Assessing Software Engineering

Technology Transfer

within NASA

Marvin V. Zelkowitz
Institute for Advanced Computer Studies
and Department of Computer Science
University of Maryland,
College Park, Maryland 20742

November, 1994

Abstract

Technology transfer is of crucial concern to both government and industry today. In this report, we address the issue of technology transfer within the National Aeronautics and Space Administration (NASA) and in particular emphasize the domain of technologies for software development. We examine the established NASA mechanisms for achieving technology transfer and then address in some detail several successful examples of software engineering technology transfer. We analyze these examples and provide conclusions and suggestions of how the practice of technology transfer is perceived to work within NASA and how it actually works. In particular, this report distinguishes between the concept of technology transfer, or the adoption of a new method by large segments of an industry, and technology infusion, or the adoption of a new technology by an individual organization.

Table of Contents

Chapter 1. Introduction

The ability to move a new technology from the development laboratory into general use in industry is the ultimate test of the applicability of scientific development – whether it be software engineering or any other branch of science or engineering. This process, generally called *technology transfer*, is of crucial concern to U. S. industry today as the need to remain economically competitive in a global marketplace forces all organizations to constantly improve their mechanisms for doing business. Government is not immune from these forces and needs to understand and participate in such activities at all levels.

The National Aeronautics and Space Administration (NASA), as a large government agency, plays a role as both a producer and consumer of such new technologies:

As producer. As the premier space agency of the United States, NASA has a mission to develop aeronautical and space technologies. Transferring these technologies to private industry and aiding in the commercialization of those technologies allows for government help in promoting U.S. industry internationally.

> “Technology transfer is a fundamental mission [of NASA]. It is as important as any NASA mission and it must be pursued.”[1]

As consumer. However, with an annual budget of over $15 billion, NASA is involved in a great many activities, and using the best techniques – whether developed internally or developed by those outside of NASA – enables NASA to wisely use its appropriated funds to work on complex tasks as economically as is practical.

Of great concern to industry is the need to improve productivity. Within the software engineering community, the ability to improve the process of developing software has been found to be a major impetus towards improving productivity and reliability of the resulting systems. Concepts like the Software Engineering Institute’s Capability Maturity Model [12] have grown in importance as a means for gradually improving the software development process. The Experience Factory concept of the NASA/Goddard Space Flight Center (GSFC) Software Engineering Laboratory (SEL) [5] has shown the value of process improvement in quantifiable defendable ways for eigtheen years.

However, all process improvement involves changes. Some of these may be relatively minor alterations to the current way of doing business (e.g. replacing one compiler or editor by another). However, some may require major changes that affect the entire development process (e.g., using the Cleanroom software development methodology and eliminating much of the former unit testing phase).

In order for an organization to continually improve its process, it must be aware of how the organization operates and what other technologies are available that may be of use. Understanding this process of technology transfer should enable NASA to better use its existing resources and to better plan for the future.

NASA understands the need for technology transfer and has instituted mechanisms for achieving that goal. This study was instituted to understand how NASA addresses technology transfer and to see how well the NASA approaches apply to the domain of software development. By reading relevant NASA documents and interviewing relevant NASA personnel, we tried to understand how several software engineering technologies were imported into or exported out of NASA. In particular, we are interested in the mechanisms

---

[1] Daniel S. Goldin, NASA Administrator, December, 1992.

that were used to accomplish the transfer of the technology, the effort involved in performing that transfer, and the time that it took to accomplish. This report extends the results presented in an earlier draft [21].

This report is organized as follows. In Section 2 we discuss the general problems of technology transfer and how this has been addressed in other organizations. In Section 3 we describe the mechanisms that NASA has created for addressing technology transfer. In Section 4 we summarize some recent technologies that have been transferred in the software engineering area, and in Section 5 we describe how certain technologies have been exported to organizations outside of NASA. We summarize our results and give some conclusions about our NASA experiences and some recommendations for improving this process in Section 6.

Chapter 2. Technology Transfer in Industry

By *technology transfer* we mean the insertion of some new technology into organizations that perform similar technological tasks. The insertion must be such that the new organizations regularly use that technology if the appropriate conditions on its use should arise in the future. We generally think of technology transfer as an industry-wide phenomenon. That is, technology transfer occurs when predominantly all organizations in the same application domain use that new technology.

A related concept is *technology infusion*, or the incorporation of a new technology into an organization that previously did not use that technology. Technology infusion is the necessary first step to a technology transfer process. Technology transfer can be thought of as technology infusion across a broad segment of a given industry.

We will call the original creator of that technology the *producer* of the technology and the organization that accepts and uses the new technology the *consumer* of that technology. The process of moving the technology out of the producing organization will be called *exporting* the technology while the process of installing the technology in the new organization will be called *infusing* the technology.

Implied by the above definitions is the notion that a successfully transferred technology becomes part of the state-of-the-practice, or normal operating procedures, of the infusing organization. For example, an organization that experiments with Ada as a programming language and then decides to use it for all applications in a specific domain (e.g., for all flight simulators) can be said to have successfully transferred that technology. On the other hand, if a technology is tried once or twice (e.g., the Prolog programming language for expert system development) and is found wanting and will not be used again, then that technology will not be considered to be transferred.

Not transferring a technology does not imply that the technology is ineffective; only that it does not apply to that particular consumer domain. For example, there is still a demand for buggy whips among horse enthusiasts and certain theme park operators, but there is little demand for them within most urban automobile repair shops.

2.1 Models of Technology Transfer

Technology transfer has typically been identified as an importation process. One member of an organization, often called the *gatekeeper*, monitors technological developments and chooses those that seem appropriate for inclusion in an organization – hence opens the “gate” to the new technology. Because this role is usually spontaneously (and informally) assigned, it often falls naturally to the most creative and technically astute individual in an organization. Besides being the one generally aware of technical developments outside of the organization, others in the group look towards this person for guidance, so it is natural to assume that this person provides the technology transfer leadership.

Organizations can do more to foster technology transfer than to wait for this ad hoc process to slowly move forward. Encouraging new technology – especially when there is no gatekeeper in the organization – is an important management role.

In addition to the gatekeeper model of technology transfer, several other models of the process have been identified. In one study of 44 technology transfer efforts at one aerospace company, Berniker [6] identified four approaches towards technology transfer, listed in general order of effectiveness:

1. **People mover model.** In this approach, there is personal contact between the developer and the user of a technology. Typically there is some facilitator within the infusing organization that knows about the new technology and wishes to import it into the new organization. This is the gatekeeper model mentioned above, which was found to be the most prevalent and effective of all technology transfer methods.

   Nochur and Allen [11] investigated this model and discovered that it really consists of three separate subcategories, which we call:

   1. **Spontaneous gatekeeper** role assumed by organization member.
   2. **Assigned gatekeeper** role imposed (by management) on some organization member.
   3. **Umbrella gatekeeper** role assumed by another organization to impose new technology on others.

2. **Communication model.** In this approach, the new technology has appeared in print and, as with the people mover model, some facilitator discovers the technology and wishes to infuse it into the new organization. The “print” mechanism may be internal documentation, conference reports or journal publications. Much like the people mover model, this approach requires some facilitator within the infusing organization to install the new technology.

3. **On-the-shelf model.** This approach, relatively rare among the projects studied by Berniker, requires the new technology to be packaged so that non-experts can discover it and learn enough about it to begin the infusion process. It requires the designer of the method to sufficiently document the method so that others can easily pick it up and use it. Reading about the technology in a “parts catalog” or buying “shrink-wrapped” version of the software are examples of this method.

4. **Vendor model.** This last method assumes that the organization that needs the new technology turns over the task to a vendor to sell it to them. It effectively turns the vendor into an external gatekeeper.

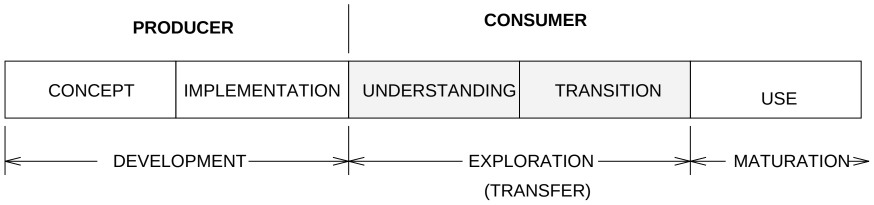
All four models consider technology transfer as an importation process. That is, an organization needs to solve a problem and decides to try a new way to do business. If successful, this new technology is used on future projects until it becomes part of the standard development process in the organization.

While Nochur and Allen found that the assigned gatekeeper was somewhat effective in importing new technology, it could not continue the transfer by moving it to other internal organizations (in essence acting like the umbrella gatekeeper). The third of these gatekeeper roles was most ineffective – technology generally has to wanted by the new organization and cannot be dictated by outsiders. However, the umbrella gatekeeper is really misclassified. It is not a people mover strategy, since it is not a technology importation process, but instead represents an exportation of technology from one organization to another. We will call this new model the rule model:

5. **Rule model.** An outside organization imposes a new technology on the development organization, which then infuses it into its own development process.

There are many examples within the federal sector of this last technology transfer model. The mandating of the Ada language by the Department of Defense’s Ada Joint Program Office for system development, the use of the Software Engineering Institute’s Capability Maturity Model to evaluate developer’s qualifications for a Department of Defense contract, and the use of Federal Information Processing Standards (FIPS) by the National Institute of Standards and Technology (NIST) are all examples of technology transfer imposed by an outside agency.

Industry is not immune to the rule model either. Organizations have imposed tool standards on their subunits (e.g., imposing particular hardware and operating system products, CASE tools, or database



Exploration stage is main object of current NASA study

Figure 2.1: Technology Maturation Life Cycle

systems), and organizations need to react to rules imposed by government contracting organizations, such as Request for Proposals that mandate a particular language, such as Ada. But in practice, however, successful examples of this imposed technology are rare. (It is not even clear if the above three instances are successful examples of imposed technology transfer.) However, it is a model of technology transfer that has been discussed and we must not lose sight of it in our study.

2.2 Software Engineering Technology Maturation

In 1985, Redwine and Riddle [13] published the first comprehensive study of software engineering technology maturation. Their goal was to understand the nature of technology maturation – what was the length of time required for a new concept to move from being a laboratory curiosity to general acceptance by industry. They defined maturation of a technology as a 70 per cent use level across the industry.

Technology maturation involves five stages – two by the producer of the technology and three by consumers of that technology (See Figure 2.1):

- The original *concept* for the technology appears as a published paper or initial prototype implementation. The initial time period is the development of the concept by the originator of the technology.
- The *implementation* of the technology involves the further development of the concept by the originator or successor organization until a stable useful version is created.
- In the initial *understanding* (or experimentation) stage, other organizations experiment, tailor, adapt, expand, modify and try to use the new technology.
- In the later *transition* (or exploration) stage, use of the technology is further modified and expands penetration across the industry.
- As *use* of the new technology grows, the final *maturation* stage is reached when 70 per cent of the industry uses the technology.

In this report, we will group the concept and implementation phases together as the *development* phase by the producing organization and the understanding and transitioning phases as the *exploration* phase of the consuming organization.

In their study, they looked at 17 software development technologies that were developed from the 1960s through the early 1980s (e.g., UNIX, spreadsheets, object oriented design, etc.). Among their results, those most related to this current study are:

- They were unable to clearly define "maturation" for most technologies, but were able to make reasonable estimates as to the length of time needed for new technologies to be widely available.
- Technologies required an average of 17 years to pass from an initial concept to a mature product.
- Technologies, once developed, required an average of 7.5 years to become widely available.

While technology transfer is an important national policy goal, individual organizations are most interested in infusing technology into their own organization. Therefore, in this report we are most interested in the exploration stage as being the main component of the technology infusion process of an organization. In the next section we discuss several potential technologies that have been infused within the industry within the last decade. In the next chapter we look at NASA technology infusion in greater detail.

2.3 Successfully Transferred Technologies

"Technology" is a very imprecise concept. For this report we are mainly concerned with tools, procedures and mechanisms that aid in the development of software products. We can divide this domain into two categories:

Infrastructure technology. This includes the tools and procedures used by the software engineering profession to build software. It includes, in addition to the usual computer-based items like machines, editors, compilers, testing tools and configuration management systems, items like electronic mail, desktop publishing, spreadsheets and any other tool or device useful for software production. This can even include general technology products such as the telephone or fax machine if either provides relevant aid in the development of software.

Software engineering technology. Software engineering technology is that subset of infrastructure technology specifically designed for software development. Thus, while it will include compilers and testing tools, it will not include items like electronic mail or the fax machine which also have uses in other domains.

Before we can understand software engineering technology transfer within NASA, it was first necessary to understand if there was any consensus about how infrastructure technology has evolved over the past decade. Papers are often written about the so-called "software crisis" with comments that software development has not changed at all in 30 years. If this were so, then obviously no technology has been transferred lately.

In order to address this, a brief informal survey was conducted both within and outside of NASA. The survey asked for the top five technologies that have affected software development practices since 1980. A list of approximately 100 items was offered, and the respondent could pick from that list or add any other items that seemed relevant.

A total of 44 responses were returned. Of these, 12 were from NASA/GSFC personnel or NASA contractors. While not enough for a definitive study of the issues, it did seem sufficient for some informal observations about the effectiveness of software engineering technologies. Table 2.1 presents The top ten (including ties) from the total list and the NASA list of respondents.

The level of consensus among the 44 returned forms was quite surprising. The top five in the list clearly dominated the others. Of the top five, only two of them (objected oriented technology and process models) were clearly software oriented technology. Two were mostly hardware (workstations and networks) and the other (graphical user interfaces) was partially a software engineering issue, but does not strictly fit into our

Total Replies (44)		NASA Replies (12)	
Item	**No.**	**Item**	**No.**
Workstations & PCs	27	∗Object oriented	12
∗Object oriented	21	Networks	10
Graphical user interfaces	17	Workstations & 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

∗ – Software engineering technologies

Table 2.1: Top Ten Transferred Technologies

definition of software engineering technology given its use as an infrastructure technology in many other application domains.

Among the NASA replies, there was likewise strong agreement with the total list. The top five on the total list were five out of six among the NASA entries. Only measurement, a strong component of the SEL, raised the recognition level among NASA personnel of its importance.

Other technologies that have been claimed as effective that were mentioned in the survey include: Measurement (in 12th place), Ada (in 14th), Formal methods (in 17th) and UNIX (in 17th).

Clearly, our initial goals for this survey were confirmed: There *were* technologies which have been transferred to aid in software development, and there was a fair amount of agreement of which ones were most effective. The questions we must now address: Which software development technologies have been transferred into NASA, which ones did NASA play a role in developing, and how did the transfer process work compared to the mechanisms NASA has put in place for performing technology transfer?

Chapter 3. NASA Model of Technology Transfer

In describing the transfer of technology (both infusion and exportation of technology) into and out of NASA, we have four potential groups of producers and consumers to consider (Table 3.1). NASA may be either a producer or a consumer of some technology. Similarly, some external organization may be either the producer or consumer of that technology. Of the four potential cases, only three are considered in this report – those involving NASA as either a producer or consumer. The case where an external consumer transfers a technology from an external producer is certainly of interest, but is outside of the scope of this work on NASA's role in technology transfer.

Producers ↓ Consumers →	**NASA Consumers**	**External Consumers**
NASA Producers	Transferred within NASA	Exported from NASA
External Producers	Infused into NASA	*Not of interest*

Table 3.1: Participants in Technology Transfer

3.1 Importance of Technology Transfer

NASA recognizes the importance of technology transfer, and NASA has long since funded offices such as the Technology Utilization Office (TUO) to foster technology transfer; however, NASA has recently found that technology transfer within the agency has not been as effective as desired, and the mechanisms in place are now undergoing revision to better adapt to the realities of the marketplace. This report represents the status of those changes as of July, 1994.

A team of senior managers surveyed the agency and found [18]:[1]

1. The agency's record of active technology transfer was not as extensive as generally perceived by the general public.
2. NASA now could benefit by actively seeking and acquiring technology from academia, industry, and other government agencies.
3. The best way to work with industry is to learn, then emulate the private sector's way of doing business.
4. Having NASA and industry work jointly on R&D projects should permit technology to move in both directions effectively.
5. NASA should work with selected industry segments where it can make a difference.
6. Agency staff performance evaluations and rewards should take into account the commercialization program.

Partly based upon the recommendations in that report, the NASA Commercial Technology Network (NCTN) is being created as a means to focus technology tranfer issues and aid the agency in transferring (i.e., commercializing) technology developed at NASA. In Section 3.2 we describe the NCTN and in Section 3.3 we explore those segments of the process that refer specifically to software development issues.

---

[1] These conclusions are consistent with the results of this present technology transfer report.

3.2 NASA Technology Transfer Mechanisms

The NASA Technology Transfer Program is organized at NASA headquarters under the Office of Advanced Concepts and Technology (OACT). At present, this is in a great state of flux. Existing offices and mechanisms are being changed and new procedures are being developed. Acronyms come and go seemingly monthly. For the most part, each NASA center has a fair amount of freedom to establish its own technology transfer mechanism according to the following general plan.

The NASA Commercial Technology Network (NCTN) consists of seven components that address aspects of developing new technology and moving those technologies into the private sector. Information about the NCTN is organized around publically available repositories using MOSAIC software on the Internet (mosaic address: http://nctn.oact.hq.nasa.gov).

The components of the NASA technology transfer program consist of:

National Technology Transfer Network (NTTN). The NTTN is the core of the older NASA technology transfer program. It consists of the Technology Utilization Officer at each of the nine NASA centers and the six regional technology transfer offices. COSMIC (Computer Software Management and Information Center – the NASA software repository at the University of Georgia), and the two application development teams complete the NTTN structure. We will discuss the NTTN components in greater detail in the next section.

Partnership Options for NASA and Industry (PONI). PONI represents the collection of joint NASA and industry partnership agreements. These can be nonreimbursible Space Act Agreements (SAA) where industry has access to NASA personnel and facilities to cooperative agreements such as jointly sponsored research agreements and reimbursible Space Act Agreements. Through May 1994, NASA has participated in approximately 950 of such agreements, with approximately 700 of such agreements being active at any one time. NASA has approximately 2400 agreements annually with industry, other government agencies and educational institutions.

Small Business Innovation Research (SBIR). SBIR is a research program for companies of 500 or fewer employees. Successful companies follow a three phase process: Small initiation concept grants, larger development grants, and a third phase commercialization grant. It is similar to sponsored research grants under PONI except that it is focused on the small business.

Space Technology Innovation (STI). Space Technology Innovation is a bimonthly newsletter describing new technology developed as a result of OACT-funded projects.

Aerospace Industry Technology Program (AITP). This is a new program to help in developing partnerships between NASA and aerospace companies for commercialization of space-related technology.

Repository Based Software Engineering (RBSE). Repository Based Software Engineering is a project of the University of Houston – Clear Lake Research Institute for Computing and Information Systems to improve software development via repository-based reuse of components [14]. It began in 1991 as a redirection and restructuring of the AdaNET repository operated in West Virginia. The goal is to foster reuse among NASA centers by building a repository-based library of reusable software components. Currently the University of Houston is working on pilot programs with NASA/ Johnson Space Center (JSC) and NASA/Langley Research Center (LaRC). In addition, the RBSE is a member of the Strategic Avionics Technology Working Group and the Software Technology Working Group to help coordinate and focus NASA software technology research.

Technology Tracking System (TechTracs). This is an online tracking system for providing the public with information on what projects OACT has in the NCTN snd what commercialization efforts are now underway.

In the next section we describe which of these mechanisms are relevant to technology transfer within the software engineering domain, and how they address the issue of software engineering technology transfer.

3.3 Software Engineering Technology Transfer Mechanisms

Given the set of mechanisms of the NCTN, as defined in the previous section, as well as other traditional technology transfer vehicles, such as conferences and journals, how does NASA address technology transfer within the software engineering domain? We will group these according to three approaches:

1. **Agents** for technology transfer. Agents are facilitators of technology transfer that assist the actual technical staff to effect the transfer.
2. **Repositories** for technology transfer.
3. **Gatekeepers** for technology transfer. The gatekeepers, like the agents, do not perform the actual technology transfer, but as we shall show, they take a more active role in performing technology transfer.

Recall that in Section 2.1, we identified five methods to achieve technology transfer:

1. People mover model – personal contact.
2. Communication model – published papers on new technology.
3. On the shelf model – packaged technology available for use.
4. Vendor model – Outside organization proposes new technology.
5. Rule model – Regulations imposed by third party.

We can associate, for each of our three approaches, the methods that these represent:

- **Agents –** Communication and vendor models. (It may seem at first glance, that these are also people mover approaches, but we will later show that the people mover aspect to these mechanisms are minimal.
- **Repositories –** Communication model and on–the–shelf model.
- **Gatekeepers –** People mover and vendor models.

Note that the rule model is not represented in the above list. NASA has no organization that imposes NASA-wide (or even center-wide) standards in the software development arena. Each group chooses its own development process. For example, the space station mandates Ada source programs, but there is no NASA-wide or even center-wide policy on Ada use.

RTTC	Location
Far West	University of Southern California, Los Angeles, CA
Mid-Continent	Texas A & M University, College Station, TX
Mid-West	Great Lakes Industrial Technology Center, Cleveland, OH
Southeast	University of Florida, Alachua, FL
Mid-Atlantic	University of Pittsburgh, Pittsburgh, PA
Northeast	Center for Technology Commercialization, Westborough, MA

Table 3.2: Regional Technology Transfer Centers

3.3.1 Agents

These mechanisms consist of the primary technology transfer components within NASA. There is a three level process for achieving technology transfer: (1) Each center maintains a technology utilization office for discovering technologies ripe for commercialization. (2) There are several regional technology transfer centers for coordinating activities among NASA centers. (3) Also two technology application teams act as “consultants” to the NASA centers and play the role of facilitators in getting industry and NASA personnel talking together about potential projects.

Technology Utilization Office. Each NASA center has a Technology Utilization Office (TUO – also called a technology transfer office or TTO) which has as a primary goal, aiding NASA scientists and engineers to bring their developments to the attention of industry. For example, at NASA/GSFC, the TUO is the Office of Commercial Programs (OCP) within the Engineering Directorate (Code 702). Technology transfer is enabled with the aid of this office. Scientists and engineers bring their products to the TUO, and if considered novel, the concept results in one of three primary products: (1) a patent, (2) a notice in *NASA Tech Briefs*, or (3) as software submitted to COSMIC.

New inventions are submitted to this office where they are evaluated as to their patentability, and if so, a patent application is submitted. Improvements to existing technology are also submitted to this office, and if deemed appropriate, they are described in *NASA Tech Briefs*. Companies are contacted and the TUO works with NASA personnel in preparing the product for commercial applications. COSMIC software will be described later in Section 3.3.2. This role of the TUO can be viewed as fitting the Communication model of technology transfer.

The TUO interacts with the Regional Technology Transfer Centers of the National Technology Transfer Network to aid in the commercialization of NASA technology. In this role, the TUO acts as a facilitator for the NASA engineer and can be viewed as part of the vendor model of technology transfer. The TUO and the RTTCs have similar roles. Both have contacts with industry, although the TUO is focused more towards industry local to the NASA center, while the RTTC has a scope of 5 or 6 adjacent states within the region served by the NASA center. Both provide mailings, meetings, and presentations at local industry shows in order to make organziations aware of the new developments at NASA.

While the NASA/GSFC TUO is oriented towards submitting software to COSMIC, The NASA LaRC Technology Application Group (TAG) takes a more proactive role in technology transfer. It promotes the transfer of technology out to industry quickly and enhances this process by creating teams of customers and NASA/LaRC personnel during development.

National Technology Transfer Network. The National Technology Transfer Network (NTTN) consists of the National Technology Transfer Center (NTTC) at Wheeling Jesuit College in Wheeling, WV and six Regional Technology Transfer Centers (RTTCs) (See Table 3.2). Providing a data base of relevant technology that is accessible by industry is a major component of the NTTC work. Each RTTC works with the NASA centers in its region and with local industry in aiding NASA scientists and engineers to commercialize their projects. For example, members of the Northeast RTTC and Mid-Atlantic RTTC regularly participate with OCP/TUO personnel at NASA/GSFC in order to make NASA/GSFC developments known to a larger class of companies than is available locally to GSFC. As with the local TUO at each center, the components of the NTTN can be viewed as facilitators of the vendor and communication models of technology transfer.

Technology Application Team. There are two Technology Application Teams (TAT) (located at Research Triangle Institute (RTI), Research Triangle Park, NC and at the University of New Mexico, Albuquerque, NM). The TAT acts as a consultant to each NASA center. The TAT maintains an interdisciplinary staff to be able to assess the commercial potential of technologies developed at each center. NASA-developed patents are studied to determine their market potential. The TAT maintains lists of corporations that would be able to use such technology, produces "Technology Opportunity" information sheets of potential products that it distributes to interested corporations, and is visible at industry trade shows and associations to learn what problems segments of industry are facing. For the most part, the TAT provides similar services to the RTTCs, and is part of the communication and vendor models of technology transfer.

3.3.2 Repositories

Various publications and data bases inform the public of information available about NASA technology. All of these are examples of the communication model of technology transfer with COSMIC also providing on the shelf technology transfer methods.

COSMIC. The Computer Software Management and Information center (COSMIC) is a repository of software developed within NASA. It is located at the University of Georgia. Over 5,000 programs have been submitted to COSMIC for distribution since 1966, and some 1200 programs are currently active in the COSMIC repository. Software for COSMIC is first submitted to the local TUO at the relevant center before being transmitted to COSMIC for evaluation. One limitation of COSMIC, however, is that software must be in the public domain. COSMIC does not address transfer of software that could be proprietary and sold to its user community. The RBSE program at the University of Houston – Clear Lake is trying to address this aspect of the problem.

NASA Tech Briefs. *NASA Tech Briefs* is a free monthly publication for announcing new inventions and innovations from NASA developed or NASA sponsored research. It is a joint publication of NASA and Associated Business Publications of New York. Each local TUO at the nine NASA centers submits patent applications and other technology improvements for publication in *NASA Tech Briefs.* In 1993, NASA/GSFC processed 45 Tech Briefs, 5,549 requests for information and approximately 2400 telephone requests related to previously published Tech Briefs [10]. Approximately 135,000 of these requests are received NASA-wide [19].

Spinoff. *Spinoff* is an annual publication [19] that summarizes some of the technologies that have been successfully commercialized during the previous year.

Space Technology Innovation. This is a new bimonthly publication describing the results of NASA-sponsored research and development. It is available on-line as part of the NCTN-mosaic database available on the Internet.

Center for Aerospace Information. The Center for Aerospace Information (CASI), located in suburban Baltimore, MD and run by NASA headquarters, coordinates technology transfer publications. It produces the annual *Spinnoff* report and maintains the mailing lists for *NASA Tech Briefs*.

NASA Commercial Technology Network. The NCTN is a new data base service being organized on the Internet. Using MOSAIC software and starting from NASA headquarters Internet servers *http://nctn.oact.hq.nasa.gov* or *http://www.nasa.gov*, the user has access to numerous data bases describing most of the mechanisms given in this report. However, the NCTN is still in its infancy and many of the “screeens” that users need to nagivate through to discover what they need in the NCTN are still being developed.

The Internet and the World Wide Web (WWW) represents one of the new mechanisms for information transfer. With an easy to use interface, users have the ability to “surf” the Internet and acquire information from numerous data repositories quite easily. Most organizations have been developing WWW “home pages” with links to other pages permitting the user to trace through an organization looking for any necessary documents. Early 1994 has ssen an explosion in the number of WWW pages, and WWW traffic on the Internet is now the dominant form of data transmission.

Along with this explosive growth is a corresponding lack of standardization in how WWW pages are linked. Information from one site is linked to another site in a fairly random manner. This includes both NASA locations and other organizations. It is clear that in order for the WWW to remain a viable search mechanism on the Internet, some set of standards in how information from one location references another must be developed.

The Internet and the WWW are already having a major impact on information technology dissemination. This will affect all organizations, the NCTN included. The impact that the NCTN will have on technology transfer needs to assessed again in a few years after WWW use has matured and after the NCTN has a chance to develop, to mature, and to gain a critical mass of users.

Conferences and publications. Conferences and publications (both NASA sponsored and non-NASA sponsored) are a major source of information on technology that has been produced both within and outside of NASA. NASA annually runs the “Technology 2000” conferences (most recently Technology 2003 on December 7, 1993 in Anaheim, CA) which brings together developers of new technologies from federal laboratories (in addition to NASA scientists) and industry representatives who are interested in commercializing those technologies. For software engineering technologies, the largest meeting is the annual Software Engineering Workshop during the first week of December, which annually attracts over 500 attendees.

3.3.3 Gatekeepers

As we have stated several times already, personal contact is the most effective means for technology transfer. Sponsoring research and being a direct partner in that research allows both the government and industry to develop the appropriate gatekeepers to infuse the technology. NASA research and development models include the following:

Partnership Options for NASA and Industry. Partnership Options for NASA and Industry (PONI) include several methods for joint government – industry work. Space Act Agreements (SAA) are like Memoranda of Understanding (MOU) or CRADAs (Cooperative Research and Development Agreements in other federal agencies) for joint industry–NASA cooperation on specific projects. SAA are individual agreements between industry and NASA scientists for joint work. These may be either non-reimbursible or reimbursible agreements. The SBIR program and the RBSE program are other examples of NASA-funded research and development activities.

Research Centers and Consortia. NASA has established several research centers for investigation of scientific problems. For example ICASE (Institute for Computer Applications in Science and Engineering) near NASA/LaRC was organized to study fluid dynamics and STScI (Space Telescope Science Institute), located at Johns Hopkins University in Baltimore, was created to operate the Hubble space telescope. These have clear scientific missions. However, others have been created where software development activities are part of the charter of the organization, as given in the list below. A related development in the technology transfer area has been the rise of consortia, i.e., segments of an industry working together to solve related problems. This is viewed as an effective way to share risk and raise the necessary capital needed for projects with a large risk component. It also provides a mechanism for sharing information among the participants in an effective manner. Several of these NASA-sponsored research groups and consortia are:

1. **AITP.** The Aerospace Industry Technology Program is a new FY 1994 program to aid in the commercialization of aerospace technologies through partnerships with aerospace corporations. Still in its first year, the program needs some time to see how it progresses.
2. **CESDIS.** The Center of Excellence in Space Data and Information Systems was established at NASA/GSFC to try and understand the problems in processing, storing, and transmitting the huge amounts of data that are being transmitted by the various satellites. Although mostly concerned with software needed by scientists to process space data, aspects of this work may be related to software development problems. CESDIS has been in operation for about 5 years and has not made much of an impact yet.
3. **RICIS.** The Research Institute for Computer and Information Systems was established between NASA/JSC and the University of Houston – Clear Lake (UHCL) as a joint partnership consisting of the university, NASA/JSC and local Houston industry. RICIS is addressing research areas in software reuse, group support systems, medical imaging, and mission and safety critical systems. Software reuse and some aspects of mission critical systems are related to software development technologies.
4. **SEL.** The Software Engineering Laboratory (SEL) is organized around the Flight Dynamics Division (Code 552) at NASA/GSFC. Although not part of the technology transfer program of OACT, it has still proven to be a very effective model of technology transfer. The organization of the SEL is described in greater detail in Section 4.2.
5. **RBSE.** The Repository Based Software Engineering Program is another example of a NASA-sponsored consortia that is part of the NCTN. This is described more fully in Section 3.2.

Merging this list of technology transfer mechanisms with our previous model of technology transfer participants (Table 3.1), we get a clearer picture of how NASA addresses technology transfer (Table 3.3).

3.4 Exporting and Infusing Technology

For the most part, the established mechanisms – NTTN, TUO, and NCTN – are designed to move NASA-developed technology into the commercial arena. However, there is a corresponding need for NASA to find and use technology to solve many of its own problems. Technology infusion is not as well developed within the agency, but some mechanisms, which have already been described, aid in this effort. For the most part, the consortia (e.g., AITP, SEL, RBSE) and PONI activities, while fostering commercialization of NASA-developed products, also puts NASA scientists in direct contact with their industrial counterparts. This aids the people-mover model of bringing new technology to the agency.

	NASA Consumers	**External Consumers**
NASA Producers	COSMIC (communication, on-the-shelf) Tech Briefs (communication) Spinoff(communication) STI(communication) NCTN(communication) Conferences (communication) Papers (communication)	COSMIC (communication, on-the-shelf) Tech Briefs (communication) Spinoff(communication) STI(communication) NCTN(communication) Conferences (communication) Papers (communication) TAT (communication, vendor) RTTC (communication, vendor) TUO (communication, vendor) PONI(people mover, vendor) Consortia(people mover, vendor)
External Producers	Conferences (communication) Papers (communication) PONI(people mover, vendor) Consortia (people mover, vendor)	*Not of interest*

Table 3.3: NASA Technology Transfer Model

NASA also funds programs, such as the SBIR program, which allows NASA to fund promising research that it may need to use later. This, however, is an indirecct form of technology transfer since by its very nature it is not always clear what results you will achieve by any specific research program.

Chapter 4. Infusion of Technology

In this current study, we are not interested in the general issue of technology maturation, but instead the infusion of a technology into or out of a single organization (NASA). Therefore, we would expect Redwine and Riddle's 7.5 year average exploration stage to be an upper bound. Once a new technology is identified, we would like to understand the time interval for infusion of this technology.

4.1 Technologies of Interest

In order to understand technology transfer in NASA, after the brief technology transfer survey was performed (Section 2.3), about 20 software experts across all nine NASA centers were interviewed by telephone (two or three at each center) to determine which software engineering techniques were being used effectively in the agency. To keep the scope of this problem manageable, the following two restrictions were imposed:

1. The technology had to clearly be software engineering. For example, successfully transferred programs, such as the widely-used modeling system NASTRAN available through COSMIC, were not included.
2. The technology had to have a major impact on several groups within NASA. With more than 3,000 software professionals affiliated with GSFC alone (including government and contractors), almost every software product has probably been used somewhere in the agency. The major technologies that were found to have been transferred are given in Table 4.1.

Once we generated the list of technologies given in Table 4.1, we chose several of them for more indepth study. We contacted the developer of that technology to obtain detailed information of the process that was used. The goal was not to explain the technology, itself, but to understand the steps that were taken to move that technology from its original developers to new users. We were most interested in the process that was used to move that technology, whether that process used the established NASA mechanisms for technology transfer, how long the process took, and what level of effort was required.

	NASA Consumers	**External Consumers**
NASA Producers	Reuse(KAPTUR), ∗ CLIPS ∗ GUI(TAE), CASE tools ∗ Measurement(SME, GQM) AI tools	Reuse(KAPTUR), ∗ CLIPS ∗ GUI(TAE) ∗ Measurement(SME, GQM)
External Producers	Cost models, ∗ Formal Inspections ∗ Object Oriented Technology, CASE tools ∗ Ada, C, C++, ∗ Cleanroom Rate monotonic scheduling, Process modeling	*Not of interest*

∗ – Technologies discussed in more detail

Table 4.1: Transferred Technologies At NASA

Unfortunately, we were not able to get more than general qualitative measures on cost of technology transfer. Except for projects monitored by SEL, which keeps detailed metrics on most development activities, obtaining concrete numbers for several infusion attributes was almost impossible. We were, however, able to address the other goals for our study: what process was used and how long it took to do technology infusion.

The technologies that follow do not necessarily represent solutions to the most important software development problems. For example, understanding the software process is clearly an important goal, and several approaches have been tried, such as the SEL use of the Experience factory model and the Software Engineering Institute's Capability Maturity Model (CMM). Neither, however, has been adopted by a substantial number of NASA sites In Section 5.2 we discuss use of the Experience Factory, but there is no NASA consensus on the use of either, and they do not yet fulfill our criteria of being a transferred technology.

In the remainder of this chapter, we discuss several technologies that have been infused into NASA. We first give several examples of technology infusion by the Software Engineering Laboratory (SEL) of NASA/GSFC. Because of its emphasis on measurement, experimentation and data collection, the SEL has detailed data on the infusion process of several of these technologies. We then discuss other infusion efforts within NASA. In Chapter 5 we discuss the exporting of technologies developed by NASA to other organizations.

4.2 Technology Infusion By The SEL

The Software Engineering Laboratory (SEL) is a joint research activity consisting of NASA/GSFC, Computer Sciences Corporation and the University of Maryland. The SEL was established in 1976 within the Flight Dynamics Division of NASA/GSFC to investigate software engineering technologies applicable to flight dynamics problems. The Flight Dynamics Division, along with CSC, has the responsibility for developing operational software for spacecraft operations, while the University, along with its two partners, investigates relevant research questions applicable for improving this software. Small pilot studies at the University, small case studies at NASA and full scale implementation of these ideas on operational software provide a wide range of opportunities for converting research ideas into practice.

The SEL operates on the principles of (1) Understanding the process to be studied; (2) Assessing the impact the process has on the product being developed; and then (3) Packaging the process via tailoring and modifications so that it is available for use by NASA and other organizations. Some technologies that the SEL has recently investigated include: (1) Development of measurement and metrics in the software engineering domain; (2) The use of the Cleanroom model for improving the reliability of software; (3) The effectiveness of Ada as a programming language within the NASA domain, and more recently, (4) The development of the Experience Factory Model for software development. The SEL is the foremost organization for the collection of development data on software development activities with data from over 100 projects collected during the past 18 years. Its Software Engineering Workshop annually attracts over 500 attendees each December. Well over 300 papers and reports have been written these past 18 years describing the results of SEL studies.

Many technologies have been studied by the SEL and have become state-of-the-practice within the Flight Dynamics Branch (e.g., measurement, process improvement) and have been reported elsewhere [4]. Figure 4.1 briefly summarizes the evolution of the SEL since its inception in 1976. *Understanding* studies are first undertaken to provide baselines of the phenomenon of interest. Figure 4.1 shows this evolution; initial studies in 1976 were mostly concerned with the basic issues of data collection and resource and error analysis while today the interest is more in reuse, maintenance, and design issues. *Assessing* studies then

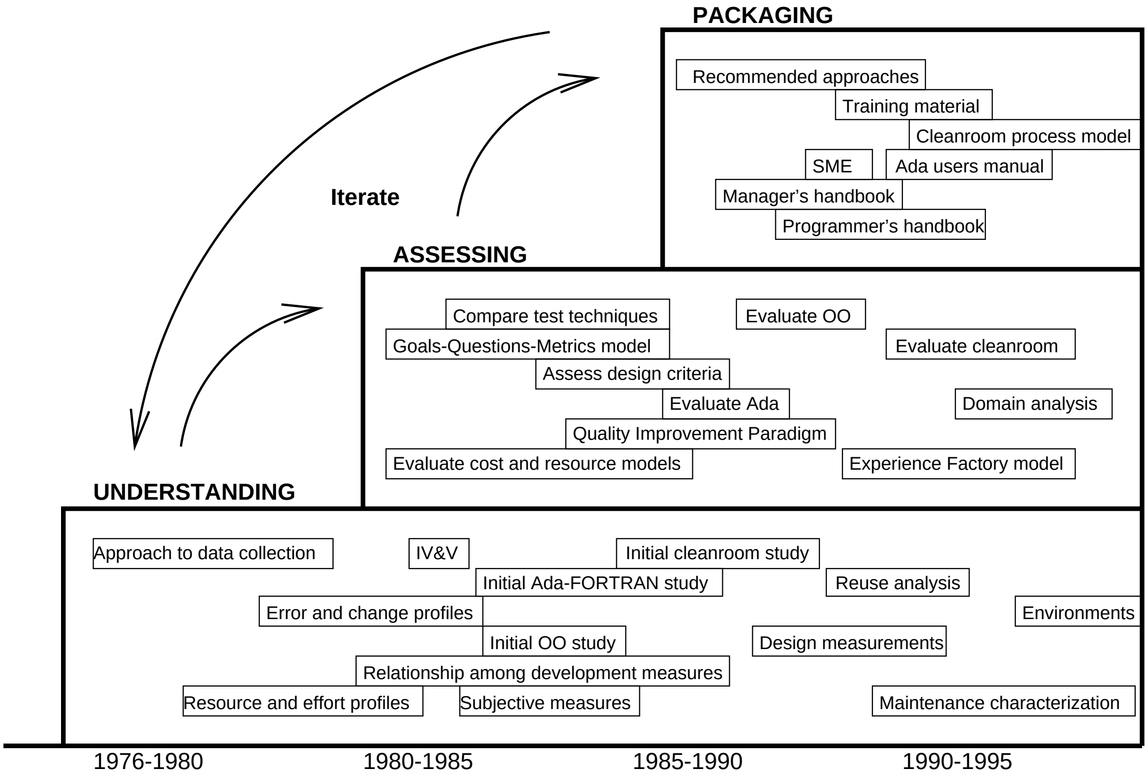


Figure 4.1: Summary of SEL Studies

followed with the development of theories, models, and guidelines applicable to software development. The final stages are the *packaging* studies with the production of tools, guidebooks, and training material to aid in the transfer of this technology to NASA and other organizations.

Later in this report we discuss three of the technologies studied by the SEL in greater detail (Ada, Object oriented technology, and Cleanroom). The details of transferring those technologies are summarized by Figure 4.2.

4.3 Use of Ada

Understanding phase of Technology Transfer. Use of Ada on flight dynamics projects was first considered in 1985. A training and sample program was the first Ada activity. However. to truly evaluate the appropriateness of Ada within the SEL environment, a parallel development of an Ada (GRODY) and FORTRAN simulator (GROSS) was undertaken. GROSS, as the operational product, had higher priority and was developed on time. GRODY, as an experiment to learn Ada, had a much longer development cycle. The experiences of the GRODY team with the typical set of requirements NASA used for such products led to a greater interest in object oriented technology described in Section 4.4. Although the development of this simulator continued until early 1988, by early 1987 it was decided that the initial project was sufficently successful to continue the investigation of Ada on other flight dynamics problems. Elapsed time since start of Ada activity was 30 months.

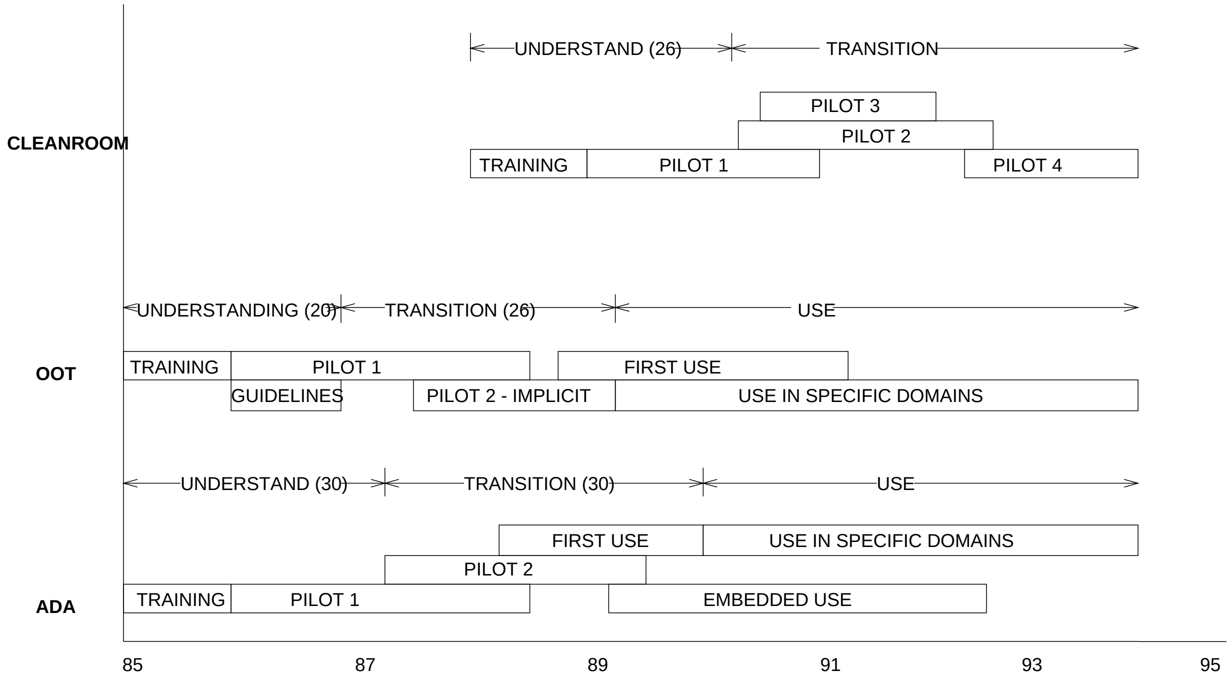


Figure 4.2: SEL Technology Transfer Experience

Experiences at Langley Research Center were similar to those of the SEL, but had a different conclusion. Understanding Ada began under the advocacy of one individual. A project was developed in both FORTRAN and Ada. Although the Ada project was deemed better than the FORTRAN version, the difference was not deemed great enough to enforce Ada on all projects.

Transition phase of technology Transfer. Because of the poor performance on the GRODY simulator and the problems with developing Ada requirements, the SEL undertook a second Ada pilot project (GOADA) as an experiment. However, there was sufficient confidence in Ada by this time to make GOADA an operational product, thus schedules and performance were more critical than with the previous GRODY experiment. In this case, the resulting product was comparable to performance of previous FORTRAN simulators. Between 1988 and 1990, four other simulators (one dynamic and three telemetry) were built successfully. In addition, one embedded application was developed beginning in 1989 and was not as successful due to the poor quality of the embedded compilers that were available. By early 1990, Ada became the language of choice for simulators in the Flight Dynamics Division. Transition time was another 30 months.

Comments on Technology Transfer. Local advocacy was crucial for starting to look at Ada as a development language. Total transfer time for Ada was approximately 60 months. Ada is the language of choice for simulator projects. Although Ada code costs more, line by line, than FORTRAN code, the higher levels of reuse result in lower overall delivery costs for such projects.

Ada was also considered as the implementation language on large mission ground support systems, but this task was never carried out. The inhibitors in this case were outside of the scope of the Ada language, itself. The operational systems at GSFC are IBM mainframe compatible, and no effective Ada compiler existed for this environment during the three times Ada was evaluated. All of the successful simulator projects were implemented on DEC VAX computers, which had an effective Ada system.

Presently, Ada is used on approximately 15 per cent of the SEL's software. Eleven operational Ada projects have been completed to date. A more complete analysis of the SEL's experiences with Ada are presented in [2].

4.4 Object Oriented Technology

Understanding phase of Technology Transfer. Use of object oriented technology (OOT) in the SEL was seriously investigated at the same time as Ada was considered. In developing the requirements for GRODY, mentioned in the previous section on use of Ada, it became apparent that the standard GSFC requirements document was oriented towards a FORTRAN functional decomposition and the use of these requirements on an Ada project would be very inefficient.

Object orientation seems more natural an approach with the use of Ada packages and generic functions. Therefore the requirements were rewritten to use a more object oriented approach. Following this, an OOT guidebook for GSFC was developed (GOOD - General Object Oriented Software Development [16]) for use on future projects.

Elapsed time for these activities took from early 1985 until August, 1986, or a total of 20 months. Expenses for understanding this technology were high since this activity was wrapped up in the Ada evaluation which required parallel system development of GRODY with the FORTRAN equivalent GROSS.

Transition phase of technology Transfer. On a second project (UARSAGSS), object oriented design was used implicitly. This was a FORTRAN ground support system, and experiences gained from the earlier GRODY effort allowed the programmers to better understand the design and use OOT. By the end of this project, it was sufficiently clear that OOT was an effective technique in some domains. Transition time was on the order of 26 months.

Comments on Technology Transfer. Total transfer time in this case was only 46 months. Although almost four years, this was relatively short since it did not require major changes in system development. The same set of tools could be used; object oriented technology was mostly a change in approach towards system building that could be used with any underlying implementation language. Although initially considered as an Ada technique, the same methods would map easily to a FORTRAN development model. Since it fit within the usual development paradigm, tailoring the method and inserting it into the usual NASA development process was relatively easy.

It should also be mentioned, that although object oriented technology was originally studied within the Ada domain, it has had a profound effect on productivity on FORTRAN projects. This is an additional reason why adoption of Ada as the development language has not been overwhelming within the SEL. Although overall productivity using Ada has greatly improved over FORTRAN productivity of the mid-1980s, FORTRAN productivity has also improved dramatically.

For example, productivity measurements in statements produced per hour of effort between Ada and FORTRAN code shows that FORTRAN is still easier to write and is easier to reuse than Ada code (This data was originally presented in [3]):

Statements	**Ada Stmts/hr**	**FORTRAN Stmts/hr**
New statements	0.9	1.1
Reused statements	4.7	9.0

Reuse has proven to provide the largest boost in productivity with both Ada and FORTRAN due to the effects of object oriented technology:

Language	1988-90 % Reuse	1990-1994 % Reuse
Ada	4-17 (3 projects)	64-88 (4 projects)
FORTRAN	4-12 (7 projects)	75-90 (4 projects)

Although not every project achieves such high levels of reuse, the trend is certainly upwards. This shows the seredipity nature of technology transfer; you generally cannot dictate exactly what you want to change, and change may occur from unexpected directions.

4.5 Use of Cleanroom

Understanding phase of Technology Transfer. Cleanroom is a software development process that emphasizes the correct design and development of source programs using formal verification methods rather than traditional techniques which emphasize testing and debugging for discovering programming errors. Programmers design modules and must develop verification proofs (either formally or informally) before submitting the module for integration into the larger system. Only integration testing of the entire system is performed.

Surprisingly, by eliminating the unit test phase, integration errors have been found to decrease, not increase. The process of forcing programmers to understand and validate their modules without the use of testing software makes them more careful about the assumptions they are making in their programs and hence, lowers the overall error rate. While additional resources are needed for the increased design time required for program verification, it has been shown that this is more than made up by decreased system testing and improved reliability of the final product.

In order to understand Cleanroom and how it could fit into the NASA development process, a series of training courses was given in 1988 by Dr. Harlan Mills, original developer of the method. A pilot project was undertaken and proved to be very successful. All participants were converts to the method, even though several had reservations about it before they began. Time to understand the method (training until the start of the second Cleanroom project) was 26 months.

Transition phase of technology Transfer. Two follow-on Cleanroom projects were undertaken. A smaller in-house development project was very successful, but a larger contracted project was not so successful, resulting in lower productivity. It was not as apparent whether problems on the larger project were due to scaling up of Cleanroom to larger tasks or to a lack of training and motivation of the development team on this project.

Because of this, a fourth Cleanroom project was undertaken that is still under development. The general conclusion seems to be that *within NASA/GSFC*, Cleanroom is very successful on smaller projects, but only marginally effective on larger projects.

Comments on Technology Transfer. Aspects of Cleanroom appear to be quite effective. Inspections, independent tests, and small builds of subsystems seem ready for incorporation into the SEL development model. The overall Cleanroom model, however, is still not ready for regular use. Understand time was 26 months and transition time is at least 46 months, with transition still underway.

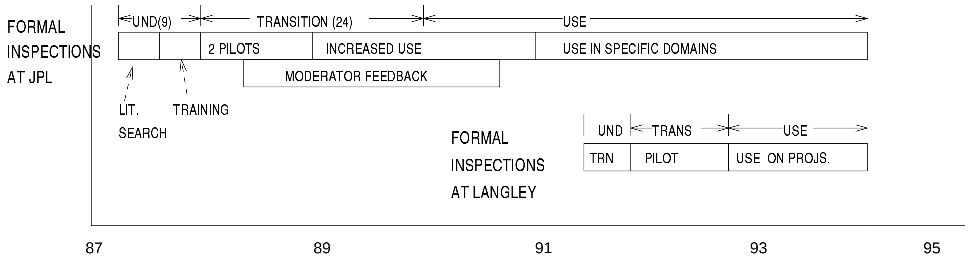


Figure 4.3: Formal Inspections Technology Transfer Experience

4.6 Use of Formal Inspections

The development of formal inspections at the Jet Propulsion Laboratory and its transfer to Langley Research Center is one of the rare technologies that has been successfully transferred to multiple NASA centers. This is summarized by Figure 4.3. We discuss here the development of the technology at JPL and comment on its exportation to NASA/LaRC in Section 5.1.

Understanding Phase of Technology Transfer. Initial work at the Jet Propulsion Laboratory (JPL) on formal inspections began in 1987 in response to a need for higher quality software. After studying the literature, a decision was made in mid-1987 to tailor inspections for JPL use. A course was developed and taught for the first time in February, 1988 at JPL. The initial advocates for inspections (John Kelly and Marilyn Bush) sought other JPL managers who would use and benefit from the technology.

Transition phase of Technology Transfer. Approximately two or three pilot projects were started in 1988 using inspections, with the first inspection being held on March 1, 1988. Bimonthly moderator meetings were held to spread the advocacy from the initial developers to other managers at JPL to have others “own” the process should the developers leave the organization. From 1989 through 1991 additional projects used the technology to help institutionalize the process. By late 1989 it was rapidly becoming a standard technology at JPL, and is now used regularly at JPL.

Comments on Technology Transfer. The elapsed time for developing the method was about 33 months and involved about three staff years of effort. Contacts with Mike Fagan of IBM, developer of the technique, aided the transition process. It has been successfully transferred to JPL and between its initial use in February, 1987 through 1990, over 200 inspections were carried out. The use of moderator meetings greatly aided the “people mover” model as other managers became advocates of the technology to help infuse it at JPL.

Chapter 5. Exporting of Technology

NASA has long recognized its role in exporting technology to industry and as given earlier, most NASA mechanisms are oriented around the technology exportation process. However, NASA was organized to develop space technology with the result that most new research has “space” as its overall theme. Therefore, within the software engineering domain, examples of successfully transferred technologies are relatively rare. The following sections give a few instances of these.

5.1 Transfer of Formal Inspections

Beginning in 1989 Kelly of JPL tried to move formal inspections from initial use at JPL to other NASA centers. With management at Kennedy Space Center, no advocate for the technology was found and the infusion process never took hold. The need for a gatekeeper role was quite apparent at KSC.

To avoid this problem at other centers, the transfer organization first tried to identify the appropriate gatekeeper who would promote the infusion process. At Langley Research Center, a course was offered and an advocate was identified who helped in the infusion process. In May of 1991 the initial Formal Inspection course was offered, and by Fall, 1991 a pilot project was started. In August of 1992, inspections were declared a standard part of all developments. This took only 16 months, because of the previous experiences at JPL. About 12 staff months of effort were required, but most of this effort was in “unpaid overtime.” No NASA support was available for developing the technology. As of July, 1994, 67 inspections have been held. Once installed at Langley, it has been transferred to several contractors working at Langley.

To avoid the initial problems at Kennedy Space Center and to duplicate the successes at LaRC, a NASA software inspections standard and an inspection guidebook were developed in 1993 and courses have been held at Ames Research Center, Johnson Space Center, Lewis Research Center, Marshall Space Flight Center and Kennedy Space Center. Outside of JPL, almost 400 individuals have been trained in the technique by the summer of 1994.

Comments on Technology Transfer. Formal inspections were successfully transferred at JPL and Langley. Total time for transferring at both centers were 33 months and 16 months. These times were relatively short since formal inspections cover only a relatively narrow and precise process in software development and can be inserted relatively easily into almost any mature development process. On the other hand, the process was not successful at other centers where no advocate was found for the technology. Use of established courses for identifying gatekeepers is proving more successful at other NASA centers.

The process was driven by the gatekeeper model. Advocates were first found at JPL and later at LaRC, who instituted the process at their center. Inspections were then imposed upon projects being developed within those centers. This included contractor personnel. Later, new gatekeepers in other non-space divisions of those contractors saw the value of inspections and started to require them on other non-NASA projects.

5.2 Experience Factory

Activities of the SEL have already been addressed with respect to technology infusion (Section 4.2). However, once a technology is deemed appropriate, the SEL has an important role in transferring that technology to other NASA groups and to external organizations.

Technology exportation in the SEL is based upon a three step process (as given earlier in Figure 4.1):

1. The SEL must first *understand* the process to be studied. This often requires small case studies performed at the University of Maryland, larger evaluations using programmers on NASA projects, or full investigation using a large software project development within Code 552 of NASA/GSFC.
2. The SEL must *assess* the impact the process has on the product being developed. This is generally accomplished by an analysis of the data collected as part of product development. Experimentation and measurement have been cornerstones in the SEL process since its beginings.
3. The new technology must be *packaged* via tailoring and modifications to make it available for use by NASA and other organizations.

The SEL has been an effective voice in transferring technologies to other organizations, although it has been more successful in transferring technology to organizations external to NASA/GSFC than to other groups within NASA. This was recognized this past May, when it was announced that the SEL was the first winner of the Software Engineering Institute – IEEE Computer Society International Software Process Achievement Award. The SEL was deemed the best organization for process improvement over a large selection of “Fortune 500” companies.

The importance of measurement in the software development process grew out of early SEL studies beginning in the late 1970s. During the 1980s, concepts like the Goal/Question/Metrics model were developed and adopted by many organizations. More recently, the Experience Factory has been developed and several organizations have been to the SEL to study its structure in attempts to infuse this model into their own organization.

Why has the SEL been effective at transferring technology to other organizations? Major components of the SEL that address technology exportation include:

1. **Publication of results.** SEL results have been widely distributed; several hundred technical reports, conference and workshop papers and journal articles have been published since 1977. This lends credibility to new results from the group. This is an effective example of the communication model of technology transfer. Three documents, in particular, have received wide circulation both within and outside of the agency. Each summarizes one aspect of the SEL’s approach towards software development:
    - L. Landis, S. Waligora, F. McGarry et al., *Recommended Approach to Software Development (Revision 3)*, SEL-81-305, June, 1992, 226 pages.
    - L. Landis, F. McGarry, S. Waligora, et al., *Manager’s Handbook for Software Development (Revision 1)*, SEL-84-101, November, 1990, 91 pages.
    - *Software Measurement Guidebook*, SEL-94-002, July, 1994, 148 pages.
2. **Operational software being studied.** The SEL is also responsible for operational NASA software, which lends credibility to the results of SEL studies. Software on the order of 500K lines of code

are studied, which move the results out of the realm of “toy” university programs. Also, by having professional programmers interact with SEL personnel, new “converts” are obtained to foster the people mover technology transfer model.

3. **Partnership structure.** As a joint activity of NASA, Computer Sciences Corporation and the University of Maryland, the SEL provides a proper mix of research and commercial experience to address realistic problems. It is the joint partnership which lends its effectiveness to the results. The approach has been adopted by other NASA research activities, such as RICIS at the UHCL (Section 3.3.3).
4. **Personal contact.** SEL personnel visit other organizations to discuss relevant technology transfer issues. In addition, corporations frequently visit the SEL to study its organization and processes. This is a great aid in the people mover model of transfer.
5. **Annual workshop.** The SEL has organized an annual Software Engineering Workshop, held at NASA/GSFC each December. With over 500 attendees, the workshop provides an effective way for individuals from other organizations to know SEL personnel and SEL results. This again fosters the people mover concept as personnal contect is the most effective method to build confidence in a new technique.
6. **Following established scientific principles.** It is perhaps unfortunate to have to list this in a scientific report, but the SEL is one of the few software engineering research groups that has modest goals and simply reports on those goals. Data is collected to validate success (and failure at times) on what was proposed. All too often within the software engineering field a new software technology is proposed as the “silver bullet” solution to good technology, only to fail to achieve all of its promises. The history of software engineering is littered with failed promises (e.g., structured programming, verification, Ada, object oriented design, 4GLs, requirements languages) [1]. Not that these are bad techniques – all are effective at improving productivity. However, the extremely high level of “hype” given to all of them when they are first proposed dooms them to be big disappointments. The SEL has always resorted to small incremental steps of modest improvements. For example, the current example of a silver bullet is the CMM. Productivity improvements of 80% to 100% are being touted by the SEI from a few anecdotal projects. Even if the CMM proves to improve productivity by 25%, it will probably be viewed as a failure given the up-front promises that are being made. However, a $250K savings on a $1M project is still a major accomplishment. The CMM would probably be more successful if productivity gains of 10% to 20% were made, and data were collected to prove those results.

5.3 TAE User Interface

TAE (Transportable Application Environment) probably represents NASA’s major success in exporting software development products outside of NASA. TAE is a graphical user interface useful as a front-end for other software products. It was developed during the period from 1981 through 1990 at GSFC, in a group independent from the SEL. Century Computing did much of the development work for NASA.

TAE development as a producer. Originally designed in 1981 (See Figure 5.1) to support ASCII terminals, this product was renamed TAE Classic when a more enhanced X-Windows and Motif compatible version (TAE Plus) was developed in 1986. TAE is distributed through the NASA software library COSMIC and over 900 licenses have been obtained for the prodct. It is one of the few software engineering technologies that has been transferred via the On-the-shelf model of technology transfer, although the communication model was the primary vehicle for “getting the word out.”

TAE is one rare example of a large-scale software engineering development within NASA. Over $3.5