption of Ada. The table shows the significant association (positive, negative, or no effect) of training (in-house and outside), written documentation, and site visits on the movement, timing, and ease of adoption of Ada during each stage of adoption.

	Movement	Timing	Ease
Training prepared in-house	+ trial + production	+ compiler acquisition + develop capabilities + trial + production	+ develop capabilities
Training prepared outside	+ trial	- compiler acquisition - develop capabilities - production	- compiler acquisition + trial
Seminars & conferences	+ trial + production	+ compiler acquisition + develop capabilities	+ trial
Written Documents	+ trial + production	+ compiler acquisition + develop capabilities + trial	- production
Site Visits	0	0	+ compiler acquisition

Table 8-5: Relationship Between Transition Mechanisms and Ada Adoption Criteria

Table 8-5 can be read as follows: extensive use of training prepared by in-house personnel was positively associated with an organization using Ada for trial and production.

The reader should note that examining the data in greater depth suggests that training prepared by in-house personnel is most effective when used during pre-acquisition, while developing capabilities, and during compiler acquisition.

9. Summary of Findings

The following sections summarize some of the findings from this study. Data from previous chapters are displayed in matrices in Section 9.1 to show the influence of each level of primary advocacy for all five of the technologies. Also, data from previous chapters are displayed in matrices in section 9.2 to show the influence of different transition mechanisms on all five of the technologies.

9.1. Advocacy Effects

9.1.1. Top Management Advocacy Effects

Table 9-1 summarizes the effects of top management advocacy on the adoption of all five of the technologies addressed in this study.

	Movement	Timing	Ease
SP	0	0	+ trial
PDL	- trial	0	0
SCM	0	+ trial	+ trial
CM	0	0	0
Ada	- trial	+ trial	0

Table 9-1: Top Management Advocate's Effect on Adoption of Technologies

Table 9-1 can be read as follows: top management advocacy was associated with an organization being less likely to use PDLs for trial projects.

9.1.2. Middle Management Advocacy Effects

Table 9-2 summarizes the effects of middle management advocacy on the adoption of all five of the technologies addressed in this study.

	Movement	Timing	Ease
SP	+ develop capabilities - production	- develop capabilities - trial - production	0
PDL	+ develop capabilities	0	0
SCM	+ develop capabilities	0	0
CM	+ develop capabilities + trial + production	+ develop capabilities	0
Ada	+ develop capabilities - trial - production	- trial + production	0

Tal .e 9-2: Middle Management Advocate's Effect on Adoption of Technologies

Table 9-2 can be read as follows: middle management advocacy was positively associated with an organization developing structured programming capabilities and negatively associated with an organization using structured programming for production tasks.

9.1.3. Technical Staff Advocacy Effects

Table 9-3 summarizes the effects of technical staff advocacy on the adoption of all five of the technologies addressed in this study.

	Movement	Timing	Ease
SP	+ develop capabilities	0	- trial - production
PDL	+ develop capabilities + trial	0	- develop capabilities
SCM	+ develop capabilities	0	0
CM	+ develop capabilities - trial	- develop capabilities	- develop capabilities - trial - production
Ada	+ compiler acquisition + develop capabilities + trial + production	- production	- develop capabilities + trial

Table 9-3: Technical Staff Advocate's Effect on Adoption of Technologies

Table 9-3 can be read as follows: primary advocacy by a member of the technical staff is associated with later use of Ada for production projects.

9.1.4. Broad-Based Advocacy Effects

Table 9-4 summarizes the effects of broad-based advocacy on the adoption of all five or the technologies addressed in this study.

	Movement	Timing	Ease
SP	+ develop capabilities + production	+ develop capabilities + trial + production	+ develop capabilities + trial + production
PDL	+ develop capabilities	0	+ develop capabilities + production
SCM	+ trial + production	+ develop capabilities	+ production
CM	+ develop capabilities	0	+ develop capabilities
Ada	+ trial + production	+ compiler acquisition + trial	- trial

Table 9-4: Broad-based Advocates' Effect on Adoption of Technologies

Table 9-4 can be read as follows: broad-based support is associated with an organization being more likely to develop capabilities for using PDLs, structured programming, and complexity metrics.

9.1.5. Overall Advocacy Effects

Some overall observations can be made about the results shown in Tables 9-1 through 9-4, as follows:

1. Overall, across technologies, adoption criteria, and phases, top-management primary advocacy has the least effect on adoption. In fact, it was associated significantly only with the trial stage of adoption.

2. Overall, broad-based support has the most widespread, positive impact on software engineering adoption.

3. The effect of middle-management primary advocacy appears to be an almost equal mix of significant positive and negative effects. However, this is across all technologies, adoption criteria, and stages of adoption. When the results are broken down by adoption criteria, middle-management advocacy has a positive effect on movement, and a somewhat negative effect on timing ($p < .1$).

4. In addition, the technologies were grouped based on whether they were targeted to software engineers (SP and PDL) or to administrators (SCM and CM). When this grouping is

done, middle-management advocacy has a positive association with administrative technologies, but a *negative* association with software-engineer targeted technologies ($p < .01$).

5. In addition, positive middle-management effects are concentrated at the stage of developing capabilities for using the technology ($p < .05$).

6. The effect of technical-staff primary advocacy also appears to be an equal mix of significant positive and negative effects. However, when the type of adoption criteria is taken into consideration, there tends to be a significant positive association with movement and a negative association with ease of adoption ($p < .01$).

7. When the technologies are grouped into methods-based technologies (SP and CM) versus tool-based technologies (PDL and SCM), there are more negative associations of technical staff advocacy with methods-based technologies ($p < .05$).

The reader is cautioned that the results of this analysis should be considered exploratory. However, the above observations may provide the practitioner with some preliminary guidelines as to primary advocacy for successful adoption for different types of software engineering technologies.

9.2. Transition Mechanism Effects

The following sections summarize additional findings from the study. The data from each of the chapters are displayed as matrices to show the influence of extensive use of different transition mechanisms for all five of the technologies. Only those transition mechanisms that were incorporated in questionnaires for all five technologies are summarized in this way.

9.2.1. Effects of Training Prepared by In-House Staff

Table 9-5 summarizes the effects of extensive use of training prepared by in-house staff on the adoption of all five of the technologies addressed in this study.

	Movement	Timing	Ease
SP	0	+ production	0
PDL	+ develop capabilities + production	+ develop capabilities	+ production
SCM	+ develop capabilities + trial	- develop capabilities	0
CM	+ develop capabilities	+ production	0
Ada	+ trial + production	+ compiler acquisition + develop capabilities + trial + production	+ develop capabilities

Table 9-5: Effects of Extensive Use of Training Prepared by In-House Staff Across Technologies

Table 9-5 can be read as follows: extensive use of training prepared by in-house staff was associated with an organization being more likely to develop complexity metrics capabilities.

9.2.2. Effects of Training Prepared by Outside Personnel

Table 9-6 summarizes the effects of extensive use of training prepared by outside personnel on the adoption of all five of the technologies addressed in this study.

	Movement	Timing	Ease
SP	+ trial	- develop capabilities - trial	- trial - production
PDL	+ trial - production	0	+ production
SCM	0	+ develop capabilities + trial	0
CM	0	0	- develop capabilities
Ada	+ trial	- develop capabilities - compiler acquisition - production	- compiler acquisition + trial

Table 9-6: Effect of Extensive Use of Training Prepared by Outside Personnel Across Technologies

Table 9-6 can be read as follows: extensive use of training prepared by outside personnel is associated with organizations developing structured programming capabilities later.

9.2.3. Effects of Written Documentation

Table 9-7 summarizes the effects of extensive use of written documentation on the adoption of all five of the technologies addressed in this study.

	Movement	Timing	Ease
SP	0	+ develop capabilities + production	0
PDL	0	0	- trial + production
SCM	+ trial	0	0
CM	+ develop capabilities - production	- production	+ production
Ada	+ trial + production	+ compiler acquisition + develop capabilities + trial	- production

Table 9-7: Effect of Extensive Use of Written Documentation Across Technologies

Table 9-7 can be read as follows: extensive use of written documentation is associated with an organization being more likely to use software cost models for trial.

9.2.4. Effects of Site Visits

Table 9-8 summarizes the effects of extensive use of site visits to other organizations where the technology is used on the adoption of all five technologies.

	Movement	Timing	Ease
SP	+ trial	+ production	+ develop capabilities + trial
PDL	- production	0	- trial
SCM	+ production	+ trial	0
CM	+ trial	0	- develop capabilities - trial
Ada	0	0	+ compiler acquisition

Table 9-8: Effect of Extensive Use of Site Visits Across Technologies

Table 9-8 can be read as follows: extensive use of site visits is associated with organizations being less likely to use PDLs for production.

9.2.5. Observations About the Effects of Transition Mechanisms

The above summary of the effects of extensive use of transition mechanisms suggests some general observations, as follows:

1. Overall, across technologies, adoption stages, and adoption criteria, extensive use of training prepared by in-house staff has the greatest positive association with software engineering adoption.
2. Across technologies, but with effects that vary somewhat with adoption criteria and adoption stage, extensive use of training prepared by outside personnel often is negatively associated with adoption.
3. In general, positive associations of extensive use of transition mechanisms vary somewhat based on adoption stage. Extensive use of in-house training is associated more with developing capabilities and pilot stages. Site visits and outside training is associated more with trial stage of adoption.
4. When technologies are grouped into tool-based (PDL and SCM) versus methods-based (SP and CM) technologies, there are some differences in transition mechanism effects. Extensive use of both in-house and outside training is more positively associated with the adoption of tools. Site visits are more positively associated with methods.
5. In general, analyzing the data in greater depth suggests that when extensive use of a transition mechanism is effective, it may be beneficial to begin extensive use as early as possible and continue extensive use through trial.

9.3. Effects of Perceived Advantages and Disadvantages

The following sections summarize the influence of organizations' beliefs about perceived advantages or disadvantages on technology adoption. Questions dealing with beliefs about the technologies were grouped together into several factors.* These factors are:

- lack of economic benefits
- training difficulties
- obtaining government contracts
- resistance of technical staff to the technology
- effects of interpersonal communications

The data for each factor are displayed in a matrix to show how each belief influences technology adoption for all five technologies addressed in this study.

* A data reduction technique known as factor analysis was used to group related questions. Essentially, factor analysis groups variables based on patterns of similarity in responses to questions. In this way, the underlying structure of beliefs is uncovered.

9.3.1. Beliefs about Lack of Economic Benefits

Table 9-9 summarizes the relationship between beliefs about lack of economic benefits and the adoption of the five technologies.

	Movement	Timing	Ease
SP	+ trial - production	0	- trial - production
PDL	- develop capabilities - production	- develop capabilities - production	- trial
SCM	- develop capabilities	0	- develop capabilities - trial - production
CM	- develop capabilities	0	0
Ada	0	- compiler acquisition - develop capabilities - production	0

Table 9-9: Relationship Between Beliefs About Lack of Economic Benefits and Adoption

Table 9-9 should be read as follows: belief that use of complexity metrics lacks economic benefits is associated with an organization being less likely to develop capabilities for using complexity metrics.

9.3.2. Beliefs About Training Difficulties

Table 9-10 summarizes the relationship between beliefs about training difficulties and the adoption of the five technologies addressed in this study.

	Movement	Timing	Ease
SP	0	- develop capabilities - trial - production	0
PDL	+ trial	- trial	- develop capabilities - production
SCM	- production	0	0
CM	0	0	- develop capabilities
Ada	0	0	0

Table 9-10: Relationship Between Beliefs About Training Difficulties and Adoption

Table 9-10 should be read as follows: Belief that training personnel to use software cost models is time-consuming and expensive is associated with an organization being less likely to use software cost models for production.

For PDLs, beliefs about training difficulties were closely related to beliefs about resistance of the technical staff to PDLs. These beliefs, therefore, have been combined into a single factor and are reported both in Table 9-10 and Table 9-12.

9.3.3. Beliefs About Obtaining Government Contracts

Table 9-11 summarizes the relationship between beliefs about obtaining government contracts and the adoption of the five technologies addressed in this study.

	Movement	Timing	Ease
SP	+ develop capabilities	0	0
PDL	+ develop capabilities	0	+ develop capabilities + production
SCM	+ develop capabilities	0	- trial
CM	0	0	+ develop capabilities + production
Ada	+ develop capabilities + production	0	0

Table 9-11: Relationship Between Beliefs About Obtaining Government Contracts and Adoption

Table 9-11 should read as follows: Belief that adopting the technology will make the organization more likely to receive government contracts is associated with organizations being more likely to develop capabilities for using structured programming.

9.3.4. Beliefs About Resistance of Technical Staff

Table 9-12 summarizes the relationship between beliefs about resistance of technical staff and the adoption of the five technologies.

	Movement	Timing	Ease
SP	- production	0	- develop capabilities - trial - production
PDL	+ trial	- trial	- develop capabilities - production
SCM	0	- develop capabilities	0
CM	0	0	- trial
Ada	0	0	0

Table 9-12: Relationship Between Beliefs About Resistance of Technical Staff and Adoption

Table 9-12 should be read as follows: belief that technical staff are resistant to structured programming is associated with an organization being less likely to use structured programming for production jobs.

9.3.5. Interpersonal Communication and Adoption

Table 9-13 summarizes the relationship between interpersonal communications and the adoption of the five technologies addressed in this study.

	Movement	Timing	Ease
SP	+ develop capabilities	0	+ production
PDL	+ trial	+ develop capabilities + trial + production	0
SCM	0	+ develop capabilities	0
CM	+ develop capabilities - trial - production	0	- production
Ada	0	0	+ compiler acquisition

Table 9-13: Relationship Between Interpersonal Communications and Adoption

Table 9-13 should be read as follows: communication among staff within the organization and persons in other organizations about software cost models is associated with an organization developing capabilities for using software cost models earlier.

9.3.6. Summary of Effects of Perceived Advantages and Disadvantages

Overall, we can summarize the information in the preceding tables as follows:

1. Overall beliefs about the economic advantages (such as obtaining government contracts) or disadvantages clearly have a significant association with software engineering technology adoption, across all of the technologies addressed in this study. The potential for obtaining government contracts in an incentive for adoption primarily at the "develop capabilities" stage.

2. Human factors (training difficulties and the resistance of technical staff) seem to have more impact on the ease of adoption than on timing or movement.

3. Human factors have their most extensive impact on technologies oriented to individual software engineers (structured programming and PDLs).

4. Beliefs in economic incentives have their most extensive impact on the tool-based technologies (PDLs and software cost models).

5. Other factors which should be taken into consideration during the technology transition process are:

   - advantages to the organization because of the prestige of adopting the innovation (leading to perceptions of leadership or innovativeness of the firm)

   - compatibility of the technology with either the mission of the firm or with the technological culture of the firm

   - interpersonal communications among individual software engineers, both within the firm and in other firms (to the extent that this is important, interpersonal communication about a new software engineering technology can sometimes be transmitted at seminars and conferences)

These factors also were found to be significantly associated with technology adoption, although not in as widespread a manner.

10. Conclusion

This report discusses a study conducted to examine the adoption process for five software engineering innovations: structured programming, program design languages, software cost models, complexity metrics, and Ada. These innovations were chosen for study because they varied in terms of the maturity of the technology, the tangibility of the innovation, and the primary user of the innovation. Organizations' adoption behavior was empirically examined as a multi-stage process. Participants representing their organizational business units responded to questions about the organization's adoption decisions, the adoption process, transition mechanisms used to facilitate adoption, and beliefs about the relative advantages and disadvantages of the innovations. Stages in the adoption process examined included: when capabilities for using the innovation were developed, when the innovation was used for a trial project, and when the innovation was used for a full production project.

The authors found that different factors often are related to adoption of the innovations at the different stages. Transition mechanisms and perceived relative advantages of the innovation which facilitate adoption at one stage do not necessarily have the same effect at other stages. The analysis also examined the effect on the adoption process of the level in the organization of the primary advocate for developing capabilities for using the innovation. By doing this type of empirical study and analysis we hope to provide substantive aid to practitioners involved in the technology transition process. An objective of this study is to enable practitioners to more fully understand the factors and processes that influence adoption, postponement, or rejection of these types of software engineering innovations, as well as to understand influences on the smoothness of the process.

11. References

Fairley, R. (1985). *Software Engineering Concepts.* NY: McGraw-Hill.

Riddle, W. (1984). *The Magic Number Eighteen Plus or Minus Three: A Study of Software Technology Maturation*, Report prepared for the DoD STARS Joint Program Office by the Computer and Software Engineering Division of the Institute for Defense Analysis, SDAM/12.

Rogers, E.M. (1983). *Diffusion of Innovations* (3rd Edition). NY: The Free Press.

Teece, D. (1980). "The Diffusion of an Administrative Innovation," *Management Science*, 26, (May), pp. 464-470.

Tornatzky, L., and K. Klein (1982). "Innovation Characteristics and Innovation Adoption-Implementation: A Meta-Analysis of Findings," *IEEE Transactions on Engineering Management*, EM-29, pp. 28-45.

Zmud, R. (1982). "Diffusion of Modern Software Practices: Influence of Centralization and Formalization," *Management Science*, 28, pp. 1421-1431.

Zmud, R., and L. Apple (1986). *Measuring the Institutionalization of the Multi-Business Unit Innovation*, Working Paper, Chapel Hill, NC, University of North Carolina, School of Business Administration.

Appendix A. Questions on Which Analysis Is Based

Appendix A.1. Structured Programming Questions

In this interview, I'm going to ask you questions about your organization's use of Structured Programming techniques. Some questions may not be applicable for your organization. For questions which are not applicable, tell me. We are interested in getting information from organizations that are just beginning to develop Structured Programming capabilities or have considered using Structured Programming but have decided not to, as well as those that are.

1a. Has your organization EVER developed Structured Programming capabilities? This may have involved such tasks as training and/or hiring personnel.

Yes ____ No ____

1b. Approximately when did your organization begin developing Structured Programming capabilities?

Month____ Year____

1c. We would like to know who, in your organization, was the primary advocate for the decision to develop a capability for using Structured Programming?

Is this person a member of [INTERVIEWER: CHECK ONE]

____top management,
____middle management,
____technical staff, or
____decision to develop Structured Programming capabilities was based on broad support of technical management or staff.

1d. In your opinion, how smooth has the process of developing Structured Programming capabilities been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH".

1 2 3 4 5 6 7

2a. Has Structured Programming ever been used in your organization for a pilot or test project?

Yes____ No____

2b. Approximately when did your organization use Structured Programming for a pilot or test project?

Month____ Year____

2c. In your opinion, how smooth has the process of using Structured Programming for a pilot or test project been? Please respond by giving me any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH".

1 2 3 4 5 6 7

3a. Has Structured Programming EVER been used in your organization in a production environment - that is, for any complete software-development projects, rather than on a trial basis?

Yes____ No____

3b. When did your organization begin using Structured Programming in a production environment? Please give me an approximate month and year.

Month____ Year____

3c. In your opinion, how smooth has the process of using Structured Programming in a production environment been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH".

1 2 3 4 5 6 7

4. There are many reasons why an organization might decide to develop a capability for using Structured Programming. To what extent was each of these reasons relevant to your organization's decision to consider development of a capability for using Structured Programming? For each, please give any number between 1 and 7 with 1 meaning "NOT AT ALL RELEVANT" to 7 meaning "VERY RELEVANT".

____Use of Structured Programming will be mandated for future government contracts;
____We believe developing Structured Programming capabilities will make our organization more competitive in getting government contracts;
____We believe developing Structured Programming capabilities will make our organization more competitive in getting consulting projects with government contractors;
____Software engineers working in OUR organization told people here about the desirability of using Structured Programming;
____Colleagues in OTHER organizations told us about the advantages of using Structured Programming;
____Upper management believed that having capabilities for using Structured Programming would benefit the organization;
____Competitors were developing Structured Programming capabilities
____Other________________________________

5. Now we would like to know some of your opinions about Structured Programming. For each of the following statements, please indicate the extent to which you agree or disagree with that statement. For each, please give any number between 1 and 7, where 1 means you "STRONGLY DISAGREE" with the statement and 7 means you "STRONGLY AGREE".

a.	Use of Structured Programming is compatible with software engineering practice in my organization.	1 2 3 4 5 6 7
b.	Organizations that use Structured Programming will be more likely to be granted government contracts.	1 2 3 4 5 6 7
c.	Use of Structured Programming is appropriate for software engineering tasks.	1 2 3 4 5 6 7
d.	The earlier an organization develops Structured Programming capabilities the more likely it will receive government contracts.	1 2 3 4 5 6 7
e.	Personnel familiar with other software development methods can easily be trained to use Structured Programming.	1 2 3 4 5 6 7
f.	Use of Structured Programming is more appropriate for military applications than for commercial applications.	1 2 3 4 5 6 7
g.	Organizations that develop Structured Programming capabilities within the next year will be perceived as being leaders in software development.	1 2 3 4 5 6 7
h.	Organizations should have a "wait and see" attitude until technical problems with Structured Programming have been ironed out.	1 2 3 4 5 6 7
i.	Organizations that currently use Structured Programming are more innovative than those that do not.	1 2 3 4 5 6 7
j.	Training costs to instruct users of Structured Programming are steep.	1 2 3 4 5 6 7
k.	Cost-to-benefit ratio of adopting Structured Programming is less favorable to the adopting company than outside developers realize.	1 2 3 4 5 6 7
l.	Maintenance costs of software developed using Structured Programming is unacceptably high.	1 2 3 4 5 6 7

m. Use of Structured Programming does not yield sufficient economic benefits for our company. 1 2 3 4 5 6 7

n. Technical staff are heavily committed to old software development methods which they feel work very well for them. 1 2 3 4 5 6 7

o. Technical staff are skeptical about the technical value of using Structured Programming. 1 2 3 4 5 6 7

p. Technical staff have no motivation to adopt Structured Programming since benefits would be realized only at corporate level. 1 2 3 4 5 6 7

q. With respect to Structured Programming, technical staff feel that they are being used as guinea pigs in a management or government experiment. 1 2 3 4 5 6 7

r. Production pressures are such that technical personnel cannot easily take time to learn Structured Programming methods. 1 2 3 4 5 6 7

s. Developing Capabilities for using Structured Programming interferes with on-going development processes. 1 2 3 4 5 6 7

6. We would also like to know to what extent YOUR OWN organization has provided each of several types of transition mechanisms to users of Structured Programming in YOUR organization.

6a First consider transition mechanisms provided by your organization DURING THE APPROVAL PROCESS. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT".

____Structured Programming training prepared by in-house personnel
____Structured Programming training prepared by outside personnel
____sending personnel to seminars or conferences
____providing written documentation about Structured Programming or articles from technical or scholarly journals
____visiting organizations where Structured Programming is used
____tools to aid transition

6b Next, consider transition mechanisms provided by your organization WHILE Structured Programming capabilities WERE BEING DEVELOPED. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT".

____Structured Programming training prepared by in-house personnel
____Structured Programming training prepared by outside personnel
____sending personnel to seminars or conferences
____providing written documentation about Structured Programming or articles from technical or scholarly journals
____visiting organizations where Structured Programming is used
____tools to aid transition

6c Now, consider transition mechanisms provided by your organization WHILE Structured Programming was BEING USED FOR A PILOT OR TEST PROJECT. For each, please give any number from 1 to 7, using the same scale as before.

____Structured Programming training prepared by in-house personnel
____Structured Programming training prepared by outside personnel
____sending personnel to seminars or conferences
____providing written documentation about Structured Programming or articles from technical or scholarly journals
____visiting organizations where Structured Programming is used
____tools to aid transition

6d Finally, consider mechanisms provided by your organization WHILE Structured Programming was BEING USED IN A PRODUCTION ENVIRONMENT. For each, please give any number from 1 to 7, using the same scale as before.

____Structured Programming training prepared by in-house personnel
____Structured Programming training prepared by outside personnel
____sending personnel to seminars or conferences
____providing written documentation about Structured Programming or articles from technical
or scholarly journals
____visiting organizations where Structured Programming is used
____tools to aid transition

7. In this question, we want to find out the relative amount of a firm's resources used by different transition mechanisms. I am going to read a list of six transition mechanisms. If you have the questionnaire in front of you, it might help at this time to look at it. After I read the list, I would like you to divide 100 points among the transition mechanisms in a way that reflects your judgment as to the relative amount of organizational resources used by each. For example, if each require the same level of resources, you allocate about 17 points to each. If one requires 40% of the resources, you allocate 40 points to that one, and allocate the other 60 points to the remaining mechanisms.

The transition mechanisms are 1) Structured Programming TRAINING PREPARED BY IN-HOUSE PERSONNEL, 2) Structured Programming TRAINING PREPARED BY

OUTSIDE PERSONNEL, 3) SENDING PERSONNEL TO SEMINARS OR CONFERENCES 4) PROVIDING WRITTEN DOCUMENTATION ABOUT Structured Programming OR ARTICLES ABOUT Structured Programming FROM TECHNICAL OR SCHOLARLY JOURNALS, 5) VISITS TO OTHER ORGANIZATIONS WHERE Structured Programming ARE USED, AND 6) TOOLS TO AID TRANSITION.
Now, please allocate points to each. of the 100 points, how many do you allocate to: [INTERVIEWER: RECORD POINTS NEXT TO EACH. CHECK TO MAKE SURE 100 POINTS ARE ALLOCATED.]

____training prepared by in-house personnel
____training prepared by outside personnel
____sending personnel to seminars or conferences
____providing written documentation about Structured Programming or articles from technical or scholarly journals
____visiting organizations where Structured Programming is used
____tools to aid transition

Appendix A.2. Program Design Language Questions

In this interview, I'm going to ask you questions about Program Design Languages. Some of the questions may not be applicable for your organization. For questions which are not applicable, just tell me. We are interested in getting information from organizations that are just beginning to use Program Design Languages, or have considered using Program Design Languages, but have decided not to use them, as well as those that are.

1a. Has your organization EVER developed any capabilities for using Program Design Languages? This may have involved such tasks as training and/or hiring personnel. We are also including possible acquisition of a specific computer-based Program Design Language here.

Yes____ No____

1b. Approximately when did your organization begin developing Program Design Languages capabilities?

Month____ Year____

1c. We would like to know who, in your organization, was the primary advocate for the decision to develop capabilities for using Program Design Languages?

Is this person a member of [INTERVIEWER: CHECK ONE]

____top management,
____middle management,
____technical staff, or
____the decision to develop PDL capabilities was based on broad support of technical management or staff.

1d. In your opinion, how smooth has the process of developing Program Design Languages capabilities been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH".

1 2 3 4 5 6 7

2a. Have Program Design Languages ever been used in your organization for a pilot or test project?

Yes____ No____

2b. Approximately when did your organization use Program Design Languages for a pilot or test project?

Month____ Year____

2c. In your opinion, how smooth has the process of using Program Design Languages for a pilot or test project been? Please respond by giving me any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."
".

1 2 3 4 5 6 7

3a. Have Program Design Languages ever been used in your organization in a production environment?

Yes____ No____

3b. When did your organization begin using Program Design Languages in a production environment? Please give me an approximate month and year.

Month____ Year____

3c. In your opinion, how smooth has the process of using Program Design Languages in a production environment been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

4. There are many reasons why an organization might decide to develop a Program Design Language capability. To what extent was each of these reasons relevant to your organization's decision to consider development of a Program Design Language capability? For each, please give any number between 1 and 7 with 1 meaning "NOT AT ALL RELEVANT" to 7 meaning "VERY RELEVANT."

____Program Design Languages will be mandated for future government contracts;
____We believe developing Program Design Languages capabilities will make our organization more competitive in getting government contracts;
____make Program Design Language tools;
____We believe developing Program Design Languages capabilities will make our organization more competitive in getting consulting projects with government contractors;
____Software engineers working in OUR organization told people here about the desirability of having Program Design Language capabilities;
____Colleagues in OTHER organizations told us about advantages of using Program Design Languages

____Upper management believed having Program Design Languages capabilities would benefit the organization;
____Competitors were developing Program Design Languages capabilities.
____Other ______________________________

5. Now we would like to know some of your opinions about Program Design Languages. For each of the following statements, please indicate the extent to which you agree or disagree with that statement. For each, please give any number between 1 and 7, where 1 means you "STRONGLY DISAGREE" with the statement and 7 means you "STRONGLY AGREE."

a. Use of Program Design Languages is compatible with software engineering practice in my organization. 1 2 3 4 5 6 7

b. Organizations that develop Program Design Language capabilities will be more likely to be granted government contracts. 1 2 3 4 5 6 7

c. Program Design Languages are appropriate tools for software engineering tasks. 1 2 3 4 5 6 7

d. The earlier an organization develops Program Design Languages capabilities, the more likely it will receive government contracts. 1 2 3 4 5 6 7

e. Personnel familiar with other software design methods can easily be trained to use Program Design Languages. 1 2 3 4 5 6 7

f. Use of Program Design Languages is more appropriate for military applications than for commercial applications. 1 2 3 4 5 6 7

g. Organizations should have a "wait and see" attitude until technical problems with Program Design Languages have been ironed out. 1 2 3 4 5 6 7

h. Organizations that develop Program Design Language capabilities within the next year will be perceived as being leaders in software development. 1 2 3 4 5 6 7

i. Organizations that currently have Program Design Language capabilities are more innovative than those that do not. 1 2 3 4 5 6 7

j. Training costs for the introduction of Program Design Languages are steep. 1 2 3 4 5 6 7

k. Cost-to-benefit ratio of adopting Program Design Languages is less favorable to the adopting company than outside developers realize. 1 2 3 4 5 6 7

l. Performance quality of Program Design Languages is too low to justify developing a Program Design Language capability at this time. 1 2 3 4 5 6 7

m. Use of Program Design Languages does not yield sufficient economic benefits for our company. 1 2 3 4 5 6 7

n. Return on investment for Program Design Languages is too long term. 1 2 3 4 5 6 7

o. Technical staff are heavily committed to old system design methods which they feel work very well for them. 1 2 3 4 5 6 7

p. Technical staff are skeptical about the technical value of Program Design Languages. 1 2 3 4 5 6 7

q. Technical staff have no motivation to adopt Program Design Languages since benefits would be realized only at corporate level. 1 2 3 4 5 6 7

r. With respect to Program Design Languages, technical staff feel that they are being used as guinea pigs in a management or government experiment. 1 2 3 4 5 6 7

s. Production pressures are such that technical personnel cannot easily take time to learn Program Design Languages. 1 2 3 4 5 6 7

t. Developing Program Design Language capabilities interferes with on-going development processes. 1 2 3 4 5 6 7

6. We would also like to know to what extent YOUR OWN organization has provided each of the following types of transition mechanisms to users of Program Design Languages in YOUR organization.

6a First we will consider transition mechanisms provided by your organization BEFORE ACQUISITION, DURING THE APPROVAL PROCESS. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT." [INTERVIEWER: READ LIST AND RECORD RESPONSE]

____Program Design Language training prepared by in-house personnel
____Program Design Language training prepared by outside personnel
____sending personnel to seminars or conferences
____providing written documentation about Program Design Languages or articles from technical or scholarly journals
____visiting other organizations where there are users of PDLs
____tool to aid transition

6b Next, consider transition mechanisms provided by your organization WHILE PROGRAM DESIGN LANGUAGE CAPABILITIES WERE BEING DEVELOPED. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT."

____Program Design Language training prepared by in-house personnel
____Program Design Language training prepared by outside personnel
____sending personnel to seminars or conferences
____providing written documentation about Program Design Languages or articles from technical or scholarly journals
____visiting other organizations where there are users of PDLs
____tools to aid transition

6c Now, consider transition mechanisms provided by your organization WHILE A PROGRAM DESIGN LANGUAGE WAS BEING USED FOR A PILOT OR TEST PROJECT. For each, please give any number from 1 to 7, using the same scale as before.

____Program Design Language training prepared by in-house personnel
____Program Design Language training prepared by outside personnel
____sending personnel to seminars or conferences
____providing written documentation about Program Design Languages or articles from technical or scholarly journals
____visiting other organizations where there are users of PDLs
____tools to aid transition

6d Now, consider transition mechanisms provided by your organization WHILE A PROGRAM DESIGN LANGUAGE WAS BEING USED IN A PRODUCTION ENVIRONMENT. For each, please give any number from 1 to 7, using the same scale as before.

____Program Design Language training prepared by in-house personnel
____Program Design Language training prepared by outside personnel
____sending personnel to seminars or conferences
____providing written documentation about Program Design Languages or articles from technical or scholarly journals
____visiting other organizations where there are users of PDLs
____tools to aid transition

7. In this question, we want to find out the relative amount of a firm's resources used by different transition mechanisms. I am going to read a list of six transition mechanisms. If you have the questionnaire in front of you, it might help at this time to look at that questionnaire. After I read the list, I would like you to divide 100 points among the transition mechanisms in a way that reflects your judgment as to the relative amount of organizational resources used by each. For example, if each require the same level of resources, you allocate about 17 points to each. If one requires 40% of the resources, you allocate 40 points to that one, and allocate the other 60 points to the remaining transition mechanisms.

The transition mechanisms are 1) Program Design Language TRAINING PREPARED BY IN-HOUSE PERSONNEL, 2) Program Design Language TRAINING PREPARED BY OUTSIDE PERSONNEL, 3) SENDING PERSONNEL TO SEMINARS OR CONFERENCES, 4) PROVIDING WRITTEN DOCUMENTATION ABOUT PDLs OR ARTICLES ABOUT PDLs FROM TECHNICAL OR SCHOLARLY JOURNALS, 5) VISITS TO OTHER ORGANIZATIONS WHERE THERE ARE USERS of PDLs, AND 6) TOOLS TO AID TRANSITION. I'll repeat them again at any point.

Now, please allocate points to each. of the 100 points, how many do you allocate to: [INTERVIEWER: RECORD POINTS NEXT TO EACH. CHECK TO MAKE SURE 100 POINTS ARE ALLOCATED.]

_____training in Program Design Languages prepared by in-house personnel
_____training in Program Design Languages prepared by outside personnel
_____sending personnel to seminars or conferences
_____provide written documentation about Program Design Languages or articles about Program Design Languages from technical or scholarly journals
_____visit other organizations where there are users of PDLs
_____tools to aid transition

Appendix A.3. Software Cost Models Questions

In this interview, I'm going to ask you questions about your organization's use of Software Cost Models. An example of a software cost model used by some organizations is COCOMO, a model in which the cost of computer software is modeled as a function of the product, computer, personnel and project. Some of the questions may not be applicable for your organization. For questions which are not applicable, just tell me. We are interested in getting information from organizations that are just beginning to develop capabilities for using Software Cost Models or have considered Software Cost Models, but have decided not to use them, as well as those that are.

1a. Has your organization ever developed capabilities for using Software Cost Models? This may have involved such tasks as training and/or hiring personnel. We are also including possible acquisition of a Software Cost Model software package here, as well as possibly developing Software Cost Models in-house, or deriving Software Cost Models from published literature.

Yes____ No____

1b. Approximately when did your organization begin developing Capabilities for using Software Cost Models?

Month____ Year____

1c. We would like to know who, in your organization, was the primary advocate for the decision to develop a capability for using Software Cost Models? Is this person a member of [INTERVIEWER: CHECK ONE]

____top management,
____middle management,
____technical staff, or
____decision to develop Software Cost Model capabilities was based on broad support of technical management or staff.

1d. In your opinion, how smooth has the process of developing capabilities for using Software Cost Models been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

2a. Have Software Cost Models ever been used in your organization for a pilot or test project?

Yes____ No____

2b. Approximately when did your organization use Software Cost Models for a pilot or test project?

Month____ Year____

2c. In your opinion, how smooth has the process of using Software Cost Models for a pilot or test project been? Please respond by giving me any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

3a. Have Software Cost Models ever been used in your organization in a production environment - that is, for any complete software-development projects, rather than on a trial basis?

Yes____ No____

3b. When did your organization begin using Software Cost Models in a production environment? Please give me an approximate month and year.

Month____ Year____

3c. In your opinion, how smooth has the process of using Software Cost Models in a production environment been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

4. There are many reasons why an organization might decide to develop a capability for using Software Cost Models. To what extent was each of these reasons relevant to your organization's decision to consider development of a capability for using Software Cost Models? For each, please give any number between 1 and 7 with 1 meaning "NOT AT ALL RELEVANT" to 7 meaning "VERY RELEVANT."

____Use of Software Cost Models will be mandated for future government contracts;
____We believe developing Capabilities for using Software Cost Models will make our organization more competitive in getting government contracts;
____We develop Software Cost Model software;
____We believe developing Capabilities for using Software Cost Models will make our organization more competitive in getting consulting projects with government contractors;
____Software engineers working in OUR organization told people here about the desirability of having Capabilities for using Software Cost Models;
____Colleagues in OTHER organizations told us about the advantages of using Software Cost Models;

____Upper management believed that having Capabilities for using Software Cost Models would benefit the organization;
____Competitors were developing Capabilities for using Software Cost Models.
____ Other ________________________________

5. Now we would like to know some of your opinions about Software Cost Models. For each of the following statements, please indicate the extent to which you agree or disagree with that statement. For each, please give any number between 1 and 7, where 1 means you "STRONGLY DISAGREE" with the statement and 7 means you "STRONGLY AGREE" with the statement.

a.	Use of software cost Models is compatible with software engineering practice in my organization.	1 2 3 4 5 6 7
b.	Organizations that use Software Cost Models will be more likely to be granted government contracts.	1 2 3 4 5 6 7
c.	Use of Software Cost Models is appropriate for software engineering tasks.	1 2 3 4 5 6 7
d.	The earlier an organization develops Capabilities for using Software Cost Models, the more likely it will receive government contracts.	1 2 3 4 5 6 7
e.	Personnel familiar with other software cost estimation techniques can easily be trained to use Software Cost Models.	1 2 3 4 5 6 7
f.	Use of Software Cost Models is more appropriate for military applications than for commercial applications.	1 2 3 4 5 6 7
g.	Organizations that develop capabilities for using Software Cost Models within the next year will be perceived as being leaders in software development.	1 2 3 4 5 6 7
h.	Organizations should have a "wait and see" attitude until technical problems with Software Cost Models have been ironed out.	1 2 3 4 5 6 7

i.	Organizations that currently have Capabilities for using Software Cost Models are more innovative than those that do not.	1 2 3 4 5 6 7
j.	Costs to train people to use Software Cost Models are steep.	1 2 3 4 5 6 7
k.	Cost-to-benefit ratio of adopting Software Cost Models is less favorable to the adopting company than outside developers realize.	1 2 3 4 5 6 7
l.	Performance accuracy of Software Cost Models is too low to justify using them at this time.	1 2 3 4 5 6 7
m.	Use of Software Cost Models does not yield sufficient economic benefits for our company.	1 2 3 4 5 6 7
n.	Technical staff are heavily committed to old software cost estimation techniques which they feel work very well for them.	1 2 3 4 5 6 7
o.	Technical staff are skeptical about the technical value of using Software Cost Models.	1 2 3 4 5 6 7
p.	Technical staff have no motivation to adopt Software Cost Models since benefits would be realized only at corporate level.	1 2 3 4 5 6 7
q.	With respect to Software Cost Models, technical staff feel that they are being used as guinea pigs in a management or government experiment.	1 2 3 4 5 6 7
r.	Production pressures are such that technical personnel cannot easily take time to learn to use Software Cost Models.	1 2 3 4 5 6 7
s.	Developing Capabilities for using Software Cost Models interferes with on-going development processes.	1 2 3 4 5 6 7

6. We would also like to know to what extent YOUR OWN organization has provided each of several types of transition mechanisms to users of Software Cost Models in YOUR organization.

6a First consider transition mechanisms provided by your organization BEFORE ACQUISITION, DURING THE APPROVAL PROCESS. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT."

____training in Software Cost Models prepared by in-house personnel
____training in Software Cost Models prepared by outside personnel
____providing written documentation about Software Cost Models or articles from technical or scholarly journals
____visiting other organizations where Software Cost Models are used

6b Next, consider transition mechanisms provided by your organization WHILE Capabilities for using Software Cost Models WERE BEING DEVELOPED. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT."

____training in Software Cost Models prepared by in-house personnel
____training in Software Cost Models prepared by outside personnel
____providing written documentation about Software Cost Models or articles from technical or scholarly journals
____visiting other organizations where Software Cost Models are used

6c Now, consider transition mechanisms provided by your organization WHILE Software Cost Models WERE BEING USED FOR A PILOT OR TEST PROJECT. For each, please give any number from 1 to 7, using the same scale as before.

____training in Software Cost Models prepared by in-house personnel
____training in Software Cost Models prepared by outside personnel
____providing written documentation about Software Cost Models or articles from technical or scholarly journals
____visiting other organizations where Software Cost Models are used

6d Finally, consider mechanisms provided by your organization WHILE Software Cost Models WERE BEING USED IN A PRODUCTION ENVIRONMENT. For each, please give any number from 1 to 7, using the same scale as before.

____training in Software Cost Models prepared by in-house personnel
____training in Software Cost Models prepared by outside personnel
____providing written documentation about Software Cost Models or articles from technical or scholarly journals
____visiting other organizations where Software Cost Models are used

7. In this question, we want to find out the relative amount of a firm's resources used by different transition mechanisms. I am going to read a list of four transition mechanisms. If you have the questionnaire in front of you, it might help at this time to look at it. After I read the list, I would like you to divide 100 points among the transition mechanisms in a way that reflects your judgment as to the relative amount of organizational resources used by each. For example, if each require the same level of resources, you

allocate 25 points to each. If one requires 40% of the resources, you allocate 40 points to that one, and allocate the other 60 points to the remaining mechanisms.

The transition mechanisms are 1) Software Cost Models TRAINING PREPARED BY IN-HOUSE PERSONNEL, 2) Software Cost Models TRAINING PREPARED BY OUTSIDE PERSONNEL, 3) PROVIDING WRITTEN DOCUMENTATION ABOUT Software Cost Models OR ARTICLES ABOUT Software Cost Models FROM TECHNICAL OR SCHOLARLY JOURNALS, AND 4) VISITS TO OTHER ORGANIZATIONS WHERE SOFTWARE COST MODELS ARE USED. I'll repeat them again at any point.

Now, please allocate points to each. of the 100 points, how many do you allocate to: [INTERVIEWER: RECORD POINTS NEXT TO EACH. CHECK TO MAKE SURE 100 POINTS ARE ALLOCATED.]

____training prepared by in-house personnel
____training prepared by outside personnel
____providing written documentation about Software Cost Models or articles from technical or scholarly journals
____visiting other organizations where Software Cost Models are used

Appendix A.4. Complexity Metrics Questions

In this interview, I'm going to ask you questions about your organization's use of Complexity Metrics. Examples of software complexity metrics used by some organizations are Halstead's effort equation and McCabe's cyclomatic complexity measure. As a rule most of these metrics incorporate easily computed properties of source code.

Some of the questions may not be applicable for your organization. For questions which are not applicable, just tell me. We are interested in getting information from organizations that are just beginning to develop capabilities for using Complexity Metrics or have considered Complexity Metrics, but have decided not to use them, as well as those that are.

1a. Has your organization ever developed capabilities for using Complexity Metrics? This may have involved such tasks as training and/or hiring personnel. Here we are also including possibly developing Complexity Metrics in-house, or deriving Complexity Metrics from published literature.

Yes____ No____

1b. Approximately when did your organization begin developing Capabilities for using Complexity Metrics?

Month____ Year____

1c. We would like to know who, in your organization, was the primary advocate for the decision to develop a capability for using Complexity Metrics?

Is this person a member of [INTERVIEWER: CHECK ONE]

____top management,
____middle management,
____technical staff or
____decision to develop Complexity Metrics capabilities was based on broad support of technical management or staff.

1d. In your opinion, how smooth has the process of developing capabilities for using Complexity Metrics been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

2a. Have Complexity Metrics ever been used in your organization for a pilot or test project?

Yes____ No____

2b. Approximately when did your organization use Complexity Metrics for a pilot or test project?

Month____ Year____

2c. In your opinion, how smooth has the process of using Complexity Metrics for a pilot or test project been? Please respond by giving me any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

3a. Have Complexity Metrics ever been used in your organization in a production environment - that is, for any complete software-development projects, rather than on a trial basis?

Yes____ No____

3b. When did your organization begin using Complexity Metrics in a production environment? Please give me an approximate month and year.

Month____ Year____

3c. In your opinion, how smooth has the process of using Complexity Metrics in a production environment been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

4. There are many reasons why an organization might decide to develop a capability for using Complexity Metrics. To what extent was each of these reasons relevant to your organization's decision to consider development of a capability for using Complexity Metrics? For each, please give any number between 1 and 7 with 1 meaning "NOT AT ALL RELEVANT" to 7 meaning "VERY RELEVANT."

____Use of Complexity Metrics will be mandated for future government contracts;

____We believe developing capabilities for using Complexity Metrics will make our organization more competitive in getting government contracts;

____We believe developing capabilities for using Complexity Metrics will make our organization more competitive in getting consulting projects with government contractors;

____Software engineers working in OUR organization told people here about the desirability of having Capabilities for using Complexity Metrics;

____Colleagues in OTHER organizations told us about the advantages of using Complexity Metrics;

____Upper management believed that having capabilities for using Complexity Metrics would benefit the organization;

____Competitors were developing capabilities for using Complexity Metrics.

____ Other________________________________

5. Now we would like to know some of your opinions about Complexity Metrics. For each of the following statements, please indicate the extent to which you agree or disagree with that statement. For each, please give any number between 1 and 7, where 1 means you "STRONGLY DISAGREE" with the statement and 7 means you "STRONGLY AGREE" with the statement.

a. Use of Complexity Metrics is compatible with software engineering practice in my organization. 1 2 3 4 5 6 7

b. Organizations that use Complexity Metrics will be more likely to be granted government contracts. 1 2 3 4 5 6 7

c. Use of Complexity Metrics is appropriate for software engineering tasks. 1 2 3 4 5 6 7

d. The earlier an organization develops Capabilities for using Complexity Metrics, the more likely it will receive government contracts. 1 2 3 4 5 6 7

e. Use of Complexity Metrics is more appropriate for military applications than for commercial applications. 1 2 3 4 5 6 7

f. Organizations that develop capabilities for using Complexity Metrics within the next year will be perceived as being leaders in software development. 1 2 3 4 5 6 7

g. Organizations should have a "wait and see" attitude until technical problems with Complexity Metrics have been ironed out. 1 2 3 4 5 6 7

h. Organizations that currently have capabilities for using Complexity Metrics are more innovative than those that do not. 1 2 3 4 5 6 7

i. Costs to train people to use Complexity Metrics are steep. 1 2 3 4 5 6 7

j. Cost-to-benefit ratio of adopting Complexity Metrics is less favorable to the adopting company than outsiders realize. 1 2 3 4 5 6 7

k. Performance accuracy of Complexity Metrics is too low to justify using them at this time. 1 2 3 4 5 6 7

l. Use of Complexity Metrics does not yield sufficient economic benefits for our company. 1 2 3 4 5 6 7

m. Technical staff are skeptical about the technical value of using Complexity Metrics. 1 2 3 4 5 6 7

n. Technical staff have no motivation to adopt Complexity Metrics since benefits would be realized only at corporate level. 1 2 3 4 5 6 7

o. With respect to Complexity Metrics, technical staff feel that they are being used as guinea pigs in a management or government experiment. 1 2 3 4 5 6 7

p. Production pressures are such that technical personnel cannot easily take time to learn to use Complexity Metrics. 1 2 3 4 5 6 7

q. Developing capabilities for using Complexity Metrics interferes with on-going development processes. 1 2 3 4 5 6 7

6. We would also like to know to what extent YOUR OWN organization has provided each of several types of transition mechanisms to users of Complexity Metrics in YOUR organization.

6a First consider transition mechanisms provided by your organization BEFORE ACQUISITION, DURING THE APPROVAL PROCESS. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT."

____training in Complexity Metrics prepared by in-house personnel
____training in Complexity Metrics prepared by outside personnel

____providing written documentation about Complexity Metrics or articles from technical or scholarly journals
____visiting other organizations where Complexity Metrics are used

6b Next, consider transition mechanisms provided by your organization WHILE capabilities for using Complexity Metrics WERE BEING DEVELOPED. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT."

____training in Complexity Metrics prepared by in-house personnel
____training in Complexity Metrics prepared by outside personnel
____providing written documentation about Complexity Metrics or articles from technical or scholarly journals
____visiting other organizations where Complexity Metrics are used

6c Now, consider transition mechanisms provided by your organization WHILE Complexity Metrics WERE BEING USED FOR A PILOT OR TEST PROJECT. For each, please give any number from 1 to 7, using the same scale as before.

____training in Complexity Metrics prepared by in-house personnel
____training in Complexity Metrics prepared by outside personnel
____providing written documentation about Complexity Metrics or articles from technical or scholarly journals
____visiting other organizations where Complexity Metrics are used

6d Finally, consider mechanisms provided by your organization WHILE Complexity Metrics WERE BEING USED IN A PRODUCTION ENVIRONMENT. For each, please give any number from 1 to 7, using the same scale as before.

____training in Complexity Metrics prepared by in-house personnel
____training in Complexity Metrics prepared by outside personnel
____providing written documentation about Complexity Metrics or articles from technical or scholarly journals
____visiting other organizations where Complexity Metrics are used

7. In this question, we want to find out the relative amount of a firm's resources used by different transition mechanisms. I am going to read a list of four transition mechanisms. If you have the questionnaire in front of you, it might help at this time to look at it. After I read the list, I would like you to divide 100 points among the transition mechanisms in a way that reflects your judgment as to the relative amount of organizational resources used by each. For example, if each require the same level of resources, you allocate 25 points to each. If one requires 40% of the resources, you allocate 40 points to that one, and allocate the other 60 points to the remaining mechanisms.

The transition mechanisms are 1) Complexity Metrics TRAINING PREPARED BY IN-HOUSE PERSONNEL, 2) Complexity Metrics TRAINING PREPARED BY OUTSIDE PERSONNEL, 3) PROVIDING WRITTEN DOCUMENTATION ABOUT Complexity Metrics OR ARTICLES ABOUT Complexity Metrics FROM TECHNICAL OR SCHOLARLY JOURNALS, AND 4)

VISITS TO OTHER ORGANIZATIONS WHERE Complexity Metrics ARE USED. I'll repeat them again at any point.

Now, please allocate points to each. of the 100 points, how many do you allocate to: [INTERVIEWER: RECORD POINTS NEXT TO EACH. CHECK TO MAKE SURE 100 POINTS ARE ALLOCATED.]

____training prepared by in-house personnel
____training prepared by outside personnel
____providing written documentation about Complexity Metrics or articles from technical or scholarly journals
____visiting other organizations where Complexity Metrics are used

Appendix A.5. Ada Questions

In this interview, I'm going to ask you questions about Ada. Some of the questions may not be applicable for your organization. For questions which are not applicable, just tell me. We are interested in getting information from organizations that are just beginning to develop Ada capabilities or have considered Ada, but have decided not to use it, as well as those that are.

1a. Has your organization acquired an Ada compiler?

Yes____ No____

1b. Approximately when did your organization acquire an Ada compiler?

Month____ Year____

1d. In your opinion, how smooth has the process of acquiring an Ada compiler been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

2a. Has your organization EVER developed any Ada capabilities? This may have involved such tasks as training and/or hiring personnel. We are NOT including acquisition of an Ada compiler here.

Yes____ No____

2b. Approximately when did your organization begin developing Ada capabilities?

Month____ Year____

2c. We would like to know who, in your organization, was the primary advocate for the decision to develop an Ada capability?

Is this person a member of [INTERVIEWER: CHECK ONE]

____top management,
____middle management,
____technical staff, or
____ decision to develop Ada capabilities was based on broad support of technical management or staff.

2d. In your opinion, how smooth has the process of developing Ada capabilities been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

3a. Has Ada ever been used in your organization for a pilot or test project?

Yes____ No____

3b. Approximately when did your organization use Ada for a pilot or test project?

Month____ Year____

3c. In your opinion, how smooth has the process of using Ada for a pilot or test project been? Please respond by giving me any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

4a. Has Ada ever been used in your organization in a production environment?

Yes____ No____

4b. When did your organization begin using Ada in a production environment? Please give me an approximate month and year.

Month____ Year____

4c. In your opinion, how smooth has the process of using Ada in a production environment been? Please respond with any number between 1 and 7 where 1 means "THE PROCESS HAS NOT BEEN AT ALL SMOOTH" and 7 means "THE PROCESS HAS BEEN EXTREMELY SMOOTH."

1 2 3 4 5 6 7

5. There are many reasons why an organization might decide to develop an Ada capability. To what extent was each of these reasons relevant to your organization's decision to consider development of an Ada capability? For each, please give any number between 1 and 7 with 1 meaning "NOT AT ALL RELEVANT" to 7 meaning "VERY RELEVANT."

____Ada will be mandated for future government contracts;
____We believe developing Ada capabilities will make our organization more competitive in getting government contracts;
____We make third party Ada support tools or compilers;
____We believe developing Ada capabilities will make our organization more competitive in getting consulting projects with government contractors;
____Software engineers working in OUR organization told people here about the desirability of having Ada capabilities;
____Colleagues in OTHER organizations told us about Ada's advantages
____Upper management believed having Ada capabilities would benefit the organization;
____Competitors were developing Ada capabilities.
____Other ____

6. Now we would like to know some of your opinions about Ada. For each of the following statements, please indicate the extent to which you agree or disagree with that statement. For each, please give any number between 1 and 7, where 1 means you "STRONGLY DISAGREE" with the statement and 7 means you "STRONGLY AGREE" with the statement.

a. Ada is compatible with software engineering practice in my organization. 1 2 3 4 5 6 7

b. Organizations that develop Ada capabilities will be more likely to be granted government contracts. 1 2 3 4 5 6 7

c. Ada is an appropriate language for software engineering tasks. 1 2 3 4 5 6 7

d. The earlier an organization develops Ada capabilities, the more likely it will receive government contracts. 1 2 3 4 5 6 7

e. Personnel familiar with languages like Fortran can easily be trained to program in Ada. 1 2 3 4 5 6 7

f. Ada is a more appropriate programming environment for military applications than for commercial applications. 1 2 3 4 5 6 7

g. There are sufficient Ada tools available to justify developing an Ada capability at this time. 1 2 3 4 5 6 7

h. Organizations should have a "wait and see" attitude regarding the Ada mandate before developing Ada capability. 1 2 3 4 5 6 7

i. Organizations that develop Ada capabilities within the next year will be perceived as being leaders in software development. 1 2 3 4 5 6 7

j. Organizations should have a "wait and see" attitude until technical problems with Ada have been ironed out. 1 2 3 4 5 6 7

k. Organizations that currently have Ada capabilities are more innovative than those that do not. 1 2 3 4 5 6 7

l.	Training costs for the introduction of Ada are steep.	1 2 3 4 5 6 7
m.	Cost-to-benefit ratio of adopting Ada is less favorable to the adopting company than outside developers realize.	1 2 3 4 5 6 7
n.	Performance quality of Ada compilers is too low to justify developing an Ada capability at this time.	1 2 3 4 5 6 7
o.	Ada does not yield sufficient economic benefits for our company.	1 2 3 4 5 6 7
p.	Return on investment for Ada is too long term.	1 2 3 4 5 6 7
q.	Technical staff are heavily committed to old programming languages which they feel work very well for them.	1 2 3 4 5 6 7
r.	Technical staff are skeptical about the technical value of Ada.	1 2 3 4 5 6 7
s.	Technical staff have no motivation to adopt Ada since benefits would be realized only at corporate level.	1 2 3 4 5 6 7
t.	With respect to Ada, technical staff feel that they are being used as guinea pigs in a management or government experiment.	1 2 3 4 5 6 7
u.	Production pressures are such that technical personnel cannot easily take time to learn Ada.	1 2 3 4 5 6 7
v.	Developing Ada capabilities interferes with on-going development processes.	1 2 3 4 5 6 7

7. We would also like to know to what extent YOUR OWN organization has provided each of the following types of transition mechanisms to users of Ada in YOUR organization.

7a First we will consider transition mechanisms provided by your organization BEFORE ACQUISITION, DURING THE APPROVAL PROCESS. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT." [INTERVIEWER: READ LIST AND RECORD RESPONSE]

____training in Ada prepared by in-house personnel
____training in Ada prepared by outside personnel
____sending personnel to seminars or conferences, for example, to SIGAda
____provide written documentation about Ada or articles about Ada from technical or scholarly journals
____visit other organizations where there are Ada users

7b Next, consider transition mechanisms provided by your organization AT COMPILER INSTALLATION. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT."

____training in Ada prepared by in-house personnel
____training in Ada prepared by outside personnel
____sending personnel to seminars or conferences
____provide written documentation about Ada or articles about Ada from technical or scholarly journals
____visit other organizations where there are Ada users

7c Next, consider transition mechanisms provided by your organization WHILE ADA CAPABILITIES WERE BEING DEVELOPED. For each, please give any number from 1 to 7, where 1 means "THE MECHANISM WAS NOT AT ALL PROVIDED", and 7 means "THE MECHANISM WAS PROVIDED TO A VERY GREAT EXTENT."

____training in Ada prepared by in-house personnel
____training in Ada prepared by outside personnel
____sending personnel to seminars or conferences
____provide written documentation about Ada or articles about Ada from technical or scholarly journals
____visit other organizations where there are Ada users

7d Now, consider transition mechanisms provided by your organization WHILE ADA WAS BEING USED FOR A PILOT OR TEST PROJECT. For each, please give any number from 1 to 7, using the same scale as before.

____training in Ada prepared by in-house personnel
____training in Ada prepared by outside personnel
____sending personnel to seminars or conferences
____provide written documentation about Ada or articles about Ada from technical or scholarly journals
____visit other organizations where there are Ada users

7e Now, consider transition mechanisms provided by your organization WHILE ADA WAS BEING USED IN A PRODUCTION ENVIRONMENT. For each, please give any number from 1 to 7, using the same scale as before.

____training in Ada prepared by in-house personnel
____training in Ada prepared by outside personnel
____sending personnel to seminars or conferences

____provide written documentation about Ada or articles about Ada from technical or scholarly journals
____visit other organizations where there are Ada users

8. In this question, we want to find out the relative amount of a firm's resources used by different transition mechanisms. I am going to read a list of five transition mechanisms. If you have the questionnaire in front of you, it might help at this time to look at that questionnaire. After I read the list, I would like you to divide 100 points among the transition mechanisms in a way that reflects your judgment as to the relative amount of organizational resources used by each. For example, if each require the same level of rescurces, you allocate 20 points to each. If one requires 40% of the resources, you allocate 40 points to that one, and allocate the other 60 points to the remaining transition mechanisms.

   The transition mechanisms are 1) ADA TRAINING PREPARED BY IN-HOUSE PERSONNEL, 2) ADA TRAINING PREPARED BY OUTSIDE PERSONNEL, 3) SENDING PERSONNEL TO SEMINARS OR CONFERENCES, FOR EXAMPLE, TO SIGADA, 4) PROVIDING WRITTEN DOCUMENTATION ABOUT ADA OR ARTICLES ABOUT ADA FROM TECHNICAL OR SCHOLARLY JOURNALS, AND 5) VISITS TO OTHER ORGANIZATIONS WHERE THERE ARE ADA USERS. I'll repeat them again at any point.

   Now, please allocate points to each. of the 100 points, how many do you allocate to: [INTERVIEWER: RECORD POINTS NEXT TO EACH. CHECK TO MAKE SURE 100 POINTS ARE ALLOCATED.]

   ____training in Ada prepared by in-house personnel
   ____training in Ada prepared by outside personnel
   ____sending personnel to seminars or conferences
   ____provide written documentation about Ada or articles about Ada from technical or scholarly journals
   ____visit other organizations where there are Ada users

Appendix B. Solicitation Letter

[DATE]

[NAME & ADDRESS]

Dear [NAME]:

We are writing to ask you to participate in the continuation of a research study which we are conducting in cooperation with the assistance of the NSIA Software Committee. We believe this study could prove to be of substantial value to you and your firm. We are faculty members in Carnegie Mellon University's Graduate School of Business working on this study as official members of the Software Engineering Institute, which recently joined the NSIA. We are interested in understanding the process by which organizations such as yours make decisions to reject or integrate new technologies into their businesses. We are writing this letter to ask you to join us as participants in the initial study of this research program, funded in part by the Software Engineering Institute and Carnegie Mellon University.

Undoubtedly, you are besieged with requests like ours; but before you put this letter in the waste basket, please read it to the end. Unlike many faculty members, we entered our academic careers after working for over twenty years (collectively) in strategy development and marketing in firms whose businesses ranged from meeting the engineering needs of the military to financial services. Reflecting on these experiences, we have become part of a small, but growing, group of scholars who are developing research programs that are of practical value to American businesses and of rigorous scientific quality as well. We are writing this letter to ask you to join us in the first phase of a larger research program that reflects these goals.

Specifically, our research program is concerned first with the factors that influence the understanding of the adoption, postponement or rejection of "new" technologies by organizations. This first phase of our research will provide the basis for the next phase, in which we plan to investigate which cost-effective actions an organization might take to accelerate the technology adoption process. Technologies of interest to us are the ADA programming environment, software metrics and cost-estimation procedures, e-mail, program development languages, structured programming, and expert systems. Technology is interpreted broadly as either new tools or methods. We are writing to ask if you and your organization would be willing to serve as a participant in our research. The thirty-four firms that have agreed to participate in the study so far include some of the best-known firms in the software development business. As a participant, your organization would be in excellent company.

Your participation in the study and your organization's involve two steps. First, we need your help as the company contact in locating individuals in your firm who have knowledge of adopt/reject decisions relating to a small set of technologies. In many cases, the appropriate individual may be you. In other cases, it may be someone else in your organization. Similarly, one individual may be knowledgeable about decision processes relating to several technologies. We will always prepare you for our telephone call by sending you a letter indicating the questions we will ask. This will be done approximately two weeks in advance of our telephone call. This should make it easier for you to guide us to a person in your organization who is knowledgeable about the adoption/rejection decisions relating to a specific technology. We anticipate that this will take 10 to 15 minutes of your time. Second, we will telephone these individuals to set up a mutually convenient time for conducting the telephone interview. Prior to conducting this interview, we will help each interviewee by sending him or her a list of the questions we will ask during the telephone interview. We anticipate that an interview for a specific technology will take approximately 20 minutes. At the conclusion of this study, we will provide you with an executive summary of these findings. A more formal presentation may also be arranged if you desire. If you are interested in continuing your relationship with the larger research program, we will offer you that opportunity.

We would very much like to have your participation in the study. We realize that you are very busy, but we believe that our research would be valuable to the practicing manager interested in accelerating technology adoption. If you are interested in having your organization participate in this study, please fill out the enclosed form and return it in the enclosed, self-addressed envelope.

Sincerely,

Judy Bayer, Ph.D.

Assistant Professor of Industrial Administration
(412) 268-8842

Nancy Melone, Ph.D.

Assistant Professor of Industrial Administration
(412) 268-3763

Enclosure

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS NONE
2a. SECURITY CLASSIFICATION AUTHORITY N/A 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE DISTRIBUTION UNLIMITED
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-89-TR-17	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-89-25
6a. NAME OF PERFORMING ORGANIZATION SOFTWARE ENGINEERING INST. 6b. OFFICE SYMBOL (If applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI JOINT PROGRAM OFFICE
6c. ADDRESS (City, State and ZIP Code) CARNEGIE-MELLON UNIVERSITY PITTSBURGH, PA 15213	7b. ADDRESS (City, State and ZIP Code) ESD/XRS1 HANSCOM AIR FORCE BASE HANSCOM, MA 01731
8a. NAME OF FUNDING/SPONSORING ORGANIZATION SEI JOINT PROGRAM OFFICE 8b. OFFICE SYMBOL (If applicable) ESD/XRS1	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962885C0003
8c. ADDRESS (City, State and ZIP Code) CARNEGIE-MELLON UNIVERSITY PITTSBURGH, PA 15213	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: 63752F; PROJECT NO.: N/A; TASK NO.: N/A; WORK UNIT NO.: N/A

11. TITLE (Include Security Classification)
ADOPTION OF SOFTWARE ENGINEERING INNOVATIONS IN ORGANIZATIONS

12. PERSONAL AUTHOR(S)
Judy Bayer and Nancy Melone

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
FINAL	FROM ______ TO ______	April 1989	150

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	technology transition technology adoption diffusion theory

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Designing effective strategies to facilitate the adoption of new software engineering technologies is a complex endeavor. This document describes the experiences of organizations in the defense industry that have considered and in many cases adopted any one of five software engineering technologies: structured programming, program design languages, software cost models, complexity metrics, and Ada. In all, 296 respondents participated in the entire study. These respondents represented approximately 120 business units with approximately 75 defense contractor organizations. Data were collected using a structured survey instrument administered over the telephone.

This report examines the motivations behind technology acquisition and adoption decisions, the use of various technology transfer mechanisms during the stages of the adoption process, and the relationship between technology transfer mechanisms and the timing, pass through, and smoothness of adoption-process stages. Adoption is assumed to be a multi-stage process that may proceed in a linear or non-linear fashion. Also explored is the relationship between managerial level of the advocate (i.e., top management, middle management, technical management, and broad-based support) and the speed and smoothness of technology acquisition and adoption.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☒	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED, UNLIMITED DISTRIBUTION
22a. NAME OF RESPONSIBLE INDIVIDUAL KARL H. SHINGLER	22b. TELEPHONE NUMBER (Include Area Code) 412 268-7630 22c. OFFICE SYMBOL SEI JPO

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE