Technical Report
CMU/SEI-93-TR-31
ESC-TR-93-317

(2)

Carnegie Mellon University
Software Engineering Institute

A Conceptual Framework for Software Technology Transition

Priscilla Fowler
Linda Levine

December 1993

Technical Report
CMU/SEI-93-TR-31
ESC-TR-93-317
December 1993

A Conceptual Framework for Software Technology

Priscilla Fowler
Linda Levine

Transition Models Project

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This technical report was prepared for the

SEI Joint Program Office
ESC/ENS
Hanscom AFB, MA 01731-2116

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

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

The Software Engineering Institute is sponsored by the U.S. Department of Defense.

This report was funded by the U.S. Department of Defense.

Table of Contents

List of Figures

A Conceptual Framework for Software Technology Transition[1]

Abstract: We present a conceptual framework that integrates and describes the intersections of three life cycles of software technology transition: research and development, new product development, and adoption and implementation in organizations. We then apply the framework to the technology transition experiences of the Software Engineering Institute.

1 Background

Technology transition[2] and closely related areas, such as diffusion of innovation [Rogers 1983, Tornatzky & Fleischer 1990] and technology management [Botkin & Matthews 1992, Moore 1991, Roberts 1991, Roussel 1991], have been studied extensively for many years. Recently, more and more researchers and practitioners have grappled with how to improve technology transition in software- and information-intensive technologies such as avionics and high-speed ground transport, medical equipment, and telecommunications networks. The very malleability of software that makes it a desirable component in these technologies also makes transitioning it into practice especially problematic.

One cause of confusion with respect to technology transition is that the subject area means different things to different people. Depending upon one's context, one may think about technology transition in terms of: information dissemination, telecommunications infrastructure, training, sharing software resources on networks, collaboration, patents and licenses, spin-off ventures, or technology introduction and implementation [Williams & Gibson 1990]. That technology transition can be seen as all of these and more—as theory, process, strategy, model, and mechanism—speaks to the complexity of the subject.

Technology transition occurs throughout technology development from the birth of a technology until its retirement. Technology that has been commercially developed and is in use in an organization has most likely been transitioned at least twice, *between* communities respectively concerned with research and development (R & D), new product development, and adoption and implementation. (In addition, the technology is transitioned as it progresses through its life cycle *within* each of these communities or businesses.) Traditionally, these communities have only limited interaction with each other. However, by looking at technology transition across

1. An earlier version of this paper appeared in: *Proceedings of 27th Annual Hawaii International Conference on System Sciences*, January 1994, Maui, HI. IEEE Computer Society Press.

2. The phrase "technology transfer" is usually preferred, except within the DoD. For the purposes of this paper, we consider "technology transfer," "technology deployment," and "technology transition" to be synonymous. In addition, we agree with Tornatzky & Fleischer [1990]: "Technology transfer, while a commonly used term, has a host of nuances, not the least of which is the image that technology is something that is physical, comes in large crates or on pallets, and gets literally moved from place to place." On this basis, they "use the more inclusive and less encumbered notion of *deployment*"; we prefer "technology transition" (p.118; italics Tornatzky & Fleischer).

these boundaries, we can understand the complete birth-to-retirement process; we can build a shared vocabulary and eliminate the need for reinvention; and we can draw from a range of transition experience, lessons learned, and research. Common understanding is particularly important to the development of practical approaches to technology transition at the Software Engineering Institute (SEI).

With this understanding, in the longer term, comes the possibility of concurrent technology transition, where each community acts in concert with the others. There is already some movement in this direction. For example, technology developed with early commercial partners draws from the results of research prototypes that are alpha tested in an end user organization.[3] Increasingly, end user organizations act as co-developers, not just test sites.[4]

In this report, we present each community's perspective and its related life cycle, in an effort to begin the dialogue that will make concurrent technology transition a reality. We describe the framework and vocabulary being developed at the SEI and the application of key strategies that attend to "push" and "pull" dimensions of transition.

---

3. The National Center for Manufacturing Sciences (NCMS) requires the commitment of a commercial partner prior to project approval and funding. Ted Olson, NCMS, at Council of Consortia Technology Transfer Committee meeting, November 1991, Pittsburgh, PA.

4. There are a number of examples of this, such as: the joint application development (JAD) methodology for information systems development; software systems prototyping as suggested by Barry Boehm's spiral model; and Dorothy Leonard-Barton's description of expert system development at Digital Equipment Corporation.

2 Terminology for Use in Software Technology Transition

One obstacle to understanding software technology transition is the varying terminology used to describe aspects of the process [Downs & Mohr 1976]. Since the research base for technology transition is interdisciplinary, new researchers and practitioners in the field are often unaware of existing work, and so they invent new terms. Lack of consistent terminology means that experience cannot readily be compared across contexts. Defining terms and understanding the relationships between those terms is critical. We find it particularly useful to distinguish between terms such as *theory, strategy, model,* and *mechanism* in the following manner.

A transition ➡**theory**[5] provides important underpinnings of empirically based research. The best known of these—Rogers' "diffusion of innovations" theory—uses an s-curve to represent how a population of individuals adopts an innovation over time, and a normal curve to illustrate the distribution of adopting populations, including the categories of innovators: early adopters, early majority, late majority, and "laggards" [Rogers 1983].

Transition ➡**strategy** is often confused with transition *mechanism*. The classic transition strategy of the Cooperative Extension Service of the U.S. Department of Agriculture is one of staged translation and application of information and technology, beginning with academics in land grant universities and ending with the farmer or homeowner. This strategy employs many different mechanisms, including: newsletters, reference material, training, boundary spanners in the form of cooperative extension agents, and joint funding at the local, state, and federal levels.

A transition ➡**mechanism** is the means by which information, procedures, or skills are communicated. These fall into two categories. The first category is information dissemination; the objective here is to carry information. Examples range from marketing brochures and advertising to engineering handbooks. The second category is technology implementation, where the objective is to alter attitudes or behavior, including new skill sets. Examples here include training courses, revised reward systems, and policy change.

A technology transition ➡**model** represents a set of key aspects of technology transition. Typically, these models offer starting points for discussion of strategic planning; they can also be used as heuristics for selecting mechanisms to realize a chosen strategy, or for informally testing the viability and the value of the strategy itself. The model in Figure 2-1 (page 4), adapted from Adler and Shenhar [1990], describes the relative breadth of impact of technologies.

5. When we introduce a new term in the text that appears in the glossary (Appendix A, page 27), we print it in ➡**bold** typeface and precede it with an arrow (➡).

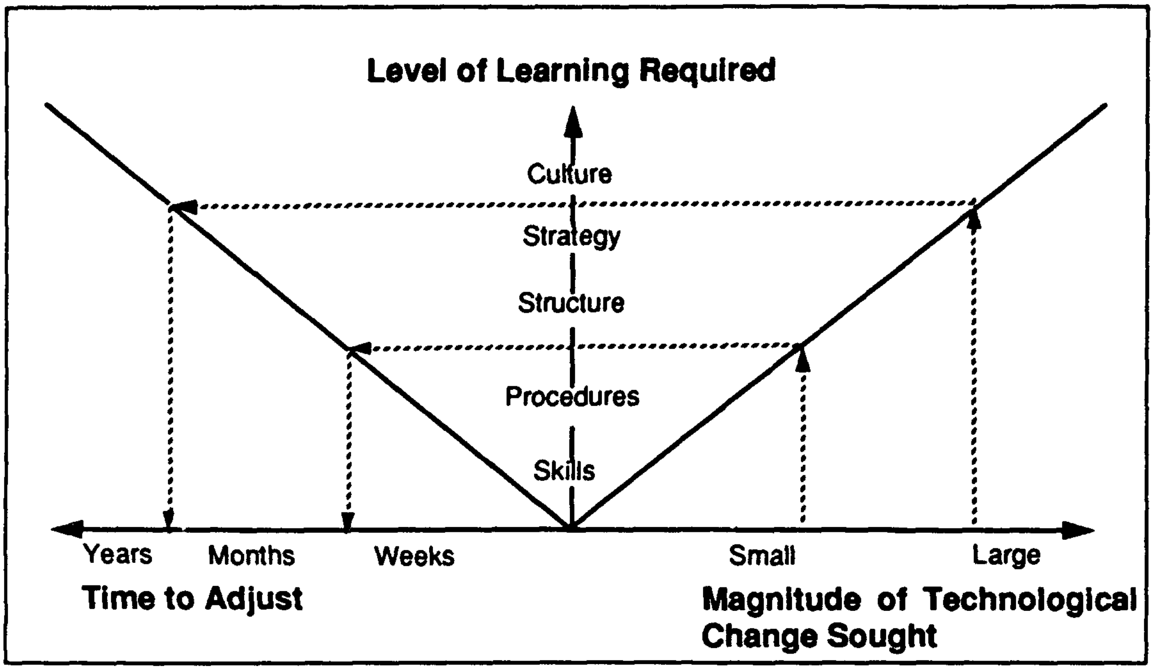


Figure 2-1: Dimensions of Change

According to Adler and Shenhar, adopting a technology that will change skills and procedures can be accomplished within the space of weeks; in contrast, adopting a technology involving a change in either structure or strategy requires months of planning and implementation. This model on dimensions of change can be used to informally assess the feasibility of an effort within a given time frame. For example, based on this model, it would be foolhardy to attempt to adopt total quality management within a matter of months.

Finally, a transition →**process** is a set of related steps that addresses a particular →**transition situation**. Three informal examples of transition processes are Bouldin's on the introduction of computer-aided software engineering (CASE) tools in organizations [Bouldin 1989], Grady and Caswell's on the introduction of software metrics at Hewlett Packard [Grady & Caswell 1987], and Strauss and Ebenau's on introducing software inspections [Strauss & Ebenau 1993]. Problem-solving strategies can be used to uncover and systematize the dynamics of transition processes, including the goals and constraints that are operating in each situation [Fowler & Levine 1992].

3 A Framework for Understanding Software Technology Transition

In addition to addressing issues of terminology, it is helpful to locate discussion—and our technology transition problems—in the context of a conceptual framework (see Figure 3-1). In the course of investigating technology transition as understood by disciplines such as management science, political science, communication, and economics, we have discovered three major perspectives or life cycles.

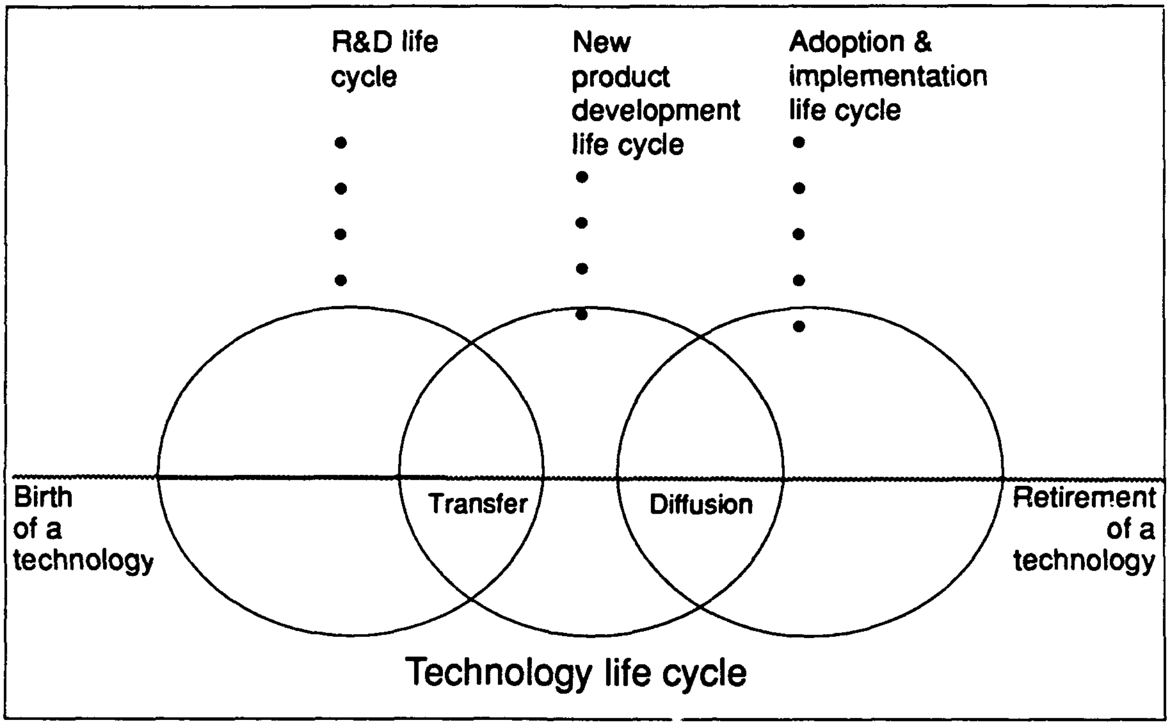


Figure 3-1: A Conceptual Framework for Technology Transition

As already indicated, these are: R & D (including the creation of prototypes), new product development, and technology adoption and implementation. Together, these three life cycles cover technology development and transition from the inception of a technology until its retirement.[6]

Location of a software technology transition problem within the large composite life cycle is an important first step in determining the requirements of a particular transition effort (which may involve moving a technology from a research lab into advanced commercial development, or managing the process of introducing a mature technology into a software development orga-

6. We define ➔**technology maturation** to include both the development of the technology and the transition processes that are attendant to the technology. When we speak of a 20 or more year process, we are concerned with radical, and not incremental, innovations.

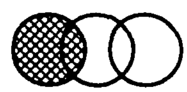
nization). In addition, understanding the processes within and between each of the three smaller life cycles and the overlap between them gives us a way of translating concurrent engineering principles for software technology transition. This opens up the possibility of reducing what is normally a 20-year process [Redwine 1984, Willis 1983]. Specific issues and approaches vary according to the nature of the particular situation.

Several clarifications about the application of the conceptual framework are in order. Typically, the process of technology transition within and between each of the life cycles is iterative, allowing for feedback and adjustment. The process is not as linear as is depicted here. Similarly, the framework outlines a comprehensive set of steps; whereas in reality, technology transition involves a subset, and not the entire set. This model is constructed to include all aspects and component parts associated with transition; it does not describe how an individual transition effort takes place, although it can serve as a heuristic for articulating a vision and related goals for a technology transition effort.

The conceptual framework can be used strategically. The interlocking life cycles function as a map allowing one to plot a course, and to consider alternate routes to reach the final destination. Leapfrogging over a phase in transition can be likened to a shortcut that may get a product to market quicker but at some cost: early access to customers may need to be traded off against fewer features. In the case of introducing a new software technology in an organization, earlier access may mean less maturity—documentation may be sketchy and training strictly informal. In each case, the associated costs and benefits must be weighed.

Our initial goal for development of the conceptual framework and vocabulary is to improve how the SEI accomplishes software technology transition. The longer term goal is to provide guidance to the broader software community. Potentially, this work can be useful to anyone responsible for the transition of software technology.

3.1 Research and Development



The focus of the *research and development life cycle* is predominantly on the changes that the technology itself goes through as it matures[7] [Botkin & Matthews 1992, Moore 1991, Redwine 1984, Tornatzky & Fleischer 1990]. Typically, this life cycle includes

- **→Concept formulation**
- Development and extension
- Enhancement and exploration (internal)
- Enhancement and exploration (external)
- (Early) popularization

7. The R & D life cycle referred to [Redwine 1984] includes "early popularization." However, our understanding of technology transition and the three interlocking life cycles argues for seeing "early popularization" as a result of the new product development life cycle.

The emphasis is primarily technology "push," and the perspective is that of the researcher or technology developer. From this perspective, transition means orchestrating the development of the technology by "moving" it systematically through stages of development[8] until it is finally incorporated into a prototype product. This view provides us with the concept of *reinvention* [Tornatzky & Fleischer 1990]. As the technology moves through the life cycle, those participating in each phase add value [Botkin & Matthews 1992] by modifying or adapting the technology, thus making it more and more robust and bringing it closer to meeting end-user needs.

The goal is relevant and usable technology. However, often the technology or prototype is not sufficiently mature for immediate integration with other technologies or for commercialization. Integration may depend on compatibility with other standards: part of the maturation of the technology may require either developing missing standards or revising already existing and incompatible standards. This points to a major disjunction in software technology transition and partially explains why the process takes so long and why the practice of employing concurrent engineering principles in technology transition efforts is so attractive.

3.2 New Product Development

Often, technology may be targeted for use in a *single application*, such as a new avionics system; in this case, the product development process is an extension of the technology development process. For example, a research prototype can be extended and incorporated directly into the new system by development engineers without a commercial vendor as an intermediary. With targeted or →**focused markets**, generic technology or product components are often assembled or adapted for a limited class of customers. For example, computer-aided design (CAD) systems allow for partial customization, but some training, commercial documentation, and customer support is necessary. Finally, if the technology is to become part of a software development technique or tool and it is to be used ultimately by the *mass market*, a different approach to transition is required. The customer base may be tens of thousands of software engineers; and development of the technology into a product (and related marketing and support activities) is necessary for broad dissemination and use. Lessons and techniques from the new product development process are needed. These are discussed below.

The *new product development process* has its own life cycle, and is used by commercial enterprises [Moore 1991, Souder 1987]. This process focuses on the embodiment of ideas in products targeted for focused and mass markets. It represents a major transformation of a technology into a form accessible to a broad group of users, as with, for example, word processing technology into software tools such as Microsoft Word or Frame Technology Corporation's FrameMaker. Support, online help, and training materials, etc., are adjunct to the base technology/product but integral to the →**whole product** concept [Moore 1991]. Similarly, consulting and support services are secondary albeit important adjunct products.

---

[8] A good example of this is the U.S. Department of Defense funding process with different types of funding for different stages of technology maturation.

The new product development process typically includes

- Generating new product ideas
- Screening the ideas
- Testing product concepts
- Business planning (e.g., sales forecasting)
- Development
- Prototyping
- Test marketing and developing pricing strategies
- Product launch

3.3 Adoption and Implementation

The third and final view, the *adoption and implementation life cycle*, addresses the case of installing technology in an organization. This view focuses on organizational matters: the users of the new technology, the management structure needed to facilitate the adoption of the technology, and the changes in work habits and preferences that a new technology may cause [Fichman & Kemerer 1993, Fowler & Levine 1992, Fowler & Maher 1992, Leonard-Barton 1988b, Zmud & Apple 1992]. This life cycle addresses the provision of knowledge and skills to the end users of a new technology. It also includes building or modifying communication channels, reward structures to encourage new technology use, support systems during the adoption of the technology, and consistent management sponsorship during adoption. The adoption and implementation life cycle links the introduction of new technology with the organizational changes needed to support its longevity in the new context [Walton 1989].

The introduction of a technology that is new to an organization typically includes the following phases[9]:

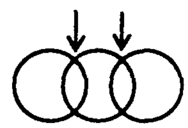
- Needs assessment
- Selection of candidate products
- Evaluation of candidate products
- Introduction of selected product to management, end users (including pilot use)
- Gathering of feedback from management and users
- Implementation planning
- Implementation
- Product maintenance
- End user support

---

9. This list is adapted from [Bouldin 1989].

Activities in this life cycle must attend to the culture of the organization in which the technology is being implemented. An innovative organizational culture can offer systemic support to the introduction, installation, adoption, and implementation of new software technology [Kanter 1983]. A more traditional, hierarchical organization may be better suited to initiating an organization-wide change effort [Nord & Tucker 1987]. The internal change agent—the project manager of the implementation effort—must be aware of the culture of his or her organization as well as of typical interactions among the organization's managerial, strategic, human, technological, and structural subsystems [Morgan 1986].

The three perspectives of technology transition outlined here are complementary, not contradictory. In the R & D life cycle, work focuses on keeping the pipeline full of viable and useful technologies and gives us a model of reinvention. Commercial enterprise adds value and delivers technologies to markets through the new product development process. Competent change agents and a systematic approach to implementation are essential to even the most complete product package.



3.4 Transactions Between Intermediaries

Only part of the story is told through the three life cycles. Another part is revealed through the examination of the intersections between R & D and new product development and between new product development and adoption and implementation. The nature of the transactions at each intersection is significantly different.

Most transactions, between technology producers and technology consumers, are facilitated by a broker of some kind. Generally the broker is an advocate on behalf of the producer, not the consumer. A complementary role—the →**receptor function**—acts on behalf of the consumer and adds symmetry to this communication [Fowler 1990] (Figure 3-2).

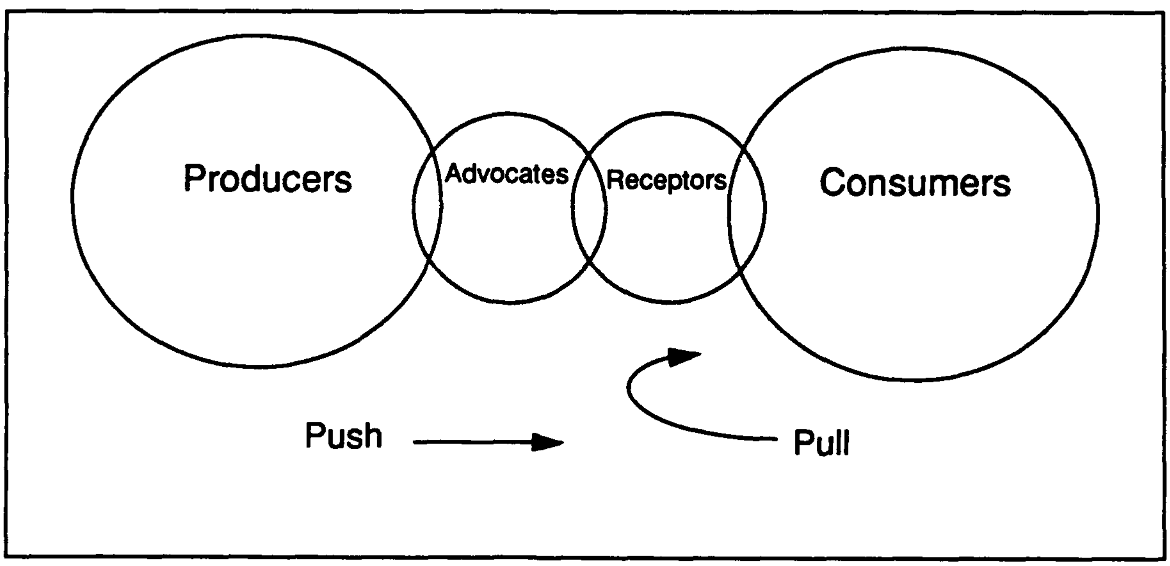


Figure 3-2: Transactions Between Intermediaries

The concept of →**mutual adaptation** [Leonard-Barton 1988a] offers a model for how the receptor function, as a recipient of technology for downstream organizations, markets, or communities, is creative and adds value. Mutual adaptation means that adjustments are made to both the receiving organization and the technology in order for adoption and use to occur. Receptors act as co-inventors or at least co-developers, translating the technology on behalf of the context they represent.

Receptor functions must be scaled in size and power according to the magnitude of their responsibilities and tasks. In addition, successful receptor functions interact with the equivalent intermediaries on the other side—with technology advocates, including marketers, entrepreneurs, and technology transfer agents in universities and government laboratories (Figure 3-3).

The transactions performed by intermediaries [Przybylinski, Fowler, & Maher 1991] can be classified roughly as one-to-one (or one-to-several) in the intersection between R & D and new product development, and as one-to-many in the intersection between new product development and adoption and implementation. Other distinctions related to the maturity of the technology are critical. We discuss each of these intersections in turn.

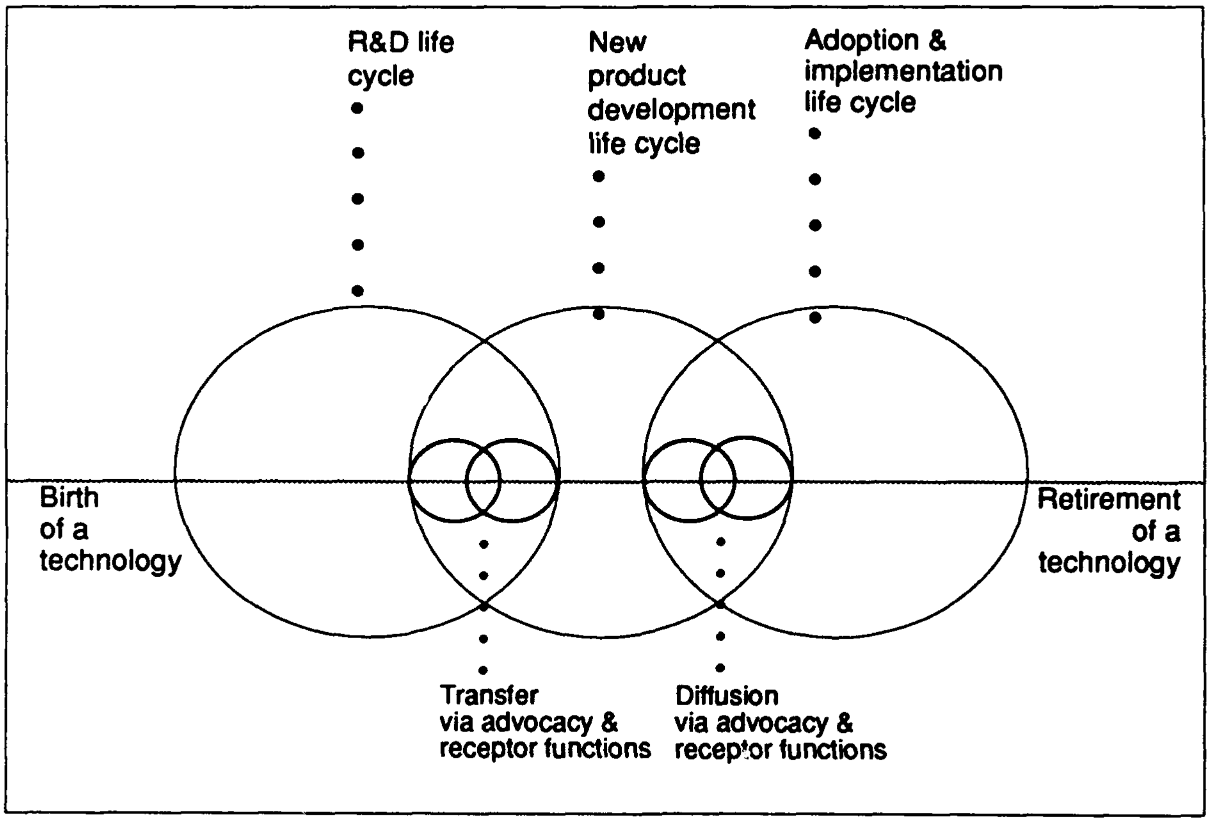
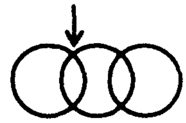


Figure 3-3: Conceptual Framework with Transactions Between Intermediaries

3.4.1 Working the Interstices



In the first intersection, *between R & D* and *new product development*, the technology itself is transitioned. At this point, the technology may not be embodied in a product. In effect, it is immature, making the process of transition labor-intensive. One-to-many information dissemination can occur through distribution of technical papers; however, this does not guarantee application of the technology or conversion into product form. Widespread information dissemination reduces the risk of missed opportunities, but identification of partners and follow-on cooperation depends upon individual efforts. In effect, while distribution of technical papers may appear to be one-to-many, the transfer of the technology relies on one-to-one interaction.

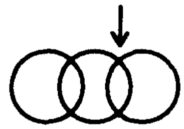
As a consequence of the immaturity of the technology, technology transition advocates in R&D tout the need for, and utility of, moving the people *with* the technology [Glynn 1990]. Transition activities are designed to allow interaction between technology inventors, developers, and recipients of custom systems or early users of mass market tools. Technical interchange meetings, seminars, colloquia, and conferences are typical at this juncture. Many organizations act as sponsors for such events, most notably professional societies, such as the Association for Computing Machinery, Institute of Electrical and Electronic Engineers (IEEE) Computer Society, Federation for Information Processing, and related special interest groups. Frequently, working groups are concerned with building an understanding of the new technology area; they may be developing foundational material, taxonomies, and frameworks.

In addition to the notion of immaturity described, two themes dominate:

1. Intellectual property rights through vehicles such as licenses, patents, trademarks, and copyrights.
2. Standards development.

Because the technology is not yet "encapsulated," there is the potential for developing many different applications and products. Product developers compete for access to the technology during this period, jockeying for advantageous positions in emerging markets. Related efforts and activities at this intersection include cooperative research and development agreements (CRDAs), strategic partnerships for co-development, licensing agreements, and consortia activity.

Efforts at creating ➞**standards** are not to be overlooked. During early productization, competing products based on the same technology but requiring different hardware and software platforms emerge. Sometimes a product is so strong that it dominates and becomes the de facto standard [Fichman & Kemerer 1993]. In other instances—for example, the computer-aided software engineering (CASE) marketplace—the situation is not so clear cut. Standards work for open systems is motivated by the increasing demand for interconnectivity of CASE products within organizations, and operating systems within networks. This type of standards work takes several years and significant pressure from the marketplace to begin.



In the second intersection, between *new product development* and *adoption and implementation*, different themes emerge. Here, the whole product concept [Moore 1991] is critical: the core technology exists (perhaps embodied in software or in a well-defined process) as well as attendant materials such as training courses, customer support, and user manuals. Ideally, the whole product incorporates the level of detail and support that users need, without the labor-intensive, one-to-one support that characterizes activity at the previous intersection. The focus is on diffusion (as opposed to transfer) and on highly leveraged, one-to-many transactions.

The development of "pull" or user capability is also evident at this intersection. User groups are emerging or underway, actively sharing the experiences that they have with the product. Within organizations, working groups may be organized to track developments related to a product type (e.g., CASE).

The nature of the dissemination process for products is also different. The product developer may not supply the whole product, but he or she may work with value-added resellers. Similarly, the developer may not be the sole distributor, allowing for partnerships where one organization develops the product and another organization, with established distribution channels, provides access to the market. Consultants, able to accumulate expertise across a range of product applications, may compensate for product inadequacies with direct support to users.

Standards issues, relevant at the first intersection, continue to play a role. While in the R & D and new product development arena, *standards making* was the focus; here, the emphasis is on *standards application*. As a technology area—that is, a set of related products that represent a maturing technology—evolves, the early majority and the late majority populations dominate the market [Moore 1991]. Stability of product, often based in standards, becomes more important. In addition, if standards development and revision have been neglected during R & D and new product development, attention to these issues is now essential. If de facto standards have not emerged, user communities will exert pressure to create them.

The SEI works both of these intersections: improving the state of the practice in software engineering requires involvement at multiple levels spanning the three life cycles described here. SEI strategy emphasizes leveraging the existing infrastructure in each of the three arenas. We describe particular strategies in the next section.

4 Applying the Framework: Technology Transition at the SEI

The development of the conceptual framework for software technology transition has grown out of the need to serve the mission of the SEI, a federally funded research and development center (FFRDC), located at Carnegie Mellon University. Its mission—to advance the state of software engineering practice—requires a transition strategy that will meet the needs of SEI customers. These customers include: U.S. Department of Defense (DoD) contractors; military services; government agencies such as National Aeronautics and Space Administration (NASA), National Institute of Standards and Technology (NIST), and the Federal Aviation Administration (FAA); large and small software vendors who serve these organizations; large corporations such as AT&T and Hewlett Packard whose businesses are software-intensive; and academic and industrial providers of education and training. As national security concerns broaden to include competitiveness issues, the constituencies served by the SEI are also extended.

The goal of the Institute is to help customers make lasting improvements in their ability to acquire, develop, and maintain software-dependent systems, and in their ability to educate people to perform these activities effectively. Currently, the SEI is constrained by charter to 250 members of the technical staff, while the population addressed by the SEI transition effort includes hundreds of thousands of technical professionals, their managers, and the academic and continuing education community. Thus, satisfying this goal depends upon applying a thorough understanding of technology transition—the terminology and conceptual framework we have just described—to derive and employ strategies that will realize the SEI mission.

Three key strategies provide the SEI with this leverage: information dissemination and outreach activities, partnerships, and infrastructure development. Through the use of these strategies, the SEI addresses transition within each arena of the conceptual framework and works the intersections between them (Figure 3-3, page 10). Below, we discuss each strategy and the role that it plays in expediting the transition of technology. We also describe mechanisms that serve each strategy.

4.1 Information Dissemination and Outreach Activities

The dissemination of information is prerequisite to use. Information dissemination enables contact, awareness, and understanding of the possible application of a technology [Conner & Patterson 1982]. Dissemination of information takes many forms and is a strategy common across all three life cycles. At the SEI, information is disseminated in a variety of conventional but effective ways, including *print, video,* and a range of people-to-people *events*. Depending upon where the technology/product is in its development, appropriate mechanisms and distribution channels are used to impart information.

In the R & D life cycle, information is disseminated through SEI technical and special reports[10] and external publications. In the new product development life cycle, information dissemination often takes the form of product documentation and tutorial material. In addition, the SEI produces a *portfolio* of its products and services.

A guidebook such as the *Software Engineering Process Group Guide* [Fowler & Rifkin 1990] is one example of how information pertaining to implementation is conveyed. Engineering handbooks, such as *A Practitioner's Handbook for Real-Time Analysis* [Klein et al 1993] are also important print-based dissemination mechanisms attending to this area.

The SEI Education Program uses video extensively: for software engineering courses delivered through the National Technological University and for a lecture series on special topics, where each tape is sold separately. A Products and Services Planning function and a related communications group advise on appropriate media for particular transition efforts.

Direct contact with customers occurs throughout the birth-to-retirement cycle. During R & D, this takes place through small technical interchange meetings, advisory board meetings, and workshops of 50 to 100 people. As the technologies that the SEI is working on mature, contact occurs in larger forums including an annual symposium and the annual Software Engineering Education Conference. External events, with a strong practitioner flavor, such as the Software Technology Conference sponsored by the Air Force, and the Tri-Ada conference also offer opportunities for outreach. Continuing education and academic courses focus on the dissemination of skills and knowledge about more mature technologies. Each SEI technology project, with guidance from business development and marketing functions, targets the appropriate opportunities for contact with their constituencies.

Day-to-day customer service is managed through a telephone information line, Internet email, and FAX. A *subscriber program* allows individuals to stay informed about SEI activities through mailings about SEI events, course offerings, work in progress, new products and new initiatives, plus a quarterly magazine *(Bridge)*, and the annual *Technical Review.*

4.2 Partnerships

Transition requires cooperation throughout the technology development life cycle. Partnerships—relationships that bring different knowledge, resources, and perspectives to bear on complex problems—are a primary form of collaboration. Informal partnerships can occur *within* each of the three life cycles: for example, interdisciplinary teams or individuals from different units within an organization working in collaboration. Often, such partnerships can be established and disbanded fairly easily. Formal (legally binding) partnerships are more common at the intersections *between* life cycles.These agreements and negotiations are necessary when the parties involved are working across different domains, and when there is variance in the communities' values and work styles.

10. Although they are print-based, most SEI reports are available through anonymous ftp over the Internet.

The SEI uses three complementary forms of partnership to extend its impact: *technical* and *strategic partnerships* at the intersection between R & D and new product development; and associations with →**distribution partners** or third-party vendors at the intersection between new product development and adoption and implementation. The technical and strategic partnership programs are intended to create well-defined and well-managed relationships with the community of industry customers. Through these partnerships, the SEI has access to an industry constituency that can

- Provide input to the SEI technical program.
- Advance development and maturity of SEI technology, products, or services.
- Provide in-kind and direct funding resources.

These partnerships are semi-formal in that they are documented but not legally binding. They are described as follows:

→**Technical partnerships** are formed for a fixed duration and involve defined areas of collaboration with a single SEI technical activity. These relationships are initiated by mutual agreement and are negotiated between the project and the potential partner with the intent of exchanging value of mutual benefit. Current examples include co-development of software tools and engineering handbooks, and also joint technology development.

→**Strategic partnerships** are long-term, collaborative relationships between the SEI and selected industry partners. These associations, as for example with Hewlett Packard, are characterized by mutual statements of strategic intent and goals. The strategic relationship is realized by executing multiple technical partnership agreements, as described above. Benefits for strategic partners include broader and more immediate access to, and influence within, the SEI. Benefits to the SEI include access to industry best practices in software engineering and early feedback on SEI work in technology development and transition.

Other partnerships involving co-development, third-party vendors, and distribution, are designed to leverage existing delivery systems. These transition partners are active in the market segment to which they intend to target the SEI product and they have sufficient resources and motivation to promote and support their SEI-related effort. In document dissemination, partners include: the National Technical Information Service (NTIS), Defense Technical Information Center (DTIC) and Research Access, Inc. (RAI).

In addition, software process assessment (SPA) associates are licensed by the SEI to evaluate software organizations; the National Technological University (NTU) delivers SEI-developed courses; Springer-Verlag publishes SEI conference proceedings; and Kluwer Academic Publishers released *A Practitioner's Handbook for Real-Time Analysis* [Klein et al 1993] in the summer of 1993.

Funding mechanisms such as direct government grants for FFRDCs allow for timely partnerships—for co-development early in SEI work and for subsequent prototyping of the technology and transition methods directly with customers. In addition, the SEI is experimenting with the use of cooperative research and development agreements (CRDAs) and cost recovery as mechanisms for strengthening its working relationships with industry. Partnering is an effective strategy because of its potential for leveraging existing infrastructure.

4.3 Infrastructure Development

We define →**transition infrastructure** as the system of technology developers, brokers (advocates and receptors), and delivery mechanisms that interact with each other and with the marketplace to move technology from birth through development and into widespread use. While this system is complex, there are clear leverage points: in education, in the standards arena, in the development of maturity models, and in the development of change agents and related "pull" capability. The SEI attempts to take advantage of all of these.

The *educational system* of the United States (higher education, education and training vendors, in-house educators, government schools, and satellites) provides the SEI with a means of gaining access to engineering practitioners and their managers, and to those who teach them. The system provides immediate training to practitioners, but viewed more broadly, it is an environment for preparing the next generation of practitioners, managers, and educators. SEI model curricula for graduate and undergraduate programs helps universities and in-house educators to bootstrap software engineering education and training offerings. NTU, described above, and the National Defense University, which serves a large government audience, are examples of how the SEI reaches large numbers of people through the educational system.

While standards often take years to reach official approval with multiple intermediate drafts circulated for comments and voting by the technical community, the SEI recognizes the importance of contributing to precompetitive consensus-building and standards efforts. Standards efforts represent a high-leverage activity for improving computing and software engineering since they are community efforts, developed and distributed by organizations such as the Institute of Electrical and Electronic Engineers (IEEE), American National Standards Institute(-ANSI), and International Organization for Standardization (ISO). Standards can also be an effective means of reducing barriers to technology adoption. For example, standards that might potentially block technology may be modified to permit its adoption or, ideally, support it. In the case of rate monotonic analysis (RMA), SEI project members worked to interpret the Ada standard to allow an operating system to conform to the standard while still supporting RMA scheduling.

Standards are an excellent way to raise people's awareness of a technology: once the technology is embodied in standards, people cite the standard as justification for using the technology. This tips the scale toward use of the technology, in effect creating "pull."[11] The SEI participates in and, in some cases, leads these efforts (for example, POSIX 1003.21 for the real-time distributed systems communication domain), thereby insuring that client needs are met while contributing to the larger aims of the Institute.

11. John Goodenough, personal communication, June 1993.

The *Resident Affiliate Program* provides the opportunity for experienced technical personnel from government, industry, and academic organizations to participate in SEI projects for a period ranging from 6 to 24 months. Thirty-four U.S. organizations and 6 foreign organizations have sent affiliates. Resident affiliates contribute both as software engineers and as application-domain experts during their stay at the SEI, and they make up about 10% of the technical staff. The immediate payoff for the sponsoring organizations is: participation in technical activities that might not be possible in their own organizations; access to SEI people, projects, and other resident affiliates; and access to early technical results. The resident affiliate benefits from working in a different technical context, from participating in the many workshops and other activities at the SEI and in the larger Carnegie Mellon community, and from interacting with colleagues from different professional, technical, and organizational backgrounds. The SEI also benefits because it obtains experience, expertise, and additional insight into the software engineering community.

Resident affiliates are change agents, both during their tenure at the SEI, and especially upon their return to their home organization. As a result, they receive training for this role through SEI courses such as Managing Technological Change and Consulting Skills Workshop.

From time to time, the SEI identifies the need for a *customer advisory board* or *working group* to provide guidance on current activities and future plans and to perform technical reviews. This enables the SEI to engage communities early in the process of technology development, which in turn expedites buy-in. Members are selected through a screening process intended to populate the board or group with a mix of technical professionals who can help to satisfy SEI technical objectives.

Recognizing the potential for a user community, the SEI developed the *software engineering process group* (SEPG) concept (based on field activity) and furthered the concept with a guidebook [Fowler 1990] and an annual national meeting. Each SEPG spearheads the tracking, introduction, and implementation of new software engineering technologies within its home organization. Typically, the objective of an SEPG is to advance the software capability of its organization through software process improvement and related technology transition activities. Large organizations such as Texas Instruments and IBM may have corporate SEPGs as well as SEPGs distributed throughout product divisions.

In September 1992, the SEI agreed to serve as coordinator for the emerging *Software Process Improvement Network* (SPIN). This network provides significant leverage for transition. The SEI interacts with SPINs, each member of which represents a number of SEPGs, each of which, in turn, acts as a receptor and change agent within its home organization. Figure 4-1 on page 18 illustrates the relationship between the SEI, SPINs and SEPGs. SPIN organizations are active in Washington, D.C.; Irvine, California (serving Orange and Los Angeles counties); Dallas; Austin; Boston; and Seattle. New SPINs are emerging in Boulder, St. Louis, and Hoboken (serving northern New Jersey).

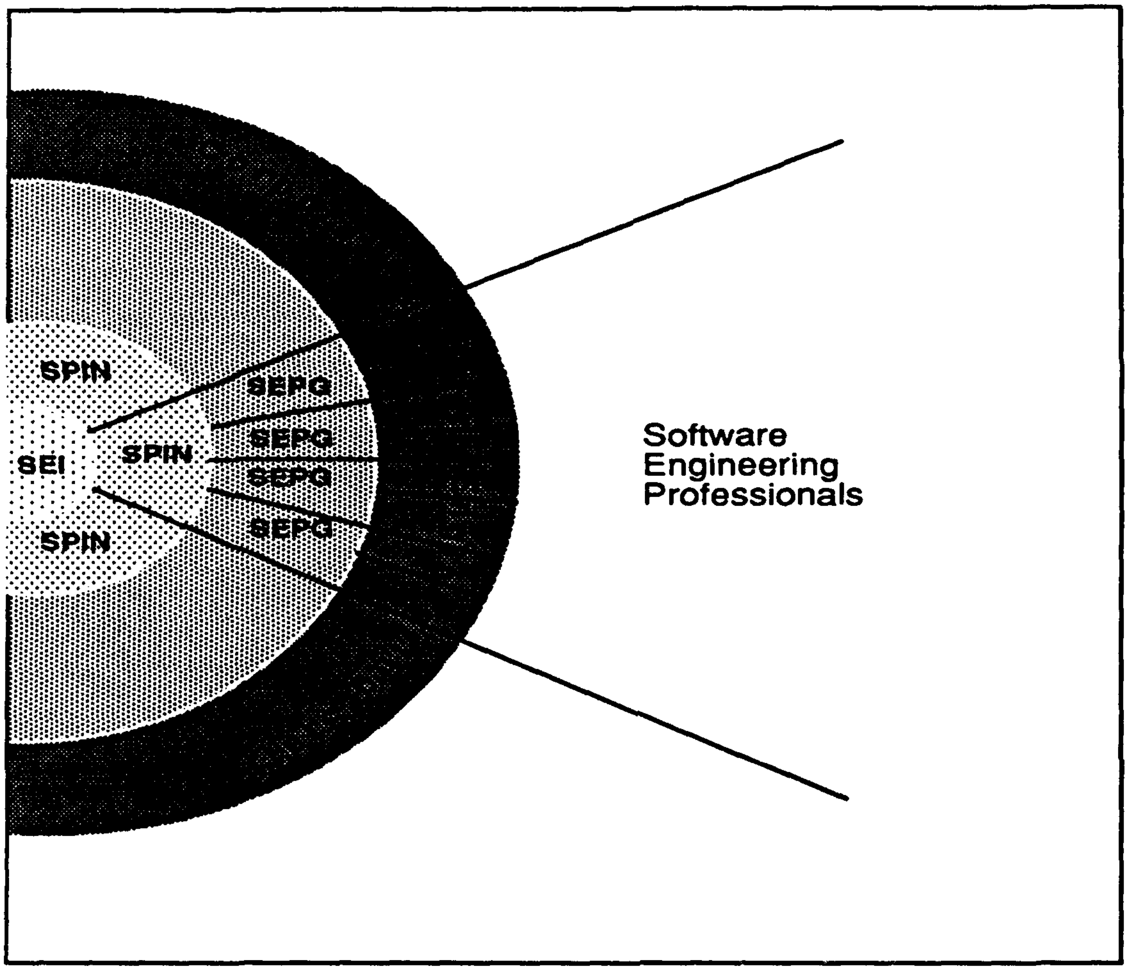


Figure 4-1: Effect of Software Process Improvement Network[12]

The SEI is also building infrastructure through the development of →**maturity models.** A maturity model is a framework for characterizing the evolution of a system or process from a less organized, less effective state to a highly structured state. Maturity models can support a paradigm shift: they can influence the state of the practice of software engineering.

The capability maturity model for software (CMM) is based on the elements of an effective software process [Paulk 1991]. The CMM describes an evolutionary improvement path from an ad hoc, chaotic process to a mature disciplined process [Humphrey 1988]. When followed, key practices described as an adjunct to the CMM improve the ability of organizations to meet goals for cost, schedule, functionality, and product quality. The CMM establishes a scale against which it is possible to measure, in a repeatable way, the maturity of an organization's

12. This figure originally appeared in the SEI *1994 1 & 5 Year Plan.*

software process. This scale can also be used by an organization to plan improvements to its software process through the introduction of the key practices (for example, project management, peer reviews, and configuration management) and the technology that supports them.

The primary focus of the CMM is on management aspects of the software engineering process. Currently, the notion of technical maturity for product engineering is also being explored. Models related to product engineering and to science and technology may evolve out of already existing practices (associated with product engineering) in the CMM. The CMM would then become a family of related maturity models, each focusing on providing guidance for maturing different aspects of software engineering.

5 Conclusion

While the development of this conceptual framework and attendant terminology has grown out of the need to serve the heavily transition-oriented mission of the SEI, we believe its potential application is broader.

If we see mature technology as readily used or ready-to-use technology, then we must define technology maturation as a combination of technology development and technology transition. Conventionally, software technology transition begins *after* development, too late to influence the shape of the technology and make it responsive to user needs. The old paradigm precludes meaningful interaction between developers and users.

Leonard-Barton describes a process of "mutual adaptation" [Leonard-Barton 1988a] in which both organization and technology must be adjusted for transition to be considered complete. This process is based largely on research into the transition of complex software innovations such as expert systems and manufacturing systems. The research suggests that those doing the planning for technology transition should anticipate changes not just to the technology (however mature) but to organizational processes as well. Leonard-Barton discusses the idea of reinvention and considers how organizations as well as technologies change as they modify and add value to maturing technologies.

We argue that the idea of mutual adaptation can be extended by the principles of concurrent engineering (see Figure 5-1), and thus lead to an improvement in the effort and time it takes to accomplish software technology transition.

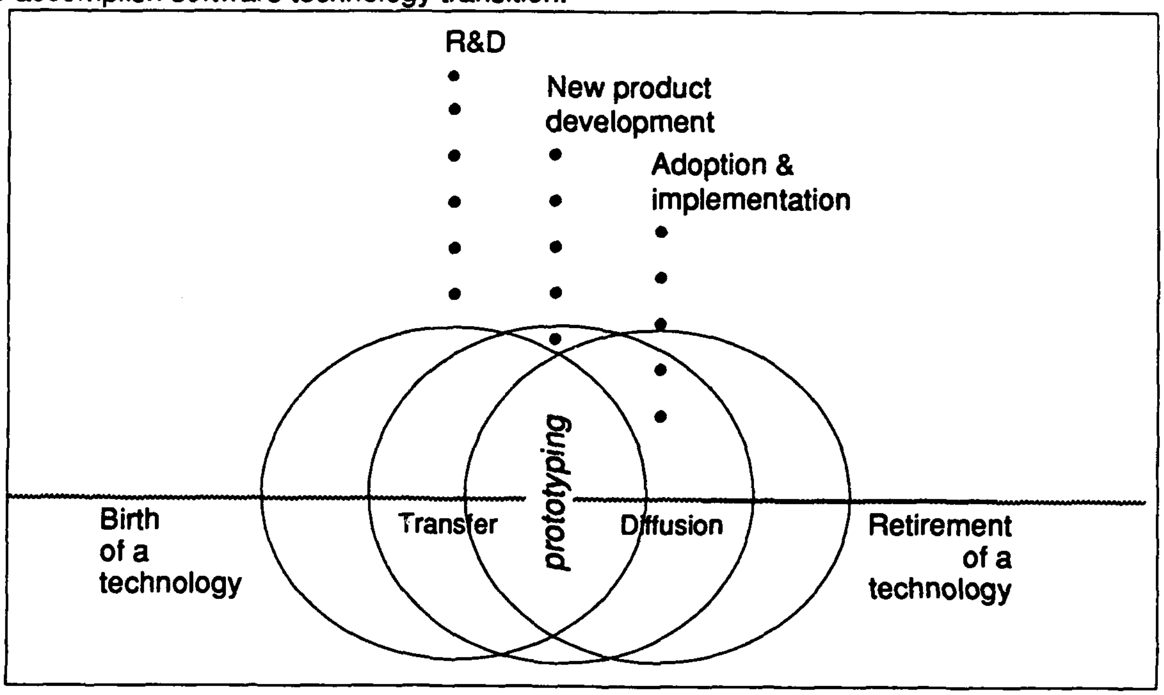


Figure 5-1: Concurrent Technology Transition

We suggest that this approach will also lead to improved products. Use of these principles stresses that research, new product development, and implementation issues and tasks are worked concurrently. When this occurs, information about how well the technology is working in context—that is, not just information about how the technology itself is developing—can be obtained immediately. Concurrent technology transition should significantly reduce the length of time required for creation and diffusion of a mature technology.

Acknowledgments

Many individuals at the SEI have contributed directly and indirectly to the ideas discussed here: Mario Barbacci of the Product Attribute Engineering Program, Mike DeRiso of Products and Services Planning, Ken McNulty of SEI Services, Tom Ralya of Products and Services Planning, and Albert Soule of Industry Operations.

We thank Pam Hughes for her help with preparing the manuscript and Bill Pollak for editorial assistance. Special thanks are due to Joe Morin for his insights on technology transition and the use of conceptual models.

References

Adler, P.S. & Shenhar, A. (1990). Adapting your technological base: The organizational challenge. *Sloan Management Review 32*(1), 25-37.

Botkin, J. & Matthews, J. (1992). *Winning combinations: The coming wave of entrepreneurial partnerships between large and small companies.* New York: John Wiley & Sons, Inc.

Bouldin, B. (1989). *Agents of change: Managing the introduction of automated tools.* Englewood Cliffs, N.J.: Yourdon Press.

Conner, D.R. & Patterson, R.W. (1982). Building commitment to organizational change. *Training and Development Journal,* April 1982, 18-30.

Deal, T.E. (1991). Developing a quality culture. In R. H. Kilmann (Ed.), *Making Organizations Competitive* (pp.156-175). San Francisco, Calif.: Jossey-Bass.

Dean, J.J. (1987). *Deciding to Innovate: How firms justify advanced technology.* Cambridge, Mass.: Ballinger Publishing Company.

Downs, G.W., & Mohr, L.B. (1976). Conceptual issues in the study of innovation. *Administrative Science Quarterly 21,* 700-714.

Fagan, M. (1976). Design and code inspections to reduce errors in program development. *IBM Systems Journal 15*(3), 182-211.

Fichman, R.G. & Kemerer, C. (1993). Adoption of software engineering innovations: The case of object orientation. *Sloan Management Review,* Winter 1993, 7-22.

Fowler, P. & Levine, L. (1992). Toward a problem solving approach to software technology transition. In J. Van Leeuwen (Ed.), *Proceedings of the IFIP 12th World Computer Congress, vol. 1* (pp. 57-64), Madrid, Spain. The Netherlands: North Holland, Elsevier Science Publishers.

Fowler, P. & Maher, J. (1992). Foundations for systematic software technology transition. *Software Engineering Institute Technical Review '92,* 1-32.

Fowler, P. & Rifkin, S. (1990). *Software Engineering Process Group Guide* (Technical Report CMU/SEI-90-TR-24; ESD-90-TR--225, ADA235784). Pittsburgh, Pa.: Software Engineering Institute.

Fowler, P. (1990). Technology transfer as collaboration: The receptor group. *Proceedings of the 12th International Conference on Software Engineering* (pp. 332-333), Nice, France. U.S.: IEEE Computer Society Press.

Glynn, G. (1990). Semi-formal process model for technology transfer. *Proceedings of the 12th International Conference on Software Engineering* (pp.334-335). Nice, France, U.S.: IEEE Computer Society Press.

Grady, R.B. & Caswell, D.L. (1987). *Software metrics: Establishing a company-wide program.* Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Humphrey, W.S. (1988). Characterizing the software process, *IEEE Software 5*, 73-79.

Kanter, R.M. (1983). *The change masters.* New York: Simon and Shuster.

Klein M. et al. (1993). *A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems.* Boston, Mass.: Kluwer Academic Publishers.

Leonard-Barton, D. (1988a). Implementation as mutual adaptation of technology and organization. *Research Policy 17*, 5 (Oct. 1988), 102-110.

Leonard-Barton, D. (1988b). Implementation characteristics of organizational innovations. *Communication Research 15*, 603-631.

Moore, G. (1991). *Crossing the chasm: Marketing and selling technology products to mainstream customers.* HarperBusiness.

Morgan, G. (1986). *Images of organization.* Beverly Hills, Calif.: Sage Publications.

Nord, W.R. & Tucker, S. (1987). *Implementing routine and radical innovations.* Lexington, Mass.: D.C. Heath.

Paulk, M. et al. (1991). *Capability Maturity Model for Software.* (Technical Report CMU/SEI-91-TR-24; ESD-91-TR-24, ADA240603). Pittsburgh, Pa.: Software Engineering Institute.

Pollak, W. (1993). Standards: Leverage for new technologies. Pittsburgh, Pa.: Carnegie Mellon University, Software Engineering Institute, *Bridge* (October), 1-7.

Przybylinski, S., Fowler, P. & Maher, J. (1991, May). *Software technology transition.* Tutorial presented at the 13th International Conference on Software Engineering, Austin, Texas.

Redwine, S.T., Becker, L.G., Marmor-Squires, A.B., Martin, R.J., Nash, S.H., & Riddle, W.E. (1984). *DoD related software technology requirements, practices, and prospects for the future.* (Technical Report IDA Paper P-1788). Alexandria, Va.: Institute for Defense Analysis.

Roberts, E. (1991). *Entrepreneurs in high technology: Lessons from MIT and beyond.* New York: Oxford University Press.

Rogers, E.M. (1983). *Diffusion of innovation.* New York: Free Press.

Roussel, P.A., Saad, K.N., & Erickson, T.J. (1991). *Third generation R & D: Managing the link to corporate strategy*. Boston: Harvard Business School Press.

Souder, W.E. (1987). *Managing new product innovations*. New York: Lexington Books.

Souder, W.E., Nashar, A.S., & Padmanabhan, V. (1990). A guide to the best technology transfer practices. *Journal of Technology Transfer 15*, 5-16.

Strauss, S., and Ebenau, R.G. (1993). *Software Inspection Process*. New York: McGraw Hill.

Tornatzky, L.G. & Fleischer, M. (1990). *The processes of technological innovation*. Lexington, Mass.: Lexington Books.

Walton, R.E. (1989). *Up and running*. Boston: Harvard Business School Press.

Williams, F. & Gibson, D.V. (Eds.). (1990). *Technology transfer: A communication perspective*. Newbury Park, Calif.: Sage Publications.

Willis, R. (1983). Technology transfer takes 6 +/- 2 years. In R. Morton (Ed.), *Proceedings of the IEEE Computer Society Workshop on Software Engineering Technology Transfer* (pp.108-117), Miami Beach, Fla. IEEE Computer Society Press.

Zmud, R.W. & Apple, L.E. (1992). Measuring technology incorporation/infusion. *Journal of Product Innovation Management 9*, 148-155.

Appendix A Glossary

adoption	"The decision to make full use of an innovation as the best course of action available." [Rogers 1983, p. 21]
concept formulation	Redwine [1984] coins this term to include: "informal circulation of ideas," "convergence on a compatible set of ideas," and/or general publication of solutions to parts of the problem (p. 85).
distribution partner	Third-party vendor.
focused market	A market segment or niche that is tightly bounded and has a limited class of customers.
implementation	The process of putting an innovation and/or technology into use in an organization's or individual's work environment.
maturity model	A framework for characterizing the evolution of a system or process from a less organized, less effective state to a highly structured state.
mechanism	The means by which information, procedures, or skills are communicated.
model	A system of claims, data, and inferences presented as a description of an entity or state of affairs.
mutual adaptation	Situation in which both the receiving organization and the technology make adjustments in order for adoption and use to occur [Leonard-Barton 1988a].
process	A series of actions or operations leading to an end.
receptor function	The intermediary role assumed by the agent who acts on behalf of the consumer.

standards	A codification or formalization of common or recommended practices. There are "many different forms of varying degrees of formality: international standards, national standards, ad hoc standards, de facto standards, standards that result from market dominance, best practices, standards produced by consortia, and reference models." [Pollak 1993, pp. 1-2]
strategic partnership	Long-term, collaborative relationship between the SEI and selected industry partners.
strategy	A plan or method devised or employed in response to a goal.
technical partnership	Partnership formed for a fixed duration and involving defined areas of collaboration with a single SEI technical activity.
theory	The general or abstract principles of a body of fact, a science, or an art.
transition infrastructure	1. The system of technology developers, brokers (advocates and receptors), and delivery mechanisms that interact with each other and with the marketplace to move technology from birth through development and into widespread use. 2. The system of sponsors, change agents (SEPGs), and working groups within an organization that plans and expedites implementation of a technology within that organization.
technology maturation	The development of the technology and the transition processes that are attendant to the technology. A mature technology typically has pulled together a whole product.

transition situation

A profile of a transition problem that considers the following questions:

1. What is the nature of the technology being transitioned?
2. What is the state of the organization that will incorporate the new technology?
3. What is the ultimate goal for acquiring and using the technology?
4. What are the steps to reach the desired goals given the state of the organization, including its human capabilities?

whole product

A technology is mature in the sense of a commercial whole product [Moore 1991] when it incorporates the secondary products and services that majority adopters need. Such products and services include: additional hardware or software, training and support, courses, documentation, handbooks, etc.

UNLIMITED, UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-93-TR-31	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESC-TR-93-317

6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute	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) HQ ESC/ENS 5 Eglin Street Hanscom AFB, MA 01731-2116
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office	8b. OFFICE SYMBOL (if applicable) ESC/ENS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962890C0003

8c. ADDRESS (city, state, and zip code))
Carnegie Mellon University
Pittsburgh PA 15213

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO	WORK UNIT NO.
63756E	N/A	N/A	N/A

11. TITLE (Include Security Classification)
A Conceptual Framework for Software Technology Transition

12. PERSONAL AUTHOR(S)
Priscilla Fowler and Linda Levine

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month, day)	15. PAGE COUNT
Final	FROM TO	December 1993	30 pp.

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)
conceptual framework
diffusion of technology
software technology transfer
software technology transition
technology development life cycles
technology maturation

19. ABSTRACT (continue on reverse if necessary and identify by block number)
We present a conceptual framework that integrates and describes the intersections of three life cycles of software technology transition; research and development, new product development, and adoption and implementation in organizations. We then apply the framework to the technology transition experiences of the Software Engineering Institute.

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ■ SAME AS RPT ☐ DTIC USERS ■	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (include area code)	22c. OFFICE SYMBOL
Thomas R. Miller, Lt Col, USAF	(412) 268-7631	ESC/ENS (SEI)

ISTRACT — continued from page one, block 19