Understanding and Improving Technology Transfer in Software Engineering

Shari Lawrence Pfleeger
Systems/Software, Inc., Washington, DC and the University of Maryland, College Park

To appear in the *Journal of Systems and Software*, February 1999

Software engineering has come a long way since 1968, when the term was first introduced at a NATO conference. Today, software pervades almost all aspects of our lives, from controlling our appliances to transporting us around the world. However, the transfer of new software engineering techniques and tools to common practice has had much more limited success; sometimes new ideas take hold immediately, but more often a new, proven idea takes many years to become accepted as standard practice. As practitioners, we are faced regularly with technology decisions: which to keep and which to replace, and how to adapt new technologies to our particular situations so that they are most efficient and effective. As researchers, we must evaluate the effects of a technology, and build a body of evidence to help in the decision-making process. This paper summarizes the history of software engineering technology transfer and suggests ways to help both practitioners and researchers understand how to shorten the time between innovation and effective practice.

Here, we use the term “technology” to encompass a large number of things, and it is important for us to understand what “technologies” to study. For example, software engineers use a variety of techniques or methods to build and maintain software. We use the terms *method* or *technique* to mean a formal procedure for producing some result. By contrast, a “tool” is an instrument, language or automated system for accomplishing something in a better way. This better way can mean that the tool makes us more accurate, more efficient, or more productive, or that it enhances the quality of the resulting product. However, a tool is not always necessary for making something well. For example, a cooking technique can make a sauce better, not the pot or spoon used by the chef. (Pfleeger 1998)

A *procedure* is like a recipe: a combination of tools and techniques that, in concert, produce a particular product. For instance, test plans describe test procedures; they tell us which tools will be used on which data sets under which circumstances so that we can determine whether our software meets its requirements. Like a cooking style, a *paradigm* represents a particular approach or philosophy for building software. Just as we can distinguish French cooking from Chinese cooking, so too do we distinguish paradigms like object-oriented from procedural development.

In this paper, we use "technology" to mean any method, technique, tool, procedure or paradigm used in software development or maintenance. We begin by examining earlier efforts to understand software-related technology transfer. Then we discuss the process of creating, evaluating, packaging and diffusing technology. Next, we consider each of these four activities in more detail, to determine how each contributes to the success of the overall transfer. Finally, we discuss areas ripe for further investigation.

Previous investigations

One of the first studies of software technology transfer was reported at the eighth International Conference on Software Engineering. There, Redwine and Riddle (1985) described their findings in investigating the pace at which software technology matures and is accepted into practice. They gathered case studies related to many concepts and technologies that were initially developed in the 1960s and 1970s, including:

- Major technology areas
  - knowledge-based systems
  - software engineering principles
  - formal verification
  - compiler construction
  - metrics
- Technology concepts
  - abstract data types
  - structured programming
- Methodology technology
  - software creation and evolution methodologies
  - software cost reduction
  - software development and acquisition standards
  - US Department of Defense software development standard STD-SDS
  - US Air Force regulation 800-14
- Consolidated technology
  - cost models
  - automated software environments
  - Smalltalk-80
  - Software Requirements Engineering Methodology
  - Unix

Redwine and Riddle were somewhat imprecise in the way they defined "technology." By including processes, standards and products, their studies addressed both the specific and the general. Nevertheless, their findings are interesting, because no matter the type of "technology," the time from innovation to common practice was longer than we would like it to be. In future studies, it would be useful to define types or classes of technology in a way that makes the objects of study more easily comparable in terms of their size and effect.

Redwine and Riddle tracked the development of each technology or concept according to a six-phase model of technology maturation:

1. Basic research
2. Concept formulation
3. Development and extension
4. Enhancement and exploration (internal)
5. Enhancement and exploration (external)
6. Popularization

Popularization of a given technology was considered in two stages: propagation throughout 40% of the community, and throughout 70% of the community. This limitation to 70% makes the implicit assumption that the entire software development community may not benefit from the technology being studied. In fact, it is often the case that a new technology is suggested to be highly beneficial only to a proper subset of the entire development community. For example, reuse may make sense only for organizations that build repeated versions or variations of "like" things; those development groups that build completely new applications each time may not benefit from a suggested reuse technology. Similarly, a particular language may be well-suited only for a particular type of application, such as building user interfaces or constructing compilers; it may not be useful for every system or program. Thus, for our discussion, we assume that propagation applies to the audience for which the technology is suitable, rather than to the entire software development community. However, we acknowledge that determining the audience for a particular technology is not always an easy task.

Adoption rate

It is important to know how long it takes for a technology to become accepted and to be integrated as standard practice, in part so that we can “package” the technology to make its adoption more likely. Redwine and Riddle (1985) found that “it takes on the order of 15 to 20 years to mature a technology to the point that it can be popularized and disseminated to the technical community at large.” In the worst case, it took 23 years to go from concept formulation to a point where popularization could be considered; the best case took 11 years, and the mean time was 17 years. Once a technology was developed, it took an average of 7.5 years for it to become widely available.

Clearly, 17 years is too long to wait in a business where time-to-market pressures require new technologies to prove themselves quickly. For example, in 1997, 50% of Hewlett-

Packard’s revenues were generated by products introduced within the previous two years. Even if not all of these products involve new technology, the relationship between revenue and speed of change tells us that we must move new technologies to the marketplace far faster than in previous years. Markets cannot wait a decade or two for technological innovation. For this reason, many development organizations grab new, promising technologies well before there is clear evidence of proven benefit. For instance, the US Software Engineering Institute's Capability Maturity Model was embraced by many companies well before the SEI and others began empirical investigations of the nature and magnitude of its process improvement effects. Similarly, many development teams are writing code in Java, even as the Java standard is evolving. In other cases, technology adoption is mandated when the technology seems to be so evidently beneficial that careful empirical evaluation is not considered to be necessary. The imposition of Ada as the standard Department of Defense programming language is one example of the use of standards to push technology adoption. Another is the British Ministry of Defence's insistence on the use of formal methods in the development of safety-critical systems, even when there was little evidence of the nature and extent of formal methods' benefits. Indeed, there is no consensus in the software engineering community on what we mean when we say something is a formal method, but the push to use such methods continues nevertheless. (Pfleeger and Hatton 1996)

Finding the right audience

This rush to adoption, frequently unsupported by careful study, has been successful in some cases but disastrous in others. Many organizations have made large investments in technologies that are no longer considered useful; indeed, many are no longer supported or available. For example, organizations have supplied their software developers with CASE tools that are no longer on the market. And organizations have trained developers in methodologies that are not really being used (or used properly) in development, either because they did not solve the problem they were intended to solve, or they did not fit the organizational culture. That is, the technologies were offered to the wrong audience.

Because the potential audience for a technology is not the same as the population of software developers at large, slow, incomplete or inadequate adoption is often due to addressing the wrong audience. That is, a great deal of time may be wasted in trying to convince users to adopt a technique, process or tool when in fact their use of the technology is not likely to result in significant, positive change. As a result, some organizations have studied technology transfer at the organizational or corporate level, rather than within the broader technological community. That is, they have tried to understand what is best for them, rather than what is best for everyone.

For example, Zelkowitz (1995) assessed software engineering technology transfer at the National Aeronautics and Space Administration (NASA). Unlike Redwine and Riddle, Zelkowitz focused on the particular problems experienced by NASA and how they were addressed by new technologies. In his report, Zelkowitz distinguishes technology transfer from infusion. Technology transfer is the insertion of a new technology into an organization that already performs similar tasks. Infusion is the incorporation of a new technology into an organization that had previously used nothing like it. Thus, infusion is the first step in technology transfer. In this paper, we use the two terms interchangeably.

Zelkowitz makes the important distinction between a technology producer and a technology consumer. In many cases, organizations create their own new technology and then encourage its widespread use. But in other cases, technologies are adopted by one group after they acknowledge its successful use by other groups. In either case, one or more "gatekeepers" are sometimes designated to identify promising technologies for a particular organization.

Roles, risk and the activities of technology transfer

Once a technology is selected, we can choose among several models to encourage the transfer of that technology (Berniker 1991):

1. *A people-mover model.* Relying on personal contact between technology developer and user, this model is the most prevalent and effective of those reported by Berniker. For example, a tool vendor may invite a development manager to a demonstration, and then use a personal relationship to promote the tool's use in the manager's organization. Similarly, an industry expert who has given the manager good advice in the past may suggest that the manager investigate a new and promising technology.
2. *A communication model.* Here, a new technology is reported in print and noticed by the gatekeeper. For instance, an article in *IEEE Computer,* the *Journal of Systems and Software, Dr. Dobb's Journal* or *Communications of the ACM* may report on the successful use of a new technology; a manager reads the article and decides to try it in his or her organization.
3. *An on-the-shelf model.* This approach relies on packaging and ease of use to make the technology appealing to potential users. For example, the millions of users of America Online's web browser have been attracted by its simple interface.
4. *A vendor model.* In this model, an organization relies on its primary hardware or software vendor to be gatekeeper, promoting new technologies when appropriate. For instance, the users of IBM or Sun hardware and software may be pleased with their purchases and eager to adopt new technology promoted by these vendors.

Zelkowitz identifies a fifth model:

5. *A rule model.* An outside organization imposes use of a technology on the development organization. This model can be invoked from inside, as when an organization imposes a development standard, or from outside, as when the

customer mandates a development process or language standard, such as DoD 2167a or Ada.

Rogers (1995), in studying technology transfer in many organizations (not just those related to software), also notes that there are distinct patterns in the way and speed with which technology is adopted. He distinguishes among innovators, early adopters, early majority, late majority and laggards. The first people to adopt a technology are innovators; they probably comprise only 2.5% of the total likely audience, as shown in Figure 1. Rogers explains that innovators are "venturesome," and they are driven by a desire to be rash and to do something daring. An innovator launches a new idea by importing it from outside of a system's normal boundaries. In this sense, an innovator is a gatekeeper for his or her organization.

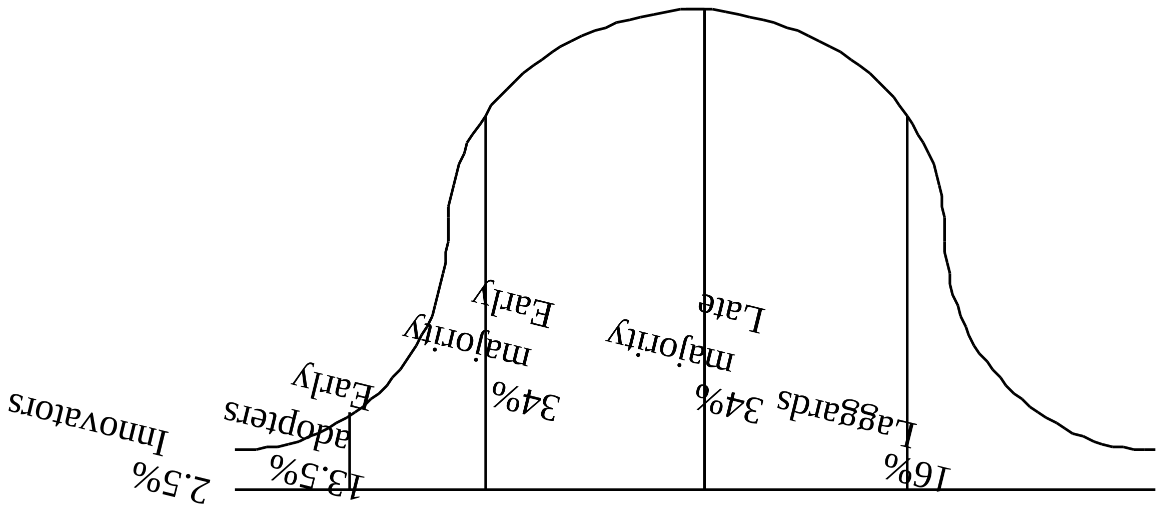

Figure 1. Adopter categorization (from Rogers 1995).

Early adopters are more integrated into an organization's development culture. They are respected by their peers, and they use new ideas discreetly. By making judicious innovation decisions, they decrease the uncertainty of a new idea by adopting it while personally informing colleagues of its success.

Early majority adopters are deliberate in their decision-making, thinking for some time before embracing a new technology. That is, they follow rather than lead, but they are willing to try new things demonstrated to be effective by others. Late majority adopters are more skeptical. Their adoption may be the result of economic pressures or peer pressures. Most of the uncertainty about a new idea must be resolved before a late adopter will agree to try it. Finally, laggards jump on the bandwagon only when they are certain that a new idea will not fail, or when they are forced to change by mandate from managers or customers.

Rogers' five categories correspond loosely to those of Berniker and Zelkowitz, in that different adopters use different styles. For example, innovators are people-movers, and they rely on personal contact to convince their colleagues to take a risk and try a new technology. Early adopters let someone else test the waters first. But when they read about the success of a technology in a respected publication, they build on someone else's success and introduce the technology to their own organizations. For example, the Software Technology Support Center at Hill Air Force Base (Ogden, Utah) evaluates technology and reports its findings regularly; early adopters may wait for something promising to be reported by STSC and then embrace those tools or techniques that sound appealing.

Early majority adopters are more cautious still. They can be convinced to try a technology not only when it has succeeded elsewhere but also when it is packaged with materials (such as training guides, help functions and simple interfaces) that make adoption relatively smooth and painless. Because late majority adopters dislike uncertainty, they find appealing the vendor model of technology transfer. The vendor can use examples of other customers' experiences to help convince the late majority that the technology will work. Finally, laggards usually adopt a technology only when they are commanded to do so. Rules imposed by an organization, a standards committee or a customer can encourage the use of a new technology when the other models fail. For instance, DoD endorsements of products, recommendations for process improvement, or mandatory rules about tools can encourage laggards to take risks and try new technologies. Thus, successful technology transfer requires not only a new idea but also a receptive audience with a particular adoption style.

We can summarize the relationship between models of adoption and receptive audiences in Table 1. Here, we note that as the technology adoption becomes less risky (that is, the body of evidence is more convincing), the less reluctant are practitioners to try it out.

Table 1. Relationships among adopters, risk, and likely transfer model.

Adopter category	*Level of risk*	*Adoption model*
Innovators	Very high	People-mover model
Early adopters	High	Communication model
Early majority	Moderate	On-the-shelf model

Late majority	Low	Vendor model
Laggards	Very low	Rule model

Successful technology transfer in a given organization

To determine the nature of successful technology transfer in a particular organization, it is instructive to look at which technologies have been most successfully transferred in your own organization. For example, Zelkowitz (1995) conducted a small study within and outside of NASA. Individuals were given a list of 200 technologies and told to list the five that had the most significant effect on their activities. He found that the top transferred technologies (both software and hardware) identified by each group surveyed (that is, within and without) were similar but not the same (as shown in Table 2), suggesting that some kinds of technology transfer are dependent on organizational need. However, the top five within and outside of NASA were the same.

Table 2. Top transferred technologies (Zelkowitz 1995).

Total replies (44)	*Numbe r*	*NASA replies only (12)*	*Number*
Workstations and PCs	27	Object-oriented technology	12
Object-oriented technology	21	Networks	10
Graphical user interfaces	17	Workstations and PCs	8
Process models	16	Process models	7
Networks	16	Measurement	5
C and C++	8	Graphical user interfaces	4
CASE tools	8	Structured design	3
Database systems	8	Database systems	2
Desktop publishing	8	Desktop publishing	2
Inspections	7	Development methods	2
Electronic mail	7	Reuse	2
		Cost estimation	2
		Communication software	2

Similar surveys have been administered by other researchers. For example, Yourdon (1998) notes

- declining interest in object-orientation
- growing interest in the year-2000 problem
- a linear decline in interest in CASE tools
- initial peaking but then declining interest in reuse

Glass and Howard (1998) surveyed 2600 IS practitioners, mostly managers or directors of IS in leading companies. They found that the top technologies in practice were fourth-generation languages, feasibility studies, prototyping, and code inspections or walkthroughs. At the same time, there was little interest in CASE tools, joint application development, and metrics. Thus, Zelkowitz's findings at NASA are similar to those reported elsewhere.

In his report, Zelkowitz describes the mechanisms that encourage technology transfer in NASA. Agents facilitate the transfer by assisting the technical staff. Repositories are set up with information about the technology, and gatekeepers watch for new and promising technologies. He discusses in depth the transfer of key technologies at NASA's Goddard Space Flight Center, including the Ada programming language, object-oriented technology, inspections, and the Cleanroom software development process. In concluding, Zelkowitz notes that technology transfer has had mixed results at NASA:

- There is no good infusion mechanism for bringing new technology to the agency. Because NASA is responsible more for research than commercial development, it is more likely to develop new technology for export to other groups than to use technology from other sources.
- The major goal at NASA has been transfer of products, rather than increases in productivity or quality.
- The people-mover model, successful in many other organizations, is rarely used at NASA.
- Most of the successful technology transfer at NASA was done outside of the mechanisms established by NASA explicitly for this purpose.

He also notes that, industry-wide,

- Most software professionals are resistant to change.
- Infusion mechanisms do not address software engineering technologies as well as they do other technologies. This problem may be the result of software engineering's process orientation, where the focus is more on producing than on transferring a product.
- Technology transfer requires far more than simply understanding a new technology.
- Quantitative data are important for understanding how and why the new technology will fit into or replace the existing processes.
- Technology infusion is not free.
- Personal contact is essential for change.
- Timing is critical.

Zelkowitz's findings confirm those reported by Berniker and Rogers. In his book, Rogers discusses how different kinds of adopters prefer different kinds of mechanisms. So it is important for us to understand our organizational cultures as well as the technology before we begin any technology transfer ¾ and indeed as we decide which technologies are good candidates for transfer. In particular, we can identify elements of the technology transfer process that act as promoters and inhibitors of the transfer. A *promoter* is a person, technique or activity that accelerates technology adoption. Similarly, an *inhibitor* is a person, technique or activity that interferes with or prevents technology adoption. In the next section, we examine a possible model for incorporating these elements into successful technology transfer.

The technology transfer process

In the last decade, several researchers have examined the effectiveness of technology transfer within information systems (IS) organizations. In each case, the investigators focused on a particular technology, rather than on technology transfer in general. For example, Rai (1995) performed a large-scale national survey of senior IS managers who were attempting to introduce widespread use of CASE tools. He found that the perceived effectiveness of knowledge transfer is associated differently with the initiation, adoption and implementation phases of the innovation process. That is, the managers had different reactions to the notion of using a new technology, depending on whether the technology was in its infancy, was being tried for the first time, or was a mature candidate for use in a particular organization. The "maturity" of the technology was itself a promoter.

We can use findings such as these, plus models suggested by Rogers, Zelkowitz, and Redwine and Riddle, to suggest a process for how successful technology transfer might be attempted by development organizations. We begin by identifying five key activities, each of which answers an important question about the technology under scrutiny. As shown in Figure 2, the first activity, technology creation, starts with a business need and asks whether a technology exists that might address it. Sometimes the appropriate technology exists and has been used before to solve the same or a similar business problem. At other times, the technology exists, untried on this problem or never tried in practice. And at still other times, no appropriate technology exists, so it must be created by researchers or developers in-house or elsewhere.

Once a candidate technology is found, the next step is a preliminary investigation to determine whether there is evidence that the technology will work in practice. This evidence can range from anecdotal accounts of successful use to carefully-documented studies involving experimental methods and statistical analysis. For example, if the organization is considering using design inspections (the technology) to help reduce faults injected during design (the business need), then the preliminary evaluation includes gathering data on design inspections from books, journals, magazines and colleagues.

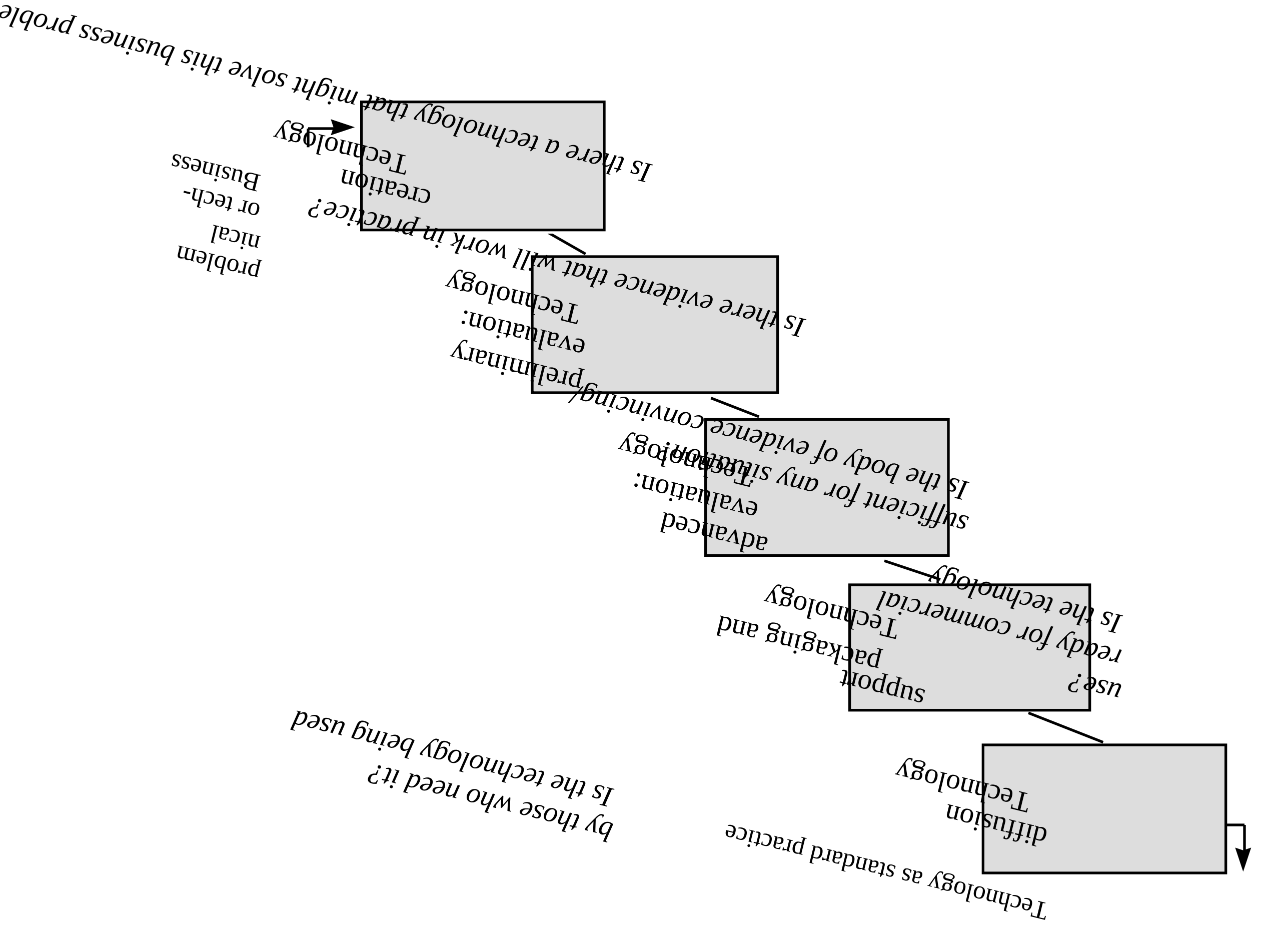
Business or tech-nical problem
Is there a technology that might solve this business proble
Technology creation
Is there evidence that will work in practice?
Technology evaluation: preliminary
Is the body of evidence convincing/sufficient for any situation?
Technology evaluation: advanced
Is the technology ready for commercial use?
Technology packaging and support
Is the technology being used by those who need it?
Technology diffusion
Technology as standard practice

Figure 2. The steps from idea to standard practice.

Once the evidence indicates that the technology has worked in practice, the next step is a more thorough evaluation of the body of evidence. We want to determine the credibility of the evidence: In what situations did the technology work? What is the nature of the studies, in terms of experimental and statistical methods? If the technology continues to look promising (that is, if the body of evidence is credible and compelling), we look next at tools and other packaging and support to aid the technology’s use. With compelling evidence combined with commercially-viable support, we take the final step: promoting adoption with those who are likely to benefit from using the technology.

We can expand the five steps to a technology transfer process, as shown in Figure 3. It is drawn using simple SADT diagrams. Thus, the arrows from the left are inputs, the arrows to the right are outputs, the arrows from above are constraints, and the arrows from below are mechanisms for accomplishing the activity represented by the box.

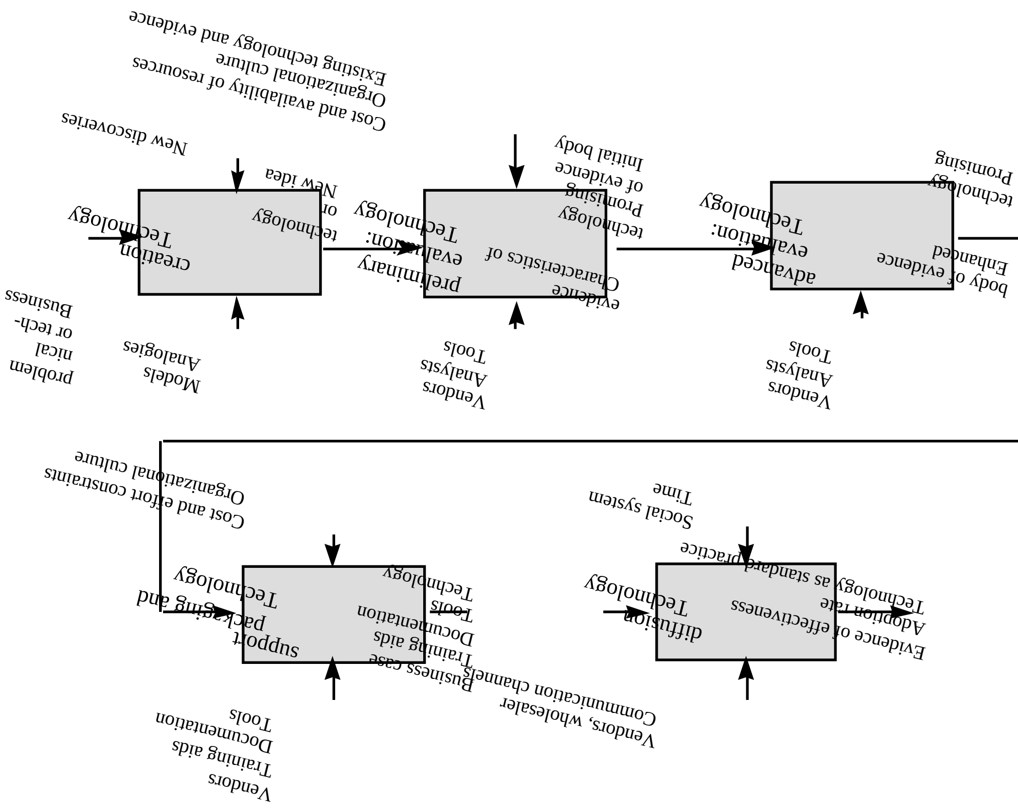

Figure 3. The technology transfer process.

In this model, the technology is created in response to a business or technical problem experienced by an organization, and supported by models of current understanding and analogies to know solutions or past experience. Next, the preliminary evaluation is performed by those who use the technology for the first time to see if it indeed solves the problem it was intended to address. Notice that this step includes activities sensitive to the organizational culture, so that a proposed technological solution is viewed in the context of how it will be received by the organization or project that will use it. This view involves identification of both promoters and inhibitors. That is, we must know who or what is likely to help technology adoption and effectiveness, and who or what is likely to hinder them. For example, the Rogers roles play a big part in determining the reception given to a particular technology, based on who is promoting it, whether similar technologies have been tried in the past, and how different the technology is from what is being performed currently.

This growing body of evidence is then subjected to a more advanced evaluation, where we examine not only the technology but also the evidence itself. That is, to assess whether the body of evidence forms a compelling argument for using the technology, we look at the situations in which the technology has been used, compare the old ways to the new ways, and determine (using surveys, case studies, experiments, feature analysis, and other techniques) whether the evidence is conflicting, consistent and objective. We want to know how "solid" a case we have for believing that the new technology will solve the business problem it addresses.

However, compelling evidence is not enough to assure technology adoption. In addition, the technology must be packaged and supported so as to make it "friendlier" and easier to understand and use. Technology transfer is helped and hindered by technological promoters and inhibitors, respectively, just as it is affected by organizational ones. For example, tools and supporting written material go a long way in assisting a user to adopt a new technique or process. When the supporting infrastructure is in place to offer this assistance, the technology is "ready for prime time." That is, this packaged technology is finally ready for broader diffusion, so that we as a community can assess its adoption rate and evidence of effectiveness as more organizations report the results of their experiences.

This model is admittedly just a simplistic overview. In fact, we may need multiple models of diffusion, based on the characteristics of the technology, the adopters, and the evidence. Still, this model offers us a vocabulary with which to discuss key issues affecting the success of technology adoption and diffusion. In the next sections, we look at the model's steps and concepts in more detail.

Technology creation

Technology creation reacts to a business need or technical problem. Thus, for a given technology, we must ask:

- What problem does it solve?
- Does it work properly?
- Does it replace/expand/enhance an existing technology?
- Does it fit easily in the existing development or maintenance process, without great disruption to established and effective activities?
- Is it easy to understand?
- Is it easy to learn?
- Is it easy to use?
- Is it cost-effective?

Typically, the gatekeeper asks questions like these, to determine if a particular technology is a good candidate for adoption by a given organization. Assuming the answers to the questions suggest that the technology be tried and perhaps adopted, we move to the first of the two evaluation steps.

Technology evaluation: preliminary

When a gatekeeper or manager finds an appealing new technology, two issues must be addressed. First, the organization must evaluate the technology relative to the organization's existing technologies and processes. That is, the organization wants to know whether there is any benefit to using the new technology relative to what they are doing now. Second, if the technology is useful in this limited context, we then want to know if the benefit extends to other similar or dissimilar organizations. The answer to the first question is the result of an initial or preliminary technology evaluation. If the evidence produced by this primary evaluation is not convincing, there is no need to continue through the next technology transfer activities in our example process model.

The nature of the evidence

In fact, a recent study by Zelkowitz, Wallace, and Binkley (1998) highlights the often vast difference between the perception and reality of a technology. They show that the research and practitioner communities have very different ideas about which preliminary evaluation results suggest that a technology will be successful. In this study, about 90 researchers and software practitioners were asked their perceptions of the value of various experimental methods necessary to validate a new technology. They found that practitioners value most the methods that are relevant to their environment. That is, techniques such case studies, field studies, and replicated controlled experiments were considered to be important to the decision-making process in choosing a new technology.

On the other hand, researchers prefer to use reproducible validation methods that can be used in isolation in the laboratory, such as theoretical proof, static analysis, and simulation. They discounted methods that required interacting directly with practitioners. In other words, researchers shunned the very methods that are most valued by practitioners: case studies, field studies and experiments. Thus, to create a body of evidence for evaluating a technology, researchers and practitioners go down two very different paths. Moreover, the body of evidence provided by researchers is not likely to be taken seriously by the practitioners who are thinking about using the technology!

We can represent and emphasize this difference by looking more closely at two of the steps in Figure 3. Preliminary evaluation is often a research task, while advanced evaluation is often an industry task. In this case, the body of evidence produced from the research task differs from the body of evidence required for the next step in the process. If technology transfer is to be successful, we must find ways for researchers to produce evidence that is read, understood and believed by practitioners.

Thus, the Zelkowitz, Wallace and Binkley study shows us that successful technology transfer requires understanding of the message as well as the messenger, both in terms of absolute evidence and in terms of perception and credibility. Such an assessment should be part of the preliminary technology evaluation.

The questions addressed by the evidence

Assuming that we are producing evidence of interest and credibility to both researchers and practitioners, what questions should the evidence try to answer? Rogers' theory of innovation diffusion (1995) suggests some key ones:

- *Relative advantage:* To what degree is the new technology better than what is already available?
- *Compatibility:* To what degree is it consistent with existing values, past experiences, and needs of potential adopters?
- *Complexity:* To what degree is it easy to understand and use?
- *Trialability:* Can it be experimented with on a limited basis?
- *Observability:* Are the results of using it visible to others?

Each question can be addressed in many different ways. Case studies, experiments, surveys, feature analyses, and other quantitative and qualitative techniques can focus on one or more of the issues, each producing some evidence for or against more global adoption of the new technology. Pfleeger (1998) describes some of these techniques in more detail; we do not address them here. Because the overriding goal of the technology adoption is to improve at least one product, process or resource or some way, the evidence should help to determine if the new technology causes the improvement. That is, we want to investigate the cause-and-effect relationship between the new technology and one or more variables of interest. Even if the benefit is the same as existing or competing technologies, we may choose the new technology simply because it reduces the uncertainty in the cause-and-effect relationship. In other words, we want the results of our development and maintenance processes to be more predictable.

The state of the evidence

However, sometimes the evidence is not clear or is conflicting, even with carefully-controlled studies. For example, the ease with which graphical user interfaces can be developed using object-oriented techniques attests to its effectiveness in enhancing reuse. But other studies show that object-orientation is not necessary for successful reuse. (Griss and Wasser 1995) Thus, we must study the body of evidence and analyze relationships among pieces of evidence; if we cannot resolve or understand conflicts, then the evidence may not be useful.

The evidence we generate can take several forms, as shown in the two columns of Table 3. The structure of Table 3 helps us to organize and sort the evidence, so that we have an overview of the evidence that supports using the proposed technology. Tangible evidence includes objects, documents, data and relationships that demonstrate direct benefit from the technology being evaluated. For example, a CASE tool may increase productivity directly by enabling all developers to record the relationships among the design components. However, the evidence can also be circumstantial, showing that the new technology was used when some benefit was experienced. For instance, productivity may have increased when a CASE tool was used, but it is not clear that using the CASE tool caused the increase. By contrast, the evidence can be indirectly relevant, as when the effects of the new technology may be confounded with other variables, such as the use of a new design technique at the same time that the CASE tool was introduced, so it is difficult to distinguish which cause resulted in the perceived effect.

Table 3. Forms of evidence. (Adapted from Schum 1994)

Type of evidence	Characteristics
Tangible	· objects · documents · images · measurements · charts · relationships
Testimonial (unequivocal)	· direct observation · second-hand · opinion
Testimonial (equivocal)	· complete equivocation · probabilistic argument
Missing tangibles or testimony	· contradictory data · partial data
Authoritative records or facts	· legal documents · census data

Testimonial evidence can be of two types, equivocal and unequivocal. Unequivocal evidence is generated by direct observation, such as when the new technology results in lower cost or higher quality. This benefit may be elicited through opinion surveys (for instance, rating a product on a scale, so that one product is perceived to be better than another), rather than through objective measurement. Equivocal testimonial evidence is less credible but still useful; respondents are uncertain but think that the new technology probably yielded some benefit.

Sometimes the evidence has missing components; we see benefit in some cases but not all, or there is testimony for some projects but not for every one. Similarly, we may have official records of project or product data, but we may have no indication of how the evidence was gathered, so we do not know how much weight to give the results.

This preliminary evaluation results in creating and understanding an initial body of evidence that may support technology adoption. The next step is to examine the evidence carefully to determine under what conditions the technology is likely to work best.

Technology evaluation: advanced

Schum (1994) describes in great detail the analysis of evidence to determine whether and how it supports the degree to which one variable causes a particular effect. He suggests several important issues to consider when evaluating the body of evidence. First, we must categorize the type of evidence in more detail than Table 3. In particular, we want to know if the evidence is testimony, heuristics, or authoritative evidence. That is, is the evidence provided by the vendor as testimony, or by users who rate the technology on some kind of scale, or by practitioners who evaluate the technology objectively and in some quantitative way? In that context, we also want to know if the judgments of cause and effect are absolute or relative. For example, does the use of the new technology produce higher-quality products all the time (i.e. an absolute improvement), or only when used under certain conditions (i.e. an improvement relative to the conditions under which it is used)?

Second, we must decide how much confidence we have in our judgments, based on the strength of the evidence. Schum calls this *ampliative induction:* the process of forming probabilistically-hedged conclusions on the basis of evidence. Sometimes this confidence is related to the degree of control we have in the studies we have performed. We may be able to say that a change in quality is definitely the result of having used a technology, because we have controlled all other variables carefully. But there are other times when we can say only that it is probable or possible that the result we see is caused by the new technology; other variables (such as experience or complexity) were not under our control.

Third, we examine the process by which the evidence is being generated. Sometimes this process is iterative; we come to a preliminary conclusion with some degree of confidence, and then we revise our conclusions and confidence level as new evidence is generated. Building up a more complete body of evidence can amplify our confidence, but it can just as easily provide conflicts. One study may show clear benefit, but the next study may show no difference at all from the current methods. Although at first this conflict indicates lack of confidence, in fact larger bodies of evidence help to show us which variables are the most important; sometimes they point out variables we had not considered in the early studies. Thus, the evidence-building process helps us to narrow in on exactly what situations are best for using a particular technology.

We must also take a fourth notion into account: the structure of the argument made from the evidence. Each piece of evidence does not stand on its own. We create a fabric of an argument out of threads of evidence, with different evidence playing differing roles in supporting our ultimate conclusion. We must assess the evidential force of the argument by asking pertinent questions about each supporting element:

- Is each piece of evidence relevant to the argument?
- What is each piece of evidence's inferential force? That is, with respect to the hypothesis, how much evidence is there, and in what direction does it push our argument's conclusion?
- What is the evidential threshold? That is, what is the point below which the evidence is irrelevant? This question is particularly important in software engineering, where small sample sizes or unrepresentative situations may make some evidence irrelevant.
- What is the perspective of the provider of the evidence, and how does the perspective affect the conclusion?
- What is the nature of the evidence? Is it documentary, testimonial, inferential, or some other category of evidence?
- How credible is the evidence? Evidence from several sources is usually more credible than evidence from a single source. And the credibility of the source affects the credibility of the evidence, not just in terms of who is providing the evidence but also in terms of how and when the evidence was collected.
- How accurate is the evidence?
- How objective was the evidence collection and results?
- How competent are the evidence providers and interpreters?
- How truthful are the evidence providers and interpreters?

Figure 4 provides several guidelines for answering the last few questions in this list. It notes that some of the evidence may be contradictory or conflicting, and our conclusions about the technology must take these imperfections into account. For example, we cannot dismiss object-orientation for reuse simply because the evidence is conflicting. Rather, we must evaluate the nature of the conflict; then, we can decide whether conflicts mean that we can have little confidence that the technology will achieve its goal, or whether they are useful in helping us distinguish those cases when the technology is sure to work.

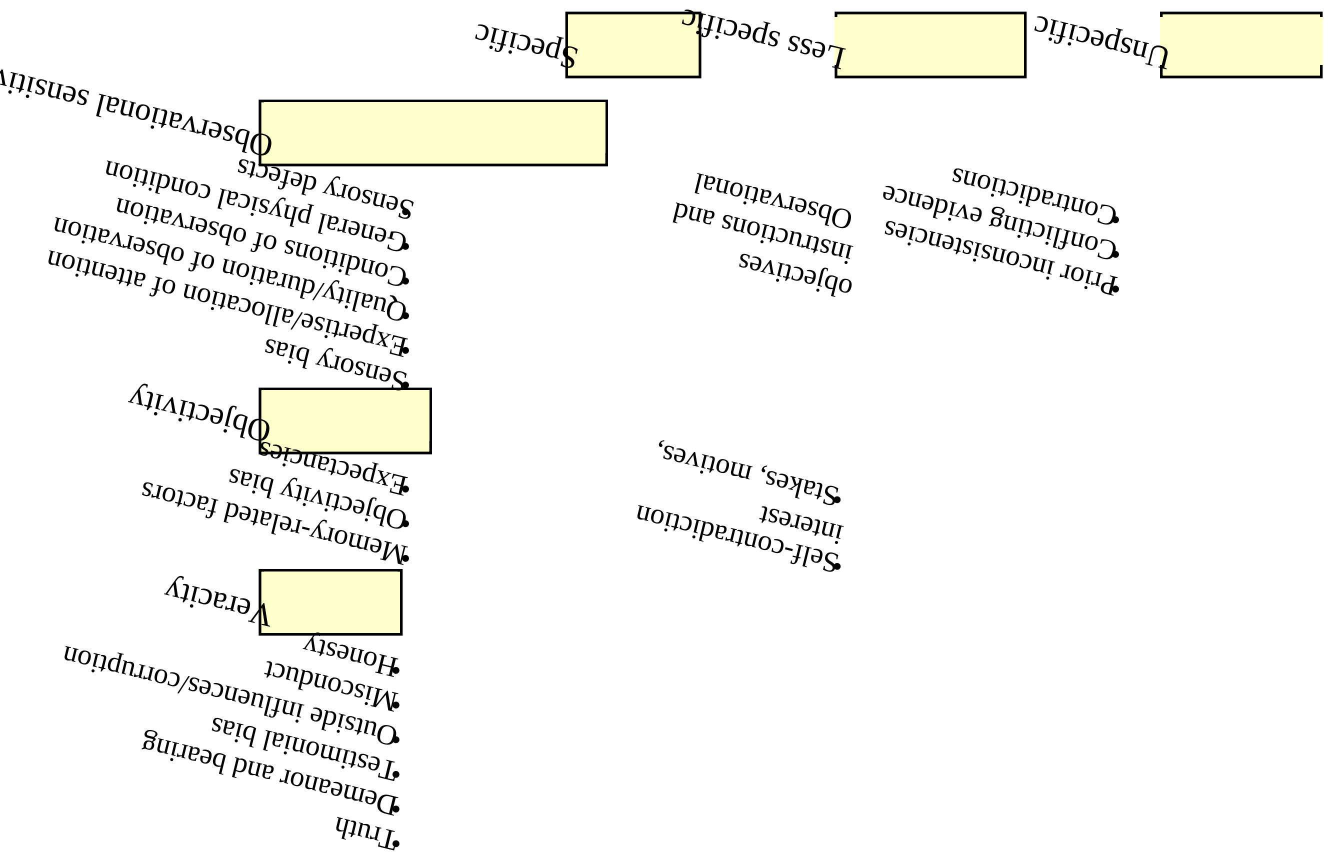

Figure 4. Evidential tests of testimonial credibility. (Adapted from Schum 1994)

Technology packaging and support

Once the evidence clearly demonstrates that a technology is useful and effective, descriptions of its use are not enough to encourage technology transfer. Fichman and Kemerer (1997) performed an empirical study of 608 information technology organizations using object-oriented programming languages. They found that packaging and support were necessary to break down the "knowledge barriers" preventing practitioners from using such languages. In particular, they note that practitioners will try new technologies when the burden of organizational learning is diminished in some way, either because much of the technical knowledge already exists in their organizations, or because the knowledge can be acquired easily or cheaply.

Thus, accompanying compelling evidence must be effective packaging that provides positive answers to the following questions:

- Are there effective tools, documentation and training aids to assist learning and using the technology?
- Is there institutional support?

- Is there cognitive dissonance with existing techniques? That is, if a potential user already knows one technique, does that prevent him or her from learning the new one?
- Has the technique been commercialized and marketed?
- Is the technology used outside the group that developed it?

To package a technology appropriately, we may need to enlist a wholesaler: someone to understand and promote the technology in ways that appeal to the needs and preconceptions of the potential audience. For example, the US Defense Information Systems Agency often customizes tools and techniques to suit its DoD audience. Its CASE readiness initiative aimed not only at selecting useful tools but also at preparing developers for using the tools most effectively. However, as Zelkowitz, Wallace and Binkley (1998) make clear, biases may prevent adoption, in spite of clear evidence of utility. The wholesaler can use the biases to his or her advantage, making adoption success more likely.

Technology diffusion

Once we have convincing evidence of the effectiveness of the new technology, plus appropriate packaging and support, we can transfer the technology to a wider audience. Several software-engineering-related studies have reported that there are key elements that help ensure the success of this transfer. For example, Premkumar and Potter (1995) surveyed information systems managers involved in CASE tool adoption. They found five variables useful in differentiating technology adopters from non-adopters:

- the existence of a product champion
- strong top management support
- lower information systems expertise
- a perception that CASE has greater relative advantage over other alternatives
- a conviction of the cost-effectiveness of CASE

Similarly, Lai and Guynes (1997) surveyed Business Week 1000 companies to determine what affected their decision to adopt ISDN technology. Companies most receptive to ISDN were larger, had more slack resources, more technology expansion options, and fewer technology restrictions.

There is substantial literature about the diffusion of innovation that offers suggestions from many disciplines about the best way to approach technology transfer. For example, Prescott and Conger (1995) examined a representative sample of 70 information-technology-related research studies. They found that traditional diffusion-of-innovation theory is most applicable to information technology organizations where the impact of the technology is within the organization, requires less organizational support, and the technology is deemed important because of its functionality and efficiency. Diffusion across organizations is more affected by contextual and environmental variables, so that economic influence and critical mass of users play a more important role in adoption.

Rogers (1995) provides general guidelines in this regard. First, he suggests that we determine if the technology changes as the user adopts and implements it. For example, the considerable body of evidence about inspections and reviews contains notes about how the inspection/review process is tailored to the organizations' needs. That is, the technology matures or evolves as studies are performed. In fact, many of the papers reporting inspection success are not studying inspections as Fagan originally described them. We must take technology evolution into account in our models, since it is unlikely that we can freeze a technology over the long period during which we must study it. This evolution is not considered in medical and biological models for empirical research; social science models may be more appropriate in helping us understanding software engineering technology.

Second, we must understand the potential audience for the technology. How alike are the potential users, and how does their "heterophily" affect the time it takes them to adopt the technology successfully? That is, we should look at the similarities between those who have already adopted and those who are about to do so. If there is a large difference in characteristics between the two groups, Rogers suggests that technology adoption will be more difficult than if the groups are very much the same. The audience should also be analyzed in terms of the probability of their adopting the technology early. We have seen that his several categories of adopters differ in the speed at which they adopt, depending on their adoption style and on the model of adoption preferred.

Third, we should understand the diffusion process itself. Rogers describes five of its aspects:

- knowledge
- persuasion
- decision
- implementation
- confirmation (leading to adoption or rejection)

These aspects are closely related to the assessment of evidence and packaging described earlier. However, there is little reported in the software engineering literature about how each aspect relates to successful software technology diffusion. Much of our understanding of these issues is anecdotal; we can draw stronger conclusions about successful technology transfer if our understanding is broadened using careful studies of representative situations.

Fourth, we must understand the role of the people involved in the diffusion process who can encourage successful adoption. Rogers calls them opinion leaders, change agents, or aides, depending on how they interact with potential adopters. We can examine our notion of "wholesaler" to determine whether it is the same as or different from these roles. Rogers also points out that organizational norms can help or hinder adoption.

The next steps

This paper highlights the issues involved in software engineering technology transfer. Although a broad survey, several things are clear.

1. There is great variety in adoption times, measured from idea conception to incorporation as standard practice. Moreover, even the short adoption times are too long for today's market pressures.
2. It is not clear how to build and assess any evidence about a technology's effects on products, processes and resources when we have minimal control of most of our software development and maintenance projects (with respect to experimental variables).
3. We know little about how developing a body of compelling evidence of the technology's effectiveness relates to the degree to which a technology is adopted. We need models that address not only the technology but also other key aspects of technology diffusion. Rogers (1995) defines diffusion as "the process by which an innovation is communicated through certain channels over time among members of a social system." In the past, our studies of software engineering technology diffusion have spent little or no time investigating the communication (its nature and channels), time or social systems involved.
4. Evidence is not enough to ensure adoption; packaging and support are crucial to widespread, sustained adoption. In particular, much research has created or examined models of technology transfer, but little of it has looked into the characteristics of promoters or inhibitors.
5. We can learn much from the diffusion and evidential literature about key aspects of successful technology transfer. In particular, we should examine the literature about diffusion in the education community, since it is there that the technology tends to change and evolve as it is evaluated.

Practitioners assisting researchers

As the pace of technology generation quickens and as software becomes more pervasive in our lives, we cannot afford to make investments in technology whose adoption is either undesirable or not likely to be successful. Practitioners can help researchers take steps to understand the nature of successful technology transfer, and use that new understanding to build models and support for effective evaluation and adoption. In particular, some of the current empirical studies of software engineering technologies are loosely-related and poorly-planned: not a very good collective generator of a coherent body of evidence. If empirical software engineering researchers would organize studies so that each piece contributes a clear and significant result to the whole, the resulting body of evidence would be more compelling. That is, practitioners and researchers together must plan what to address in each study and how that study's result will contribute to the overall body of evidence.

To that end, collaborative work between researchers and practitioners using new technology can be enhanced in at least three key ways:

1. Researchers should look for examples of technology transfer and assess them to understand the key variables that make adoption fast and effective.
2. Researchers should develop a set of guidelines for evaluating bodies of evidence. These guidelines can tell both practitioners and researchers how to organize a series of studies and how to participate in them so that we can produce good evidence. They can also tell us when we have enough evidence to make a reasonable decision about a technology, as well as how to deal with issues such as conflict and inconsistency.
3. Practitioners should assist researchers in studying and understanding diffusion theory and its associated literature, determine which of its principles apply to software engineering technologies, and participate in studies that will help build models of technology transfer. These models, in turn, can act as guidelines for organizations that want to maximize technology investment and adoption effectiveness.

Practitioners using technology

Practitioners interested in using new technologies effectively can learn from the many studies already performed and apply their lessons to their own projects. In particular, they can include technology transfer promoters and avoid technology transfer inhibitors when introducing new techniques, processes and tools on our projects. The inhibitors and promoters fall into three categories, as shown in Table 4: technological, organizational and evidential. As we have seen, technology transfer is inhibited by lack of packaging, by lack of relationship to a pressing technical or business problem, and by difficulty of use and understanding. By contrast, the technology can be promoted with tools, a well-understood context, and an understanding of clear potential benefit to those on the project now or who will use the resulting product later. In other words, the technology is easier to transfer if it is easy to use and easy to see how it will help.

Table 4. Technology transfer inhibitors and promoters.

	Inhibitors	**Promoters**
Technological	· Lack of "packaging"	· Supporting tools,

	· No clear relationship to technical problem · Difficult to use · Difficult to understand	manuals, classes, help · Clear benefit to technical problem · Well-understood context for using the technology · Easy to use · Easy to understand
Organizational	· No management support · Cognitive dissonance · Biases and preconceptions · Cross-organizational mandate · Not cost-effective · Heterophily	· Gatekeeper · Technology booster · Trial within one organization · Results (especially improvement) visible to others · Perceived advantage · Success in similar organizations · Compatible with values · Compatible with experience · Compatible with needs · Cost-effective · Homophily
Evidential	· Conflicting evidence · Lack of evidence · Unclear meaning of evidence · Experiments in toy situations	· Consistent evidence · Case studies and field studies · Credible messenger · Relative advantage · Cause-and-effect evident

Similarly, biases and preconceptions can prevent an organization from adopting a technology successfully, as can interference from similar technologies. We can avoid these inhibitors by using a technology champion, by making the improvement visible to others, by obtaining institutional support, and by making sure that all on the project understand how the technology relates to organizational needs.

Finally, practitioners can evaluate current evidence and help researchers to base new studies on existing findings. Whether practitioners are designing new studies or just participating in them, they should have credibility in the conveyors of information about the technology, be sure that the technology is clearly going to be the cause of anticipated effects, and apply the technology to a realistic problem in the field, not just a toy problem with no relevance to real work.

The bottom line is this: We do not know all we need to know about which software engineering technologies work best in which situations. And we cannot wait for years to know, given the blistering speed at which technology changes. But there are lessons to be learned, both from software engineering research and from other disciplines with similar interests in technology evaluation and transfer. By working with both practitioners and researchers, we can take steps now both to speed up the rate of adoption and to build a credible body of evidence about the effectiveness of our practices.

Acknowledgments

Many thanks to Bernard Woolfolk, Marvin Zelkowitz, Winifred Menezes, Likoebe Maruping, Victor Basili and James Bieman for their suggestions and comments.

A portion of this work was supported by contract 1800312 between ITT Systems and the University of Maryland.

References

Berniker, E., “Models of technology transfer: a dialectical case study,” *Proceedings of the IEEE Conference: The New International Language*, pp. 499-502, July 1991.

Fichman, R.G. and C. F. Kemerer, “The assimilation of software process innovations: an organizational learning perspective,” *Management Science*, 43(10), pp. 1345-1363, October 1997.

Glass, Robert and Alan Howard, “Software development state-of-the-practice,” *Managing System Development*, June 1998.

Griss, Martin and Marty Wasser, Quality Time column, *IEEE Software*, January 1995.

Lai, V. S. and J. L. Guynes, “An assessment of the influence of organizational characteristics on information technology adoption decision: a discriminative approach,” *IEEE Transactions on Engineering Management*, 44(2), pp. 146-157, May 1997.

Pfleeger, Shari Lawrence, *Software Engineering: Theory and Practice*, Prentice Hall, Englewood Cliffs, New Jersey, 1998.

Pfleeger, Shari Lawrence and Les Hatton, “Investigating the influence of formal methods,” *IEEE Computer*, February 1997.

Premkumar, G. and M. Potter, “Adoption of computer-aided software engineering (CASE) technology: an innovative adoption perspective,” *Data Base for Advances in Information Systems* 26(2-3), pp. 105-124, May-August 1995.

Prescott, M. B. and S. A. Conger, “Information technology innovations: a classification by IT locus of impact and research approach,” *Data Base for Advances in Information Systems* 26(2-3), pp. 20-41, May-August 1995.

Rai, A., “External information source and channel effectiveness and the diffusion of CASE innovations: an empirical study,” *European Journal of Information Systems*, 4(2), pp. 93-102, May 1995.

Redwine, Samuel T. and William E. Riddle, “Software technology maturation,” *Proceedings of the Eighth International Conference on Software Engineering*, IEEE Computer Society Press, Los Alamitos, California, pp. 189-200, August 1985.

Rogers, Everett M., *Diffusion of Innovations*, fourth edition, Free Press, New York, 1995.

Schum, David A., *Evidential Foundations of Probabilistic Reasoning*, Wiley Series in Systems Engineering, John Wiley, New York, 1994.

Yourdon, Edward, *Application Development Strategies* newsletter, February 1998.

Zelkowitz, Marvin V., “Assessing software engineering technology transfer within NASA,” NASA technical report NASA-RPT-003095, National Aeronautics and Space Administration, Washington, DC, January 1995.

Zelkowitz, Marvin V., Dolores R. Wallace and David Binkley, “Understanding the culture clash in software engineering technology transfer,” University of Maryland technical report, 2 June 1998.