NIST GCR 02-834

Benefits and Costs of ATP Investments in Component-Based Software

November 2002

National Institute of Standards and Technology • Technology Administration • U.S. Department of Commerce

NIST GCR 02-834

Benefits and Costs of ATP Investments In Component-Based Software

Prepared for
Economic Assessment Office
Advanced Technology Program
National Institute of Standards and Technology
Gaithersburg MD 20899-4710

By
William J White, Ph.D.
Michael P. Gallaher, Ph.D.
Research Triangle Institute
Research Triangle Park, NC 27709

Grant 50 SBNB6C9323

November 2002

Abstract

This report assesses the economic impact of the Advanced Technology Program's (ATP's) focused program in Component-Based Software Development. From 1994 to 2000, ATP provided a total of $42 million in public funds to support 24 separate technology development projects in this emerging field. Quantitative and qualitative analyses presented in this report demonstrate that ATP's support created significant economic benefits.

Two-thirds of the funded projects achieved their technical objectives; many of the firms and one joint venture proceeded to release successful commercial products based on the technologies developed. ATP was credited by many of these firms with *enabling* their R&D efforts, *accelerating* the technology development process, and *increasing the probability* of technical and commercial success.

When the 24 funded projects are viewed as an investment portfolio, the social returns exceed any reasonable benchmarks for public or private investment. The estimated 80 percent internal (social) rate of return reflects a substantial benefit to the nation in excess of the return to the companies funded and is an indication of inefficient capital markets for projects with high technical risks. The calculated net present value (NPV) of $840 million and benefit to cost ratio (B/C) of 10.5 suggest that the funds were a worthwhile expenditure of public funds.

Acknowledgments

The authors would like to acknowledge the assistance of many on the ATP staff, especially our primary contacts Jeanne Powell and Barbara Cuthill. Jeanne provided RTI with the project documents needed for our study, reviewed deliverables for accuracy and readability, and coordinated this report's internal review at ATP. Barbara contributed much of the technical background for the study, as well as a wealth of information about the individual firms and their software development projects. We would also like to thank Kurt Wallnau, from the Software Engineering Institute at Carnegie Mellon University, who reviewed a draft version of this report and provided many helpful comments.

In conducting the eight in-depth case studies, we relied heavily on structured interviews with the principal investigators from the funded firms. We owe a great deal of thanks to Gary Falacara from Aesthetic Solutions, Robert Glushko of Commerce One, Barbara Hayes-Roth from Extempo Systems, Jeff Rees at Hewlett-Packard (formerly at Intermetrics), Stanley Schneider of Real-Time Innovations, Elaine Kant from SciComp, Brendan Madden at Tom Sawyer Software, and Gregor Kiczales of Xerox.

Finally, we would like to thank Taylor Bingham of RTI for his leadership and guidance in the initial stages of this study. Brian Kropp and Michelle Bullock provided valuable research and analytical support.

Executive Summary

Component-based software development (CBSD) is a relatively new software production paradigm that focuses on building large software systems by assembling readily available components. Historically, about 85 percent of all large software projects have been customized applications used within a single firm, involving a minimal amount of code reuse. Systems programmers would employ proprietary or shared libraries for a minor portion (up to 20 percent) of the new application's code, but the remainder would be written specifically for the project at hand. The remaining 15 percent of programs, sold as packaged software to the retail market, have increasingly used embedded components, but even in these programs, the overall levels of reuse are quite low.

ATP's focused program in component-based software development is an effort to change the paradigm of custom application development to a "buy, don't build" approach for most, if not all, software projects. The basic concept is that computer code should be reused rather than rewritten whenever possible. Greater reuse is expected to produce the following benefits:

- reduce the cost of developing and maintaining software systems,
- increase the reliability of software, and
- yield greater synergies across portions of software code and applications.

The goal of ATP's focused program in component-based software was to develop the technologies that enable reuse of software code and automation of software development. ATP focused-program competitions held in 1994, 1995, and 1997 resulted in the selection of 24 software development projects. Sixteen of the technologies developed during these projects are currently being used in commercial applications.

This study uses case-study methodology to analyze ATP's investment in the portfolio of projects funded by the ATP's component-based software focused program. Research Triangle Institute (RTI) performed an evaluation of the economic impact of ATP's investment in this focused program, conducting eight in-depth case studies to support this estimation. The conclusions from the study are that the focused program was highly successful from a social perspective, yielding benefits of $1.5 billion on a combined public and private investment of about $119 million. In addition, the study found a number of qualitative benefits related to assisting firms in strengthening their planning and management functions and enhancing the credibility of the mostly small software firms that were funded.

Case studies are an important part of ATP's economic analysis strategy. They provide an in-depth view of how ATP-funded technologies lead to economic benefits for the awardees, other companies, and consumers. Case studies also provide qualitative details about how ATP funding affects the investment decisions of companies and the success of projects. Ideally, case studies apply state-of-the-art methodologies that provide credible quantitative estimates of the economic outcomes of ATP's investments in these technologies.

ES.2 COMPONENT-BASED SOFTWARE

Understanding the component-based market and the role of ATP within the market is key to developing an estimation strategy to measure the qualitative and quantitative benefits of the ATP focused program in component-based software.

ES.2.1 The Component-Based Software Market

A component is an independent piece of software that interacts with other components in a well-defined manner to accomplish a specific task. Components can be used to build both custom enterprise-critical software and "off-the-shelf," shrink-wrapped software. Although the ATP program focused on particularly challenging and complex large-scale software development for commercial and industrial applications, benefits from the new innovations will spill over to small-scale or simple applications. As a software component market emerges, most software developers are expected to shift focus from design and implementation processes in large-scale and enterprise-critical situations to solving application problems through adaptation of existing components or combinations of components.

The concept of using components to assemble software systems has been in existence for several decades. Firms have reused up to 20 percent of their code internally since software was first developed. However, several market failures have slowed the development of a component-based market, including lack of interoperability, network externalities, lack of trust in externally developed code, and lock-in effects. ATP's focused program in component software was designed to overcome the technical challenges associated with these barriers to success.

Interoperability

Interoperability is a measure of how well different pieces of technology function together. For example, one application might generate results that another application can use. The lack of interoperability slows the development of a component-based market. ATP addressed this problem by selecting projects for funding that focused on component technologies with multiple applications. ATP also funded projects that focused on enterprise-wide applications, where the technical challenge of maximizing interoperability was a goal.

Network Externalities

Components must interact with other components and infrastructures to create a valuable software product. For example, a new spell-checking component has no value unless basic word-processing components exist that can interact with the new component. As a network of interacting components is developed, the initial spell-checking component becomes more valuable. This concept is called a "network externality." Each individual component is more valuable than it is perceived to be because it adds value to all other components.

However, when an individual firm is making market decisions about what research to pursue for specific products, the firm does not take into account the network benefits that are being created because it cannot capture these benefits as profits. This divergence between the social and private values of a new component has slowed the investment in technology development needed to spawn a market of component-based products. By providing federal funding to cost-share high-risk R&D in this area, ATP can compensate for the discrepancy between the social and private returns from a new component, and thereby help stimulate development of reusable components and their commerce.

Lock-In

Legacy software and large, highly customized software systems are abundant in large firms. These systems contain massive amounts of information about how the company operates and identify critical data that must be collected and analyzed. Significant resources are spent on maintaining and updating these systems and monitoring their performance. Migrating a complex, monolithic system to a newer component-based system, in which updates and improvements are handled quickly and efficiently, could lower maintenance costs. However, firms have been reluctant to change to component-based systems. The main explanation is that the short-run costs of migrating from the existing system to a component-based system are prohibitively high, even though the long-term benefits could be substantial. ATP's funding of the risky and complex aspects of component-based technology development was anticipated to enable a multitude of component software products. As more companies develop software component products, more firms are expected to adopt component-based systems for their economic benefits.

ES.2.2 The Role of ATP

ATP's funding of the risky phases of technology development, while products are still many years away, gives companies a chance to pursue ideas that the private sector either will not fund or not fund in a timely manner. As a result, new products emerge that would not have been possible without ATP.

In 1994, ATP held a series of workshops with industry and academia to discuss the potential for a focused program on component-based software. These workshops, along with various white papers from the technical community, resulted in the ATP recommendation for a focused program. Another workshop was held in 1995 to update the program scope. Over the course of three rounds of funding, ATP committed a total of nearly $70 million and industry committed an additional $55 million of cost-sharing funds to a total of 24 projects. Of the 24 projects, 18 were completed by the time of this study, two are still under way, and four failed to complete. Two-thirds have yielded commercial products. To date, three-fourths of the projects have reached the commercialization phase even though some of the resulting products are not yet generating revenues.

Historically, the process of developing and using software components has been extremely complex. A number of barriers have created substantial technical risk to innovation in this area, including a lack of the following elements:

- ➤ automated tools for building, locating, and adapting components;
- ➤ interfaces for nonprogrammers to enable them to use the components; and
- ➤ specification of interface semantics for bridging applications.

The ATP's focused program for component-based software development sought to support development of such tools, with special emphasis on tools that enable complex, large-scale, commercial applications that can be used by a wide spectrum of industries. ATP focused its funding efforts in two separate directions: building components and building component architecture. Developing the basic building blocks for components and architectures that apply to all component developers and users is likely necessary to the maturation of component-based software. Once component-based software markets have emerged, software developers are expected to specialize and focus on specific applications of components. The component-based software industry would supply automated high-performance software tools and architectural services that could be incorporated into the development of computer systems. This model will allow a programmer to focus on the problem at hand without being concerned about syntax or the programming process.

ES.3 BENEFITS OF ATP INVOLVEMENT

ATP investments across a number of technology areas helped to eliminate or reduce market failures to develop risky new technologies in a variety of ways. ATP maintains a Web site with information to help participants conceptualize and write proposals, complete required audits and reports, and, through its R&D Alliance Network, establish successful cooperative research efforts. ATP sponsors a number of seminars across the country in which funded firms can participate.

Among the benefits of ATP participation hypothesized at the focused program's inception, RTI found the strongest evidence for the following impacts:

- accelerating the technology's development and adoption,
- increasing the likelihood of the technical success of each funded project, and
- widening the applications of ATP-funded technologies.

For the component software study, RTI developed an economic model to estimate the benefits attributable to ATP-supported component-based software R&D projects, most of which resulted in the creation of new software prototypes. Because the final products will have a wide variety of potential uses and customers, data collection for a price-index type analysis of benefits to users of the technology would have been difficult or impossible to accomplish. However, RTI developed a simple model that allowed an estimate of consumer and producer surplus directly from data provided by the target companies. Although this type of economic analysis is theoretically straightforward, this is the first ATP evaluation in which it has been used.

ES.3.1 Economic Modeling Methodology

The firms engaged in software R&D are unlike companies typically envisioned in basic economics theory in many respects. Although they sell final products in competitive markets, their ability to differentiate their products gives them a good deal of market power in new, highly differentiated product areas and thus the ability to raise prices above marginal and average cost. Because of the nature of the R&D process, however, these firms must commit significant funds far in advance of achieving either technical or commercial feasibility. R&D costs become the equivalent of high fixed costs. These precommercial, high fixed costs, along with low marginal costs of reproducing software and highly differentiated products, produce many of the features of natural monopoly markets. However, the rapid pace of technological development in software limits the scope and duration of the market power the innovating firms exert, as expanded product markets and competitor companies inevitably emerge around the new idea and product.

Many of the companies involved in component-based software are small start-ups that face severe financing constraints. Because the bulk of their expenditures occur prior to earning any revenues, indeed before technical feasibility has even been established, these firms have difficulty obtaining capital from loans or equity participation. Funding by ATP or another source of patient capital is thus the only way that these technology development projects can be undertaken. In addition, these firms are concerned about the possibility of other firms entering the market, so they must price at a point that is high enough to cover the fixed costs of production, but low enough to slow the entry of other firms into the market. They will set prices only high enough to recover their fixed development costs over the entire product life cycle. In the detailed case studies, RTI estimated the demand curves, R&D costs, and marginal costs of production to determine the quantitative benefit of ATP-funded projects in component-based software. In addition, RTI assessed the extent to which other qualitative benefits occurred as a result of these projects.

ES.3.2 Estimation of Economic Performance

RTI separated the economic benefits from ATP investment into quantitative and qualitative benefits and used a portfolio approach to calculate the quantitative returns to ATP-funded projects. RTI selected eight projects that were expected to generate significant economic returns, including firms of different sizes and at least one firm from each of the three funding rounds. Benefits were estimated from these eight successful projects and compared to the costs for the entire set of 24 projects. This is equivalent to assuming that the remaining 16 projects generated zero economic benefit, an extremely conservative approach.

From each of the eight in-depth case studies, estimates of the total benefits attributable to the ATP project were made. This stream of benefits was then compared to total outlays by ATP and industry to generate portfolio measures of performance. Table ES-1 shows several measures of economic performance attributable to the ATP program in component software.

Table ES-1. Quantitative Measures of Economic Performance

Net Present Value (in 2000 dollars)	$840 million
Benefit-Cost Ratio	10.5
Internal Rate of Return	80%

The ATP component-based software focused program was successful. The benefit-cost ratio for the entire portfolio of CBSD projects was projected to be 10.5; that is, with cash flows adjusted to 2000 dollars, a return of $10.5 was projected for every dollar of ATP and industry investment. The net present value of the investment, projected to be $840 million, describes the net total benefit to the nation, in 2000 dollars, based on a 7 percent, OMB-designated discount rate. The internal (social) rate of return on investment of 80 percent describes the projected rate of return to the nation.

These measures provide a conservative estimate of the net benefits expected from the technologies that ATP funded because they include benefits from just eight of the projects but costs of all 24; the measures also assume very limited life spans of resulting products relative to their likely potential. Most of the products for which benefits were estimated have already generated some revenues to the ATP-funded companies and benefits to customer-users. Three of the projects—the Commerce One JV, and the Tom Sawyer and Intermetrics single-company projects—generated enough returns independently to cover the entire cost of the focused program.

In addition to the benefits discussed above, the funded firms reported that ATP made several contributions to their success that are more difficult to quantify. Standardized ATP requirements for proposal writing, record keeping, and progress reporting motivated firms to be more thorough in their project planning and execution. The ATP funding requirements and management environment were well suited to the unpredictable nature of R&D, as both patience and flexibility on the part of ATP supported the firms' efforts. The participation of ATP also imparted a "halo effect" of increased credibility, and lowered barriers to commercialization of the technologies developed.

ES.3.3 Summary

The quantitative and qualitative analyses presented in this report demonstrate the significant impact of ATP's investment in component-based software technology. Two-thirds of the funded projects achieved their technical objectives; many of these firms and one joint venture created successful commercial products based on the technologies developed during their ATP project. When the 24 funded projects are viewed as an investment portfolio, the 80 percent internal rate of return, $840 million net present value, and 10.5 benefit-to-cost ratio suggest that the focused program was a worthwhile expenditure of public funds.

Contents

Figures

Tables

1 Introduction

Technological progress is the key to offering future populations the potential for improved standards of living. Technical change enables firms to combine inputs in a novel manner to produce existing products more cheaply and to develop new products to meet consumer needs. Economists and other social scientists are in broad agreement that technological change is the most important contributor to economic growth in the modern era. Based on Robert Solow's and Moses Abramovitz's ground-breaking work more than 40 years ago, economists have estimated that more than half of the United States' long-run growth is attributable to technological change (Solow, 1957; Abramovitz, 1956).

Whenever an individual or a firm makes a technological advancement that improves the performance or quality of a product or reduces the cost of making it, the overall level of social well-being in the economy is increased. Likewise, when a new product or service is developed, society benefits as long as some consumers are willing to pay more than the costs of producing the product or service. Established principles of public economics argue that the private level of investment in such innovations will be optimal in the absence of market failures or externalities; that is, if the innovator is able to fully realize the benefits generated by the technology improvements through profits.

In most cases, however, a portion of these benefits "spills over" to consumers or to other economic agents (Mansfield et al., 1977; Scherer, 1999), because the innovating firm typically cannot extract all of the surplus created. If there is sufficient rivalry among producers, for example, prices may be driven down to the point that the innovating firm cannot retain any surplus and thus is unable to recover its investments in research, development, or purchase of long-lived assets. In other cases, benefits may accrue to competitors and firms in related or unrelated fields, in the form of knowledge or network spillovers.

The risk that innovators may not recover their investment lies behind our nation's patent and copyright protection systems. The promise of a limited monopoly offers firms and individuals assurance that they will be able to retain some of the surplus from their creations. If this intellectual property protection is sufficient to induce firms and individuals to pursue all socially beneficial innovations, private levels of investment will be optimal. Even if some improvements are not made because of the existence of spillovers, the losses from these marginal innovations may not be large enough to justify an extensive government role in research supporting product and process development.

1.1 GOVERNMENT'S ROLE IN SUPPORTING RESEARCH AND DEVELOPMENT

The situation for R&D aimed at producing or improving private goods and services is quite different from the creation of scientific and technological knowledge, the goal of most basic and applied research. In the latter cases, it becomes difficult or impossible for innovators to achieve the major portion of the benefit from their inventions. Standard public economics tells us that private markets will yield a suboptimal level of basic and applied research, leading to a lower than desirable level of technical progress.

To correct for this potential market failure, a large number of government organizations provide funding for research activities. These entities, including such giants as the National Science Foundation (NSF), the National Institutes of Health (NIH), and the Defense Advanced Research Projects Agency (DARPA), fund in-house research activities, university research programs, corporate projects, and joint ventures. Government support for technology infrastructure and standards development is provided by the National Institute of Standards and Technology (NIST) and related organizations.

NIST's Advanced Technology Program (ATP) was created in 1990 to promote the development of high-risk and high pay-off technologies where market failures or externalities are likely to lead to underinvestment by private firms. ATP provides funds, on a cost-sharing basis, for precommercial research and development of enabling technologies that support new products and process improvements where substantial spillovers are expected and where technical and investment recovery risks are both high. For example, ATP can provide funding for small firms that may not have access to other sources of capital due to lenders' risk evaluation processes. Banks and venture capital firms, which are primarily concerned with the private return on their investment rather than improvements in social welfare, often are unwilling or unable to properly evaluate the risks and potential returns involved.

1.2 EVALUATING THE EFFECTIVENESS OF ATP PROJECTS

Since its inception, ATP's Economic Assessment Office (EAO) has taken an active role in supporting evaluation of ATP-funded projects. To date, more than a dozen external assessments have been completed and shared with the public. These studies have measured the impact of the ATP on U.S. firms, industrial sectors, and the overall economy. The studies that ATP has conducted and funded include

- real-time evaluations of project progress, using ATP's project management teams and analysis of the data reported by the companies through the business reporting system on a confidential basis and used in statistical analyses;
- third-party surveys of participating companies to assess ATP's effect on the companies' decisions and success;
- project case studies that assess the costs and benefits of ATP's investments in specific technologies or technology areas;
- general studies of how ATP funding leads to spillover benefits to beneficiaries other than the ATP award recipients; and
- models that link large-scale macroeconomic models with microeconomic project analyses.

Case studies are an important part of ATP's economic analysis strategy. They provide an in-depth view of how ATP-funded technologies lead to economic benefits for the awardees, other companies, and consumers. They also provide qualitative details about how ATP funding affects the investment decisions of companies and the success of the projects. Ideally, case studies apply state-of-the-art methodologies that provide credible quantitative estimates of the economic performance of ATP's investments in these technologies.

The economic methodologies and tools used by the external assessors have varied widely in these case studies, depending on the types of activities funded, outcomes of the public investment involved, and EAO's evaluation needs. For example, for projects yielding process improvements that raise finished product quality at a somewhat higher unit cost, a macroeconomic analysis would capture the net social benefits. Such an approach was used in two studies in the automotive sector (CONSAD, 1997; Ehlen, 1999). Mansfield et al. (1977) pioneered methods that evaluate the effects of new products and/or processes in reducing downstream production costs. Price-index concepts for measuring the value of performance improvements have been applied in a recent study of digital data storage technologies (Austin and Macauley, 2000).

1.3 RTI'S APPROACH TO THE CURRENT STUDY ON COMPONENT-BASED SOFTWARE

In the current assessment, Research Triangle Institute (RTI) was asked to evaluate the benefits from a number of new ATP-funded component-based software projects, most of which resulted in the creation of new software products. Because these products have a wide variety of potential uses and customers, data collection for a price-index analysis would have been difficult or impossible to accomplish. However, RTI's skills in primary data collection and case study methods allowed us to estimate consumer and producer surplus directly from data provided by the ATP-funded companies. Although this type of economic benefit estimation is theoretically straightforward, this is the first ATP evaluation in which it has been used.

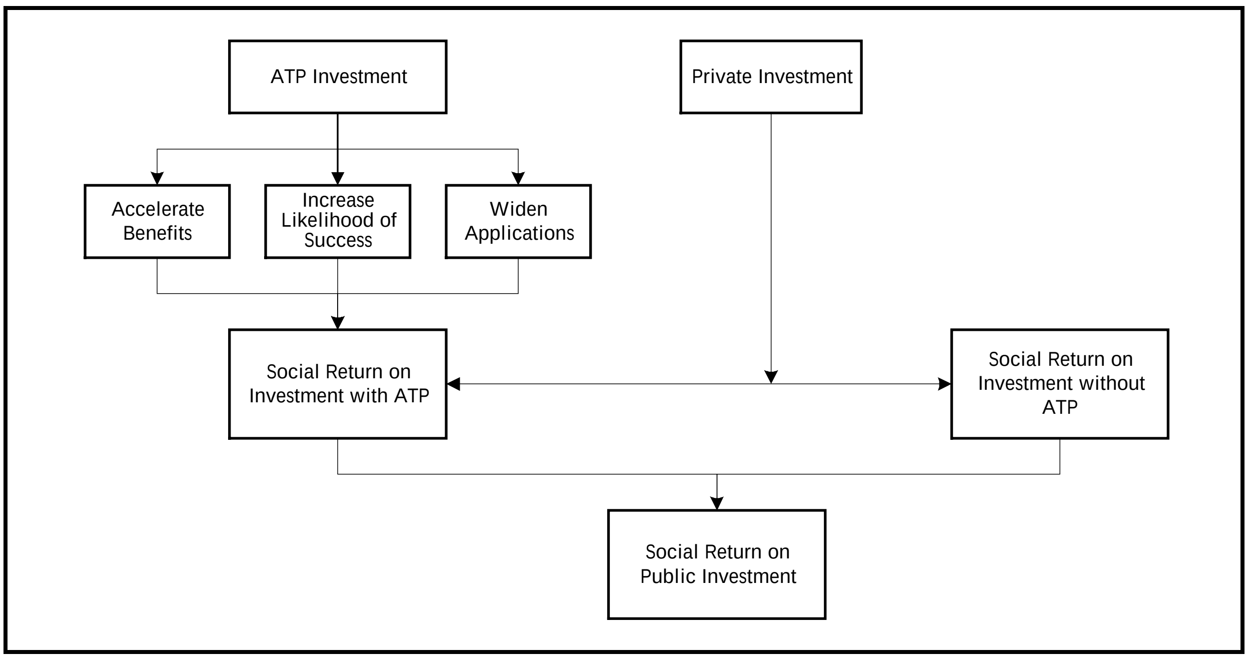
In completing this assessment, RTI refined and improved a case study methodology developed during a previous project conducted for ATP, "A Framework for Estimating the National Economic Benefits of ATP Funding of Medical Technologies" (Martin et al., 1998). That study demonstrated how ATP funding could increase the societal benefits of a technology by

- accelerating the technology's development and adoption,

- ➤ increasing the likelihood of the technical success of the project, and
- ➤ widening the applications of ATP-funded technologies.

Figure 1-1 demonstrates how these aspects of ATP investment lead to social return on the public's investment. For this project, we augmented the factors listed above by examining ATP's impact on the development of new high-risk ideas for component-based software technologies, its influence on the formation of firms and joint ventures, and its impact on the commercial success of these technologies

Figure 1-1. Elements Determining Social Return on Public Investment and Social Return on Investment



1.4 OUTLINE OF THE COMPONENT-BASED SOFTWARE DEVELOPMENT STUDY

Section 2 of this report provides a technical description of component-based software, profiles the relevant portions of the software industry, and outlines ATP's component-based software development (CBSD) initiative. Of the 24 projects initially accepted for funding by ATP, 17 had successfully completed their period of ATP funding in time to be included in this analysis. In the first phase of our assessment, these 17 projects were studied qualitatively to identify common themes and key ATP impact areas. Based on the characteristics of the funded firms, RTI selected eight for detailed case studies. The selection criteria appear as Appendix A in this report.

In Section 3, we present a detailed description of a firm's R&D decision-making process, including the separate phases of idea generation, technology and product development, and commercialization. Next, we examine potential externalities and market failures in the component-based software context, along with actions that ATP could take to eliminate or mitigate the adverse economic impacts. The section concludes with a qualitative discussion of our findings on ATP's impacts from analyzing the 17 funded and completed projects.

Section 4 contains an economic model based on natural monopoly markets, building on the qualitative results from the first phase of the project. We then discuss how this model was operationalized as a means to estimate the economic benefits from the ATP focused program, and describe the valuation metrics employed.

Section 5 presents the quantitative analysis of the eight projects selected for the in-depth case studies. The economic model developed in the previous section was used to estimate benefits from these projects, all of which were technically successful and most of which have been commercialized. With the expenditure data provided to us by ATP, we calculate the net present value (NPV), internal rate-of-return (IRR), and benefit-to-cost (B/C) ratio for each of the studied projects. Detailed write-ups of the eight case studies are included in Appendix B.

2 Overview of the CBSD Initiative

Component-based software development (CBSD) is a relatively new software production paradigm that focuses on building large software systems by assembling readily available components. Historically, about 85 percent of all large software projects have been customized applications used within a single firm, involving a minimal amount of code reuse (ATP, 1999). Systems programmers would employ proprietary or shared libraries for a minor portion (up to 20 percent) of the new application's code, but the remainder would be written specifically for the project at hand. The remaining 15 percent of programs, sold as packaged software to the retail market, have increasingly used embedded components, but even in these programs the overall levels of reuse are quite low.

CBSD changes this paradigm and follows a "buy, don't build" approach for most, if not all, software projects. The basic concept is that computer code should be reused rather than rewritten whenever possible. This approach can potentially be used to reduce software development time and costs, decrease the incidence of program errors, and reduce maintenance costs associated with software change.

There are other valid motivations for CBSD that may not require or even involve reuse. For example, component developers John Cheesman and John Daniels argue that adaptability to new and evolving functionality is far more important in enterprise system development than reuse (Cheesman and Daniels, 2000). The authors reason that the benefits of component technology may be more attributable to a design paradigm and a set of associated component standards. Still, this development method has been strongly endorsed by software architect Clemens Szyperski, who emphasizes the importance of component reuse and component markets (Szyperski, 1998). It is therefore important to understand that software reuse does not necessarily require software component technology, nor does software component technology focus only on software reuse.

In this section, we describe the historical approach to software development and how CBSD differs from and improves on existing techniques. We describe what a component is and what types of components are available to build software. We also discuss the important characteristics that software developers look for in a component. Finally, we describe ATP's efforts to support industry in influencing the emergence of the CBSD market and the CBSD supply chain.

2.1 HISTORICAL APPROACH TO SOFTWARE DEVELOPMENT

2.1.1 A Brief History of the Software Industry

The watershed event in the development of the software industry can be traced to 1969, when the U.S. Justice Department forced IBM to "unbundle" its software from the related hardware, and required that the firm sell or lease its software products. Prior to that time, nearly all operating system and applications software had been developed by hardware manufacturers, dominated by IBM, or by programmers in the organizations using hardware. Software developers in the 1950s and 1960s worked independently or in small teams to tackle specific tasks, resulting in customized one-of-a-kind products. Since this landmark government action in 1969, a software development market has emerged, and software developers and engineers have moved through several development paradigms (Egan, 1999).

During the 1970s, improvements in computing capabilities caused firms to expand their use of automated information-processing tasks, and the importance of programming to the activities of firms increased substantially. Simple tools to aid software development, such as programming languages and debugging tools, were introduced to increase the software programmer's productivity. The introduction of the personal computer and its widespread adoption after 1980 caused demand for software and programming to accelerate, rapidly outpacing previous productivity improvements. Processor speed, roughly doubling every 18 months, has dramatically outpaced the rate of improvement in software, creating a “software bottleneck” (Egan, 1998, 1999). Although software programs are easily duplicated and distributed, allowing for economies of scale, the customized approach to software development was so entrenched that economies of scale in programming never emerged.

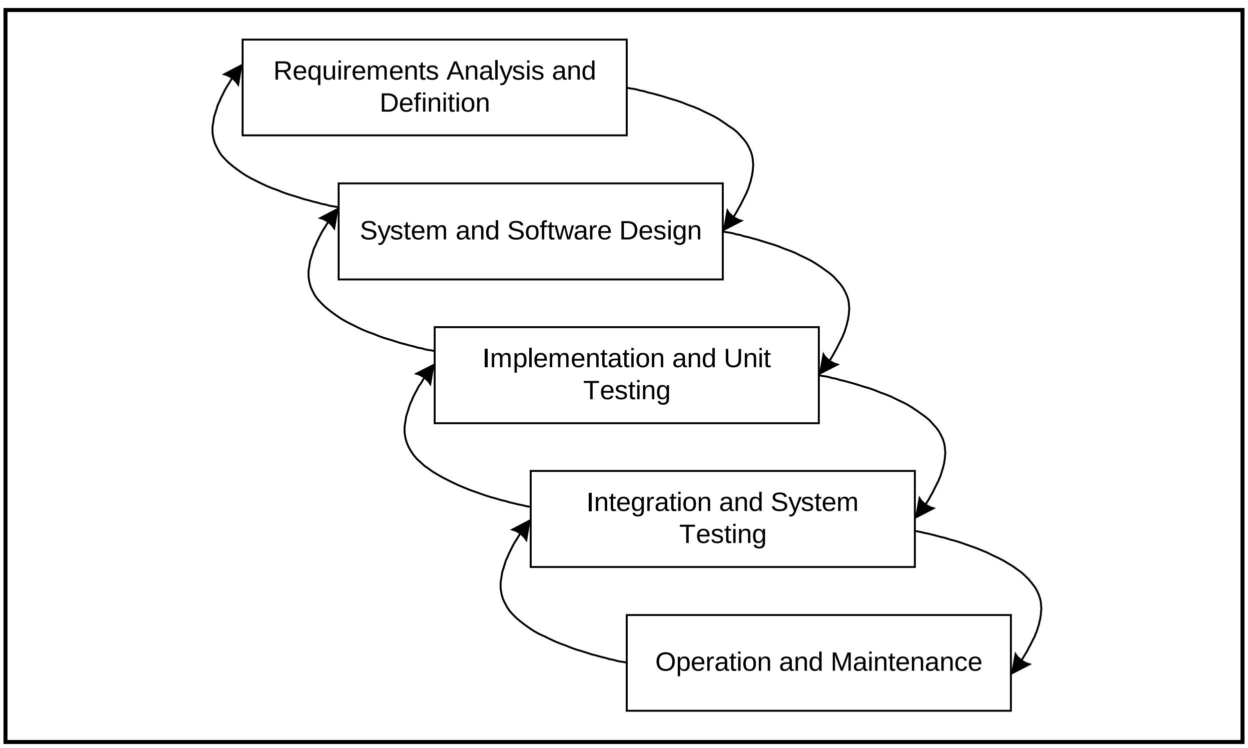
Recognition of the importance of the bottleneck, and its role in hindering progress in information technologies, has influenced many software development activities over the past 20 years. Object-oriented languages, which today include Smalltalk, C++, and Java, built on the concept of software modularization to create objects that are completely reusable. Although a significant step in increasing productivity, object technology by itself is not sufficient to reach optimal levels of reuse (Brown and Wallnau, 1998a).

The fact that a software product has been produced with an object technology does not ensure that it has plug-and-play capability, interoperability, and a standard interface, nor does it mean that the product will be sold in the market rather than being bundled into a larger monolithic product. The ATP focused program in CBSD and other noncommercial ventures, such as the COTS-Based Systems initiative (COTS stands for commercial off-the-shelf) at Carnegie Mellon’s Software Engineering Institute, are attempting to move software development technology to a full component basis, thus achieving the economic benefits of optimal reuse.

2.1.2 The Software Development Process

The historic approach to the software development process, focused on system specification and construction, is often based on the waterfall model (Andersson and Bergstrand, 1995). Figure 2-1 shows how this process separates software development into several distinct phases with minimal feedback loops. First, the requirements and problem are analyzed; then systems are designed to address the problem. Testing occurs in two stages; the program itself is tested and then how that program works with other programs is tested. Finally, normal system operation and maintenance take place. Feedback loops exist only between the current stage and its antecedent and the following stage. This model can be used in a component-based world for describing the separate activities needed in software development. For example, the requirements phase and the design phase can include the identification of available reusable software. However, in a component-based world, development does not exist in isolated boxes and steps. Rather, it is a process full of feedback loops across all stages of development. Feedback loops throughout the entire development process increase the ability to reuse components. Reuse is the key attribute in CBSD but does not exist in the waterfall model. When building a component-based program, developers need to examine the available products and how they will be integrated into not only the system they are developing, but also all other potential systems. Feedback loops exist throughout the process and each step is no longer an isolated event.

Figure 2-1. Waterfall Model



Adapted from Andersson and Bergstrand (1995), Table 2-1 illustrates where software developers have historically placed their efforts.[1]

Table 2-1. Allocation of Effort

<table>
<tr><th></th><th>Requirements Analysis</th><th>Preliminary Design</th><th>Detailed Design</th><th>Coding and Unit Testing</th><th>Integration and Test</th><th>Unit Test</th></tr>
<tr><td>1960s–1970s</td><td colspan="3">10%</td><td>80%</td><td colspan="2">10%</td></tr>
<tr><td>1970s–1980s</td><td colspan="2">20%</td><td colspan="2">60%</td><td colspan="2">20%</td></tr>
<tr><td>1990s–2000s</td><td>40%</td><td colspan="2">30%</td><td colspan="3">30%</td></tr>
</table>

Source: Andersson and Bergstrand (1995).

In the 1960s and 1970s, software development focused on writing code and testing specific lines of that code. Very little effort was spent on determining its fit within a larger system. Testing was seen as a necessary evil to prove to the final consumer that the product worked. Andersson and Bergstrand estimate that 80 percent of the effort put into early software development was devoted to coding and unit testing. This percentage has changed through time. Starting in the 1970s, software developers began to increase their efforts on preliminary design and requirements analysis, spending 20 percent of their effort in these phases.

Additionally, software developers started to invest more time and resources in integrating the different pieces of software and testing the software as a unit rather than as independent entities. As developers have moved toward CBSD, the allocation of effort has again changed. The amount of effort spent on determining the developmental requirements of a particular software solution has increased in importance. Forty percent of the software developer's effort is now spent in the requirements analysis phase. Design phases in a CBSD world are extremely important because these phases determine the component's reuse possibilities. Developers have increased the time spent in this phase to 30 percent, which reflects its importance.

[1]These data are based on the effort that software developers put into developing a product to release. Over time, the amount of effort spent post-release—primarily on maintenance and upgrade—has increased.

Based on work by Aoyama (1998), Table 2-2 compares the conventional approach to software development and CBSD. The architecture of a system, defined as how the system is set up, is described as *monolithic* (designed for a specific purpose) in the conventional approach. Monolithic software tends to be rigid and unable to be adapted for purposes other than those for which the software was originally designed. Even though a monolithic system typically contains modules, they are designed for use only within the specific architecture of one application. For each new program, most of the lines of computer code have to be written from scratch.

Table 2-2. Comparison of Conventional Approach to CBSD

Characteristics	Conventional	CBSD
Architecture—how the system is set up	Monolithic	Modular
Components—the pieces of the system	Implementation and white box[a]	Interface and black box[b]
Process—how the system is put together	Big bang and waterfall (analysis to design to implementation to testing)	Evolution and concurrent engineering (component development to component integration)
Methodology—how the system changes through time	Build from scratch	Composition
Organization—market for buying and selling components	None	Specialized—component vendors, brokers, and integrators

[a]When a "white box" component is used, the programmer is given access to the source code from the component. The programmer then can change the code, and adaptation is relatively easy.

[b]"Black box" components cannot be adapted or changed.

Source: Aoyama, 1998.

In a component-based system, the architecture for software production is described as *modular* because developers build programs from component pieces. From the initiation of the project idea to completion of the component, the developer must envision how the final program will fit together and how any new components will be made available to other developers and users; there is a feedback loop from the first stage to the final stage.

The way the information and lines of code are treated within a program is also significantly different. Developers in a conventional approach know each line of code that they are assembling, but in a component-based world, developers only know about the properties of the component; they do not know the specific lines of code within the component. This difference leads to problematic comparisons between the two approaches. The conventional approach measures programmer efficiency as the number of documented lines of code written. In CBSD, measures of efficiency can be based on the frequency of reuse (Aoyama, 1998).

The final major difference between the two methods of software development is the establishment of a market for the product. Formal markets do not exist in the conventional approach. If a firm needs to alter or update its computer network, it has to hire a developer to build on its existing system or write new code from scratch. For CBSD to be successful, markets must exist in which firms can purchase the architecture and components that they need.

Several specific types of firms have emerged in the CBSD market that do not exist in the conventional approach (Brown, 1999):

- component developers—develop the necessary infrastructure and components;
- component brokers—act as single point of contact connecting vendors to purchasers;
- component educators—have tutorials, certification for various infrastructures; and
- component-based consultants—offer mentoring services throughout the entire process.

The brokerage process has become increasingly advanced in recent years, with Web-based brokerage firms clearly taking the lead. Two sources, www.componentsource.com and www.flashpoint.com, provide a marketplace that connects buyers to vendors. Both of these firms provide extensive information on the quality, interoperability, and other attributes of every component sold.

2.2 DEFINITION OF COMPONENT

Determining what constitutes a component is one of the primary steps in determining the impact of ATP-funded projects. Gallaher, et al. (2000) have argued in a previous study that by identifying and defining an activity, a common set of terminology emerges within an industry. The emergence of this terminology can be an effective step towards eliminating market failures and expanding the size of markets. Meyers and Oberndorf (1997) point out that developing a common definition for a component and component characteristics is a necessary step in building a component-based software market.

Parrish et al. (1999) argue that a component is a software artifact that consists of three attributes. A component has a service interface that determines what services it provides, a client interface that states what services it uses, and the code that is necessary for the component to execute its commands. Ning (1999) offers a second definition. He states that a component is an encapsulated, distributable, and executable piece of software that provides and receives services through well-defined interfaces. Szyperski (1998) offers a similar definition by defining components as binary units of independent production, acquisition, and deployment that interact to form a functioning system. Meyer (1999) provides an additional characteristic that helps define a component: the component developers do not need to know the component's users. The common theme that emerges from these different definitions is that a component is an independent piece of software that interacts with other components in a well-defined manner to accomplish a specific task.

Components can be used to build both enterprise-critical software and shrink-wrapped software. While the ATP program focuses on the enabling technologies needed for complex and large-scale software development for commercial and industrial applications, benefits from the new innovations will spill over to small-scale or simple applications. As a software component market emerges, most software developers will shift focus from design and implementation processes in large-scale and enterprise-critical situations to solving application problems in small-scale purposes. Shrink-wrapped and small-scale software includes word-processing, spreadsheet, and other specific applications.

2.3 COMPONENT CHARACTERISTICS

CBSD replaces the notion of building a program by writing code with the idea of building a new application by assembling and integrating existing components into a new system. CBSD shifts the focus of system development to construction of components where the following characteristic activities become important: identification, qualification, adaptation, integration, and upgrade (Brown and Wallnau, 1996). Of these characteristics, *integrability* is the key difference compared to previous approaches to software development. Rather than focusing on how a particular piece of code will be implemented, the focus is on how the particular piece of code can be integrated into multiple programs.

The best approach to considering different components is to imagine a component as a bundle of particular goods. This bundle consists of the five characteristics given by Brown and Wallnau. Some components display a high level of integrability; others display better adaptability. In some cases, tradeoffs between characteristics may occur. For example, the greater the ability to adapt a component to different circumstances, the harder it is to qualify and test that component.

2.3.1 Identification

Before a building is constructed, the builder must know what pieces of brick and mortar are available to be used. A software developer writing a new program under a CBSD paradigm has the same concern. Knowing what components (building blocks) are available, the software developer designs programs at minimum cost and maximum efficiency.

2.3.2 Qualification and Suitability Testing

Because a component is initially designed to solve one particular type of problem, it needs to be qualified before it can be used in additional settings. Component qualification is the process of determining the “fitness for use” of previously developed components that are being applied in a new system. In other words, are different components available to address a particular programming need, and how well does each component fit that need?

Component qualification consists of two phases: discovery and evaluation. The component’s properties are identified in the discovery phase. For example, what does the component do or what standards does it meet? Following the discovery phase, the evaluation phase occurs. The evaluation phase focuses on quantifying the component. For example, how reliable is the component, or does it actually do what the component manufacturer says it does? Third parties (e.g., the International Standards Organization) have created criteria that software developers can use during this stage. Several software initiatives produced tools to test component qualification, including SAAM, PRISM, and PLAS (Haines et al., 1998). Continuing with the building analogy, the discovery phase is similar to deciding on which of the available tools, workers, and bricks should be used to construct a building. The evaluation phase would then correspond with checking the workers' references.

Failure to adequately foresee how a component might be used, and therefore failure to adequately specify a component can create potentially disastrous consequences. Jézéquel and Meyer (1997) describe how inadequate specification of a particular component in Ariane 5, a European Space Agency Mission vehicle, caused the $500 million project to fail. The problem was not due to the failure of a component, but rather to the lack of specification of the component's range of acceptable uses.

2.3.3 Component Adaptation

One component can rarely be plugged into multiple programs without any adaptation. When components are used in multiple systems, some changes must be made to fit each software environment. The changes must be made to components to minimize the amount of conflict among the components.

Component adaptation can be accomplished in three ways. Components can be written so they are white box, gray box, or black box in nature. When a "white box" component is used, the programmer is given access to the source code from the component. The programmer then can change the code, and adaptation is relatively easy. However, this approach also requires the most monitoring and maintenance because the developer might make changes to the source code of the component that cause numerous potential and unintended consequences.

The second approach is to use "gray box" components. These components are not supplied with their source code, but an extension language is provided that allows the programmer to incorporate the component into a program with minimal disruption.

The final approach to component adaptation is use of “black boxes,” components that are difficult to adapt or change. They can either be put into place to accomplish a predetermined and specific task, or they must be left out. No access to source code is given and no extensions are made available to incorporate a black-box component into other programs. Sometimes a predefined set of alternatives controllable through parameters may be provided in a black-box component.

2.3.4 Integration

In CBSD, components are assembled to form a system. Appropriate integration is necessary to ensure that the components work together in an efficient and desired manner. Where component adaptation focuses on how to change components to fit the need of the situation at hand, integration focuses on how to fit the components together in the entire program. Returning to the building analogy, adaptation is similar to deciding how to modify or alter each room, and integration can be thought of as the blueprints.

2.3.5 System Evolution/Upgrade

System evolution is the dynamic aspect of CBSD. When a component needs to be changed or updated or when additional features are desired, the ideal solution would be a “plug and play” approach. If the programmer could simply take out the old or defective component and replace it with the new one, system evolution would be simple. However, the new component rarely performs exactly the same function as the old component and rarely fits perfectly into the entire system. Rather, wrappers must be written to fit the new component into the program, and tests of the new component in isolation and in concert with the rest of the program must be performed (Kogut and Clemens, 1999).

System evolution is similar to remodeling the building. Rarely can one room be taken out of the building and a second room put in without disturbance to the building. Usually, electrical wiring and plumbing need to be changed and reworked for the renovated room to have access to electricity and water and to prevent disruption of those services to the rest of the building.

2.4 ATP INVOLVEMENT

ATP started funding component-based software development projects in 1994. Before this time, a component-based market was slow to develop for several specific reasons. Section 2.4.1 explains the market failures that were slowing the development of a component-based market. Section 2.4.2 describes the steps that ATP has taken to help alleviate these failures.

2.4.1 Rationale for ATP Involvement

The concept of using components to assemble software systems has been in existence for several decades. Firms have reused up to 20 percent of their code internally since software was first developed. However, several market failures have slowed the development of a component-based market, including lack of interoperability, network externalities, and lock-in effects. ATP's efforts in funding a focused program in component software were aimed at overcoming the technical challenges associated with these barriers.

Interoperability

Interoperability is a measure of how well different pieces of technology function together (Tassey, 1997). In the CBSD case, two types of interoperability exist. First, components must be interoperable with a given system architecture. Second, components must be able to interoperate with other components.

However, software companies are developing different architectures and different components. When there is no commonly agreed-upon platform or interface, software developers will not want to develop new components because they may not work with other components or with existing system architectures. A second problem occurs when there are multiple types of architectures in use. By increasing the number of architectures, the component development costs increase; software developers would have to produce specialized components for each architecture. This lack of interoperability slows the development of a component-based market. ATP has addressed this problem by selecting projects that focus on cross-domain and enterprise-wide applications where the technical challenge of maximizing interoperability is a goal (ATP, 1997).

Network Externalities

Components must interact with other components and infrastructures to create a valuable software product. For example, a new spell-checking component has no value unless basic word-processing components exist that can interact with the new component. As more components are developed, the initial spell-checking component becomes more valuable because a network of components develops. This concept is called a "network externality." Each individual component is more valuable than it is perceived to be because it adds value to all other components.

However, when the individual firm is making market decisions about what research to pursue for specific products, it does not take into account the network benefits that are being created from the initial framework and components. This divergence between the social and private values of a new component has slowed the development of the component-based market. By providing additional funding for high-risk R&D that supports technologies needed for reuse, ATP can help compensate for the discrepancy between the social and private returns from new components and thereby help stimulate development of reusable components and their commerce.

Lock-in

Legacy software and extensively customized large software systems are abundant in large firms. They contain massive amounts of information about how the company operates and identify critical data that must be collected and analyzed. Significant resources are spent on maintaining, updating, and monitoring their performance. Migrating complex, monolithic systems to newer, component-based systems that can easily be updated and improved could lower these costs.

However, firms have been reluctant to change to component-based systems. The main explanation is that the short-run costs of migrating from the existing system to a component-based system are prohibitively high, even though the long-term benefits could be substantial. Firms are locked in to their existing systems and are unwilling to adopt new systems (Brown and Wallnau, 1998b).

There is anecdotal evidence that fixing the Y2K bug may have pushed many firms into replacing monolithic systems because of the excessive cost of fixing the problem within the existing legacy software. However, the studies are not yet in place to verify that this was a widespread phenomenon or that these firms adopted component-based approaches for their newer systems.

2.4.2 Role of ATP

In 1994, ATP held a series of workshops with industry and academia to discuss the potential for a focused program on component-based software. These workshops, along with various white papers from the technical community, resulted in the ATP recommendation for a focused program. Another workshop was held in 1995 to update the program scope.

Over the course of three rounds of funding, ATP committed a total of nearly $70 million and industry committed an additional $55 million of cost-sharing funds to a total of 24 projects. Table 2-3 provides information on ATP-funded projects throughout the program's history. Table 2-4 describes the status of each project at the time this study was initiated. Of the 24 projects funded between 1994 and 1997, 18 were completed by the time of this study, two are still underway, and four failed to complete. Two-thirds have yielded commercial products. To date, three-fourths of the projects have reached the commercialization phase and have begun marketing activities even though some of the resulting products are not yet generating revenues.

Table 2-3. ATP CBSD Funded Projects

Project Number	Company	Project Title	Company Size	Project Type	Project Status December 2000	Project Budget (Thousands)
94-06-0003/5h1017	SciComp Inc.	Automatic Generation of Mathematical Modeling Components	Start Up	Single	Closed, selling product	2,235
94-06-0006/5h1018	Cubicon Corporation	Cubicon's Visual Programming Environment for Reusable Software Components	Start Up	Single	Failed to complete	2,430
94-06-0007/5h1019	Aesthetic Solutions Inc.	A Component Technology for Virtual Reality Based Applications	Start Up	Single	Closed, selling product	2,277
94-06-0011/5h1020	Lucent Technologies	Automation of Dependable Software Generations with Reusable Components	Large	Single	Closed, technology not in use	5,435
94-06-0012/5h1021	Accenture	Component Integration: An Architecture-Driven Approach	Large	Single	Closed, technology not in use	4,012
94-06-0012/5h1022	APT/Torrent Systems Inc.	Component Based Software System for Parallel Processing Systems	Start Up	Single	Closed, selling product	2,308
94-06-0026/5h1023	Reasoning Inc.	Component-Based Re-Engineering Technology	Small	Single	Closed, selling product	3,433
94-06-0032/5h1024	Kestrel Develop-ment Corporation	Scalable Automated Semantic-Based Software Composition	Led by Small	JV	Failed to complete	45,655
94-06-0034/5h1025	Continuum Software Inc.	Scalable Business Application Development Components and Tools	Start Up	Single	Closed, selling product	3,894
94-06-0036/5h1026	Xerox, Palo Alto Research Center	Reusable Performance-Critical Software Components Using Separation of Implementation Issues	Large	Single	Closed, in commercialization	3,141
94-06-0037/5h1027	Unisys	Scalable, Extendable Methods and Tools for Integrating Components (SEMANTIC)	Led by Large	JV	Failed to complete	7,108

(continued)

Table 2-3. ATP CBSD Funded Projects (continued)

Project Number	Company	Project Title	Company Size	Project Type	Project Status December 2000	Project Budget (Thousands)
95-09-0016/5h1159	Analogy, Inc.	A Component-Based Software Approach to Analog and Mixed Signal Model Development	Small	Single	Closed, in commercialization	2,399
95-09-0021/5h1160	Reliable Software Technologies Corp	A Plausible Dependability Model For Component-Based Software	Small	Single	Closed, in commercialization	2,375
95-09-0032/5h1163	Extempo Systems, Inc.	Component-Based Software for Advanced Interactive Systems in Entertainment and Education	Start Up	Single	Closed, selling product	2,537
95-09-0033/5h1162	Tom Sawyer Software	Graph Visualization Technology	Small	Single	Closed, selling product	2,919
95-09-0045/5h1163	Real-Time Innovations, Inc.	Component-Based Software Tools for Real-Time Systems	Small	Single	Closed, selling product	2,618
95-09-0052/5h1164	HyBrithms (Hynomics Corp)	Cost-Based Generation of Scalable, Reliable, Real-Time Systems	Start Up	Single	Closed, in commercialization	2,107
95-09-0059/5h1165	Semantic Designs, Inc.	Design Maintenance System	Start Up	Single	Closed, selling product	2,131
97-06-0005/7h3049	Reliable Software Technologies Corp	Certifying Security in Electronic Commerce Components	Small	Single	Ends 1/1/2001	2,358
97-06-0008/7h3045	Data Access Technologies, Inc.	Business Object Component Specification, Generations and Assembly	Start Up	Single	Ends 6/30/2001	2,649
97-06-0023/7h3046	Synquiry Technologies, Ltd.	A Programmable Framework Based on Semantic Modeling Components	Start Up	Single	Closed, in commercialization	4,062
97-06-0032/7h3048	Commerce One, Inc.	Component-Based Commerce: The Interoperable Future	Led by Start-up	JV	Closed, in commercialization	10,156
97-06-0037/7h3047	Sterling Software, Inc.	MirrorBall: A Component Infrastructure Initiative	Medium	Single	Failed to complete	
97-06-0038/7h3050	Intermetrics Inc., now Averstar Inc.	Debugging Component-Based Software for Enterprise Systems	Medium	Single	Closed, Licensed Technology for Commercialization	2,670

Note: Small companies are defined by ATP as those with fewer than 500 employees. Large firms are members of the Fortune 500. Medium-sized firms are all others.
Source: Data extracted from the ATP Focused Program Status Reports and Closeout Reports housed at NIST; information was current as of December 2000.

Table 2-4. December 2000 Status of CBSD Projects by Funding Round

	Total Projects	In Progress	In Commercialization	Technology Not in Use	Failed to Complete
1994	11		6	2	3
1995	7		7		
1997	6	2	3		1

Source: Data in Table 2-3. "Failed to complete" indicates that the project was abandoned prior to achieving its technical objectives.

Historically, the process of developing and using software components has been extremely complex. A number of barriers created substantial technical risk to innovation in this area, including a lack of the following:

- automated tools for building, locating, and adapting components;
- interfaces for nonprogrammers to enable them to sue the components; and
- specifications of interface semantics for bridging applications.

The ATP's focused program for CBSD sought to support development of such tools, with special emphasis on tools that enable complex, large-scale, commercial applications that can be used by a wide spectrum of industries. Within this area, ATP has focused its funding efforts in two separate directions: building components and building component architecture. Developing the basic building blocks for components and architectures that apply to all component users is likely necessary to the maturation of component-based software. Once component-based software markets have emerged, software developers are expected to specialize and focus on specific applications of components. The component-based software industry would supply automated high-performance software tools and architectural services that could be incorporated into the development of computer systems. This model would allow a programmer to focus on the problem at hand without being concerned about syntax or the programming process (Cuthill, 1997).

ATP investments have created several widely agreed-upon effects, including acceleration of R&D spending on projects with high technical risks, and increasing the probability of a project's success. ATP funding may also broaden the scope of technology development, and thereby enable wider application. ATP's investment in CBSD is likely to accomplish similar objectives. For example, ATP's investment should accelerate the benefits that emerge. ATP funding may increase the scope of CBSD by providing more tools for users and by expanding opportunities for use.

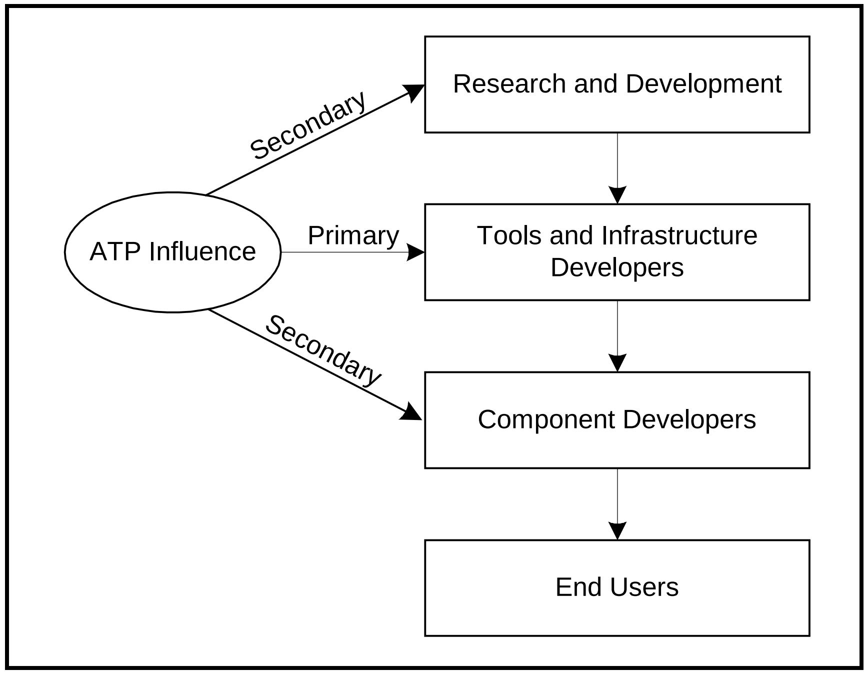
Before ATP involvement, industry software developers, universities, and other researchers were already engaged in the development of a component market. Given that ATP funding is small compared to the total amount of component-based research, it is likely that the market for CBSD would have developed without ATP funding, but perhaps along a delayed trajectory or to a lesser extent.

2.5 CBSD SUPPLY CHAIN

The supply chain within the software development industry consists of approximately 4,500 firms, with an average of fewer than 20 employees per firm (ATP, 1999). Figure 2-2 provides the supply chain for the industry, while Table 2-5 gives examples of firms or organizations that are operating at each level. The supply chain starts with firms, think tanks, and universities that are engaged in research and product development. The next level of the supply chain consists of software developers that are building the infrastructure for CBSD. The third level consists of firms that are designing and producing the components that the end users will use as an input into their production process.

Research centers, such as the COTS-Based Systems initiative at Carnegie Mellon University and the Reusable Software Research Group at Ohio State University, are engaged in the basic research needed to develop a component-based market. These researchers are producing the basic knowledge that other developers would apply to component production. It is appropriate that universities, funded by various governmental agencies, are conducting these activities, because appropriating the knowledge that they create would be a near impossible task.

Figure 2-2. CBSD Supply Chain



Following the basic research stage, software firms engage in R&D for writing, designing, and developing the architecture for supporting a component-based system and the tools that can be used to design components. They are building and designing the production tools that can be used to build components. This area is where ATP has focused its three rounds of funding. ATP funding in this area has two potential effects. First, ATP funding can accelerate the building process. Second, ATP can put more or better tools into the production shop; rather than having one type of hammer, software developers may now have an entire toolbox.

The next step down the supply chain consists of the software developers that use the tools and architecture developed in the production shop to design and build components. Because the tools and infrastructure are completed sooner, software developers are able to design and build components earlier. Currently, over 221 firms are designing and building components (Components Source, 2000).

Table 2-6 provides the number of firms, employment figures, and sales receipts for firms that were engaged in these two phases of the production process in 1997. Table 2-6 captures all firms engaged in producing software, including firms engaged in CBSD and firms

Table 2-5. Selected CBSD Supply Chain Participants

Research and Development	
	Software Systems Generator Research Group, University of Texas
	SEI COTS-Based Systems Initiative, Carnegie Mellon University
	Reusable Software Research Group Ohio State University
	Composable Software Systems Group, Carnegie Mellon University
	Component applications group, Microsoft Research
Tool and Infrastructure Developers	
	Sterling Software
	Microsoft
	Object Management Group (OMG)
	Accenture
	IBM
	Kestrel
	Extempo
	Commerce One
	Averstar
Component Developers	
	Seagate
	Data Dynamics
	Sheridan
	Apex
	Far Point
	Janus
	Video Soft
End Users of Components	
	Bank of America
	U.S. Air Force
	Environmental Protection Agency
	Lockheed Martin
	Merrill Lynch
	AMF Bowling Centers
	NASDAQ Stock Market

Table 2-6. Industry Establishments, Receipts, and Employment

NAICS Code	Industry Classification	Establishments	Receipts	Employment
54151	Computer System Design and Related Services	72,278	108,967,614	764,659
541511	Custom Computer Programming Services	31,624	38,300,515	318,198
541512	Computer Systems Design Services	30,804	51,212,916	337,526
5415121	Computer Systems Integrators	10,571	3,527,055	207,741
5415122	Computer Systems Consultant	20,233	15,942,861	129,785
541513	Computer Facilities Management Services	1,445	15,114,194	71,821
541519	Other Computer Related Services	8,405	4,339,989	37,114

engaged in the traditional approach to software development. In 1997, total receipts by firms engaged in this activity totaled over $100 million. Based on NAICS definitions, firms in this industry are engaged in providing one of four activities:

- ➤ writing, modifying, testing, and supporting software to meet the needs of a particular customer;
- ➤ planning and designing computer systems that integrate computer hardware, software, and communication technologies;
- ➤ providing on-site management and operation of clients' computer systems and/or data processing facilities; or
- ➤ supplying other professional and technical computer-related advice and services.

Firms that fall into this group are Sterling Software, Microsoft, and Accenture; they have all engaged in efforts to generate component-based software and structures to support components. Some of the successful architectures that have been developed and that are currently in use include Active X, Enterprise Java Beans, Component Object Model (COM), and Common Object Request Broker Architecture (CORBA).

CBSD is much more than the languages, tools, and architectures needed for software development that appeared contemporaneously with component-based software. The languages, tools, and architectures provide some of the pieces needed to build a component-based system, but a component-based system is more.

CBSD consists of the enabling technologies described above, the industrial standardizations, the common marketplaces, and the design approaches that are needed to fully develop a market (Brown and Wallnau, 1998a). The existing object technology (OT) simply does not address all of the abstractions needed to have a fully functioning component-based market. ATP has emphasized the search for solutions to these problems in defining the focused program

Once the components are built, end users assemble the components to form software products to use in their production processes. Again, because the components are available to end users sooner, they are able to design and build their software sooner. Table 2-7 lists some of the system architectures, component tools, and components that are currently in use.

Table 2-7. System Architecture, Component Tools, and Components Currently In Use

System Architecture
Active X
Sun's Java Beans and Enterprise Java Beans
Microsoft's Component Object Model (COM) and Distributed COM (DCOM)
Object Management Group's Component Object Request Broker Architecture (CORBA)
Visual Basic
Component Tools
dBarcode-2D Pro
ALSTRA
Total Visual Agent 2000
EcosimPro
KeepTool Hora
ActiveX Manager
System Components
Janus GridEX 2000
Data Dynamics ActiveReports V1.1
Seagate Crystal Reports V8.0
Sheridan ActiveThreed Plus V3.0x
APEX True DBGrid Pro for VB V6.0
VideoSoft VSFlexGrid Pro V7.0
FarPoint Spread V3.0.x

Source: Component Source, 2000.

3 Analysis of Completed Projects

The qualitative evaluation in this section is based on our analysis of information from all 24 funded projects in the ATP focused program. The information and data included here are based on company proposals and business and technical closeout reports provided to us by ATP, as well as a limited amount of independent research. Summaries of the confidential information obtained are included in a separate document delivered to ATP. To inform the quantitative analysis in Section 5, in-depth case studies were conducted for eight of the projects; these results are summarized in Section 5 and presented in more detail in Appendix B.

3.1 INTRODUCTION

ATP began its focused program in component-based software in 1994 with the awarding of funds to 11 companies for software development projects. In two subsequent competitions, the ATP program funded seven new projects in 1995 and six more in 1997. From the program's inception, ATP committed to invest almost $70 million in R&D support. Of the 24 projects awarded funding, 18 were successfully completed, four failed either prior to or during the project itself, and two are just completing the ATP-funded research phase. All but two of the 18 successfully completed projects have produced commercialized technology. Table 3-1 presents the status of the 24 projects as of December 2000.

Table 3-1. Funded and Commercialized ATP Projects by Year and Company Size/Project Type[a]

	Start-Up	Small	Medium	Large	Joint Venture[b]	Total
1994	5/4	1/1	0/0	3/1	2/0	11/6
1995	3/3	4/4	0/0	0/0	0/0	7/7
1997	2/1	1/0	2/1	0/0	1/1	6/3
Total	10/8	6/5	2/1	3/1	3/1	24/16

[a]The first entry in each cell gives the number of companies funded; the second entry gives the number that have successfully commercialized their technology. Source is data in Table 2-4 and company close-out reports.

[b]One of the 1994 JVs was led by a large corporation, the other by a small firm. The 1997 JV was led by a start-up company.

At the beginning of this project, RTI developed 11 hypotheses about potential economic benefits of ATP's involvement in this emerging technological arena. Among these hypotheses were several focused on the firms themselves, including improved quality of outputs, faster development timelines, or increased probability of success. An additional group of potential benefits focused on spillovers, both those that would lower the costs of users of the technology as well as network externalities in the emerging component-software market. Of the hypotheses initially considered, we found the strongest evidence for the following:

- ➤ ATP funding eliminated market failures or missing markets in financing small and start-up firms in the new technology arena.
- ➤ ATP helped accelerate the development of software products, as measured by the estimated time to market.
- ➤ ATP promoted the development of a component-based market, thus serving to increase demand for each of the software products.

In the discussion that follows, we describe the qualitative economic impacts of ATP's involvement based on details obtained from reports submitted by the funded firms. Section 3.2 provides a conceptual model of the new technology development process. Section 3.3 focuses on how ATP has influenced the development process within component-based software. Section 3.4 provides some insights into why some companies that received ATP funding failed to produce a marketable product.

3.2 CONCEPTUAL DEVELOPMENT MODEL

To evaluate the 11 hypotheses about ATP's potential impact on funded firms and the component software market, RTI mapped the CBSD program onto a conceptual framework of the software technology development process. It was somewhat surprising that no widely accepted, comprehensive model for development of software technologies is in use, given all of that has been written about R&D management, the value of new technologies, and the emergence of information technologies. The following treatment comes out of our background understanding of the subject matter and from recent research findings from this study.

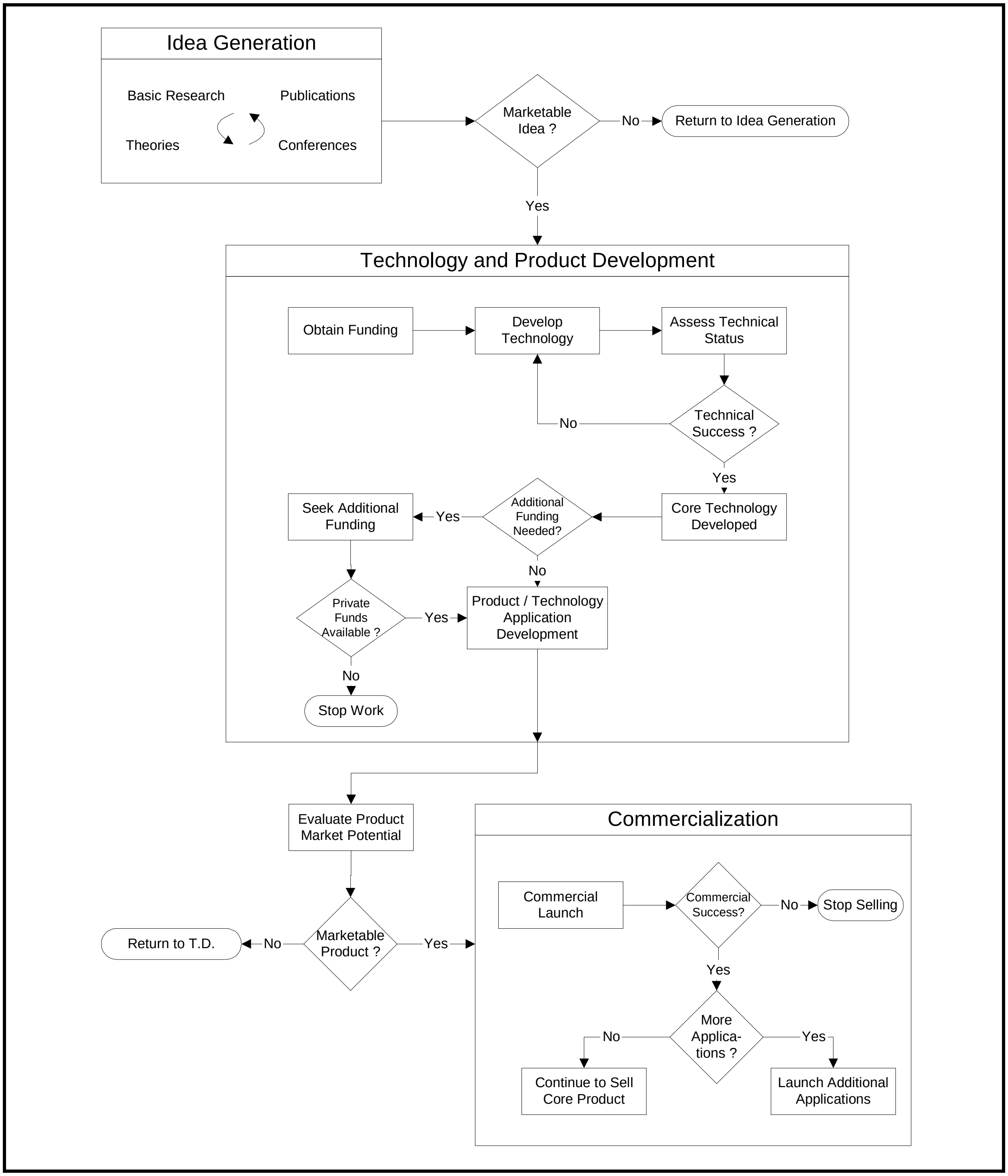
3.2.1 The Development Process

The new technology development process can be separated into three stages: idea generation, technology and product development, and commercialization. Figure 3-1 is a flow diagram depicting the major elements of these steps. A researcher first creates a marketable idea from theoretical breakthroughs, basic research, and a variety of communication activities. This idea is developed into a product or service in the development process, which involves significant expenditure of time and resources and may feature several false starts or dead ends. Once a marketable outcome is created, the product is ready for commercialization in one or several sectors. Although each step in the process is important for developing a new technology, our study focuses on ATP's influence on the last two stages.

Idea Generation

Idea generation is the first stage in the new technology production process. Entrepreneurs take the available body of knowledge in their field and combine it with new innovations and insights to create a marketable idea. Because the pure research undertaken in this phase has the characteristics of a public good, it is widely agreed that public support for this effort is needed. Universities, governmental organizations, and private firms all provide funding for basic research. Many of the entrepreneurs leading the ATP-funded projects were university researchers whose work was supported by institutional and governmental funds.

Figure 3-1. New Technology Development Process



Technology and Product Development

Once an idea has been created, the entrepreneur needs to raise funds to pursue technology and product development. However, different sources of funding have different implications about the amount, type, intensity, and the probability of success of R&D. Different types of funding also have different implications regarding the appropriability and timing of financial returns that emerge from the product that is developed. These different implications can be viewed as constraints placed on the funding that the entrepreneur attempts to raise. The entrepreneur is modeled as trying to raise sufficient funds to maximize the expected return on the development effort, conditional on the constraints imposed by each type of funding source.

Once an entrepreneur has developed a marketable idea, he/she has to seek funding to engage in technology development. ATP can influence this step by providing the first round of funding for projects that meet ATP selection criteria. After acquiring sufficient funding, the entrepreneur engages in research to create new technologies, which can then be embodied in marketable products. Several potential outcomes are possible at this point. If the R&D invested in the marketable idea was successful and the entrepreneur has produced new technologies and products, he/she can move to the commercialization stage. If the R&D has revealed that the idea was not feasible or the technology not marketable, the project ends.

By providing additional resources, ATP can influence this part of the development process in two ways. First, ATP funding can increase the rate at which technology development is pursued. Second, ATP's involvement can increase the probability of achieving successful technical results, thus raising the probability the company will move to the next stage of the process. Although ATP does not fund product development, its involvement in earlier stages can raise the probability of success in this stage as well.

Results from R&D efforts often do not fall into one of those two terminal outcomes. The more common result is that some innovations were made, but questions remain about the potential success of the marketable idea. In this case, the entrepreneur needs to conduct more technology and/or product development to determine the ultimate success or failure of the project. At this stage of the R&D process, the entrepreneur may need to engage in another round of fundraising to conduct the additional work.

Because a significant amount of research has already been completed, the entrepreneur has better and improved information about the marketable idea. By updating his/her understanding of the characteristics of the technology and potential products, the entrepreneur may change his/her decision about where to obtain funding. For example, as the entrepreneur begins working on a marketable product, he/she must switch financing from ATP to internally generated funds or to funding by banks or venture capitalists. ATP can influence this part of the process by increasing the company's credibility when it seeks additional funding.

Commercialization

Once an entrepreneur has developed one or more marketable products, he/she proceeds to commercialization. This process includes deciding where firms should try to sell their products, whether they should expand the scope of the project to fit into multiple industries or uses, and how much of the market they should try to capture.

ATP funding can influence this process in at least two ways. First, ATP technology development funding may allow the entrepreneur to expand the number of potential uses of the project. For example, a component may be designed to be used by multiple architectures. Second, ATP funding can increase the credibility that the company has when it is trying to commercialize the product. A funded firm that successfully completes its project is perceived as having developed the potential to create high-quality products

3.2.2 Sources of Funding

Entrepreneurs have numerous potential funding sources. Although each funding source generally offers the same services and provides a similar product (money for development work), each source is expecting different returns from the investment that they are making with the entrepreneur. This section focuses on the different potential sources of funding that are available to the entrepreneur.

Owner/Angel—The most common way that entrepreneurs fund the first round of financing is by using their own resources or through an "angel." Although this type of funding allows the entrepreneur to maintain ownership of all or a significant amount of the company, it is rarely enough funding to engage in all of the necessary R&D and commercialization efforts to bring a new product to the market.

Stock Issue: Venture Capital—A second funding source that entrepreneurs can pursue is venture capital. Venture capitalists exchange funding for a percentage of the company and generally focus on projects that are close to being commercialized, have appropriable returns, and have little technical risk. Venture capitalists expect the return on investment to occur much sooner than other forms of funding; they will rarely fund basic research (Morgenthaler, 2000).

Stock Issue: Public Offering—A second form of public funding available to entrepreneurs is the issuance of a public stock offering. Similar to funding from a venture capitalist, this approach turns a significant share of the company over to the public. Although issuing a public offering can raise a substantial amount of funding, small and start-up companies are rarely in a position to sell stock. Most larger firms have already gone public, which further limits public offerings as a method to raise R&D dollars.

Long-Term Debt—Rather than selling an equity stake in the company, firms may wish to raise revenue through the issuance of debt. Debt financing allows the firm to raise a significant amount of revenue while retaining ownership. Lenders, however, must face the risk of default in the event of project failure, without the potential for added gains in the case of success. Furthermore, with most of the firm's assets in the form of human capital, collateral for the loans is likely to be inadequate. For this reason, lenders are normally unwilling to supply debt financing for small, start-up firms.

Government Program—Entrepreneurs can also seek governmental assistance in raising revenue for R&D. Most government programs subsidize basic research projects that have numerous potential applications. Governmental funding in basic research is limited in the total amount of resources that it makes available to a company but rarely takes an equity stake in the company's financial returns.

3.2.3 Factors Affecting the Economics of the R&D Decision

Once an entrepreneur has an idea or if he/she is returning to the financing stage, several criteria influence who will be approached for additional financing. This section discusses those criteria.

Retention of Ownership—Entrepreneurs are interested in maintaining control over their project or innovation for two reasons. First, they would like to keep creative control over the production process and direction of the R&D. Second, the greater the retention of ownership, the greater is the percentage of economic returns that entrepreneurs are able to keep. Different sources of funding have varying effects on the entrepreneur's retention of ownership.

Project Acceleration—ATP investment in projects has had a positive effect on moving the R&D investment into the present and has increased the intensity at which R&D occurs (Martin et al., 1998; Silber, 1996). Firms engage in R&D sooner and they spend more money in a shorter period of time. In traditional benefit-cost studies, this acceleration of benefits has always been construed as unambiguously positive for two specific reasons. First, because of discounting, benefits that accrue sooner are valued more than benefits that accrue later. Second, technological innovations that occur sooner have a greater window of market opportunity, allowing them to generate benefits over a longer period of time.

Only recently have researchers started to invest more time and effort into unraveling the impact of accelerating the R&D effort (Dixit and Pindyck, 1994; Vonortas and Hertzfeld, 1998). Recent efforts to understand the impact of R&D investment have compared this investment decision to that of purchasing an option in a stock market (Dixit and Pindyck, 1994). Purchasing the option gives the holder the right, but not the obligation, to purchase a particular asset at a particular price in the future. The funding of R&D is similar in that it gives the provider of capital the right, but not the obligation, to use the product that is developed in the future. In an uncertain world, the option is a valuable asset.

Researchers and scientists have argued that the value of the "technological option" associated with R&D is enough to justify the investment in R&D even if traditional measures of benefit-cost ratios are less than one (Vonortas and Hertzfeld, 1998). The knowledge and information that is generated from the R&D expenditure may open up new and potentially highly profitable areas for products. Traditional benefit-cost analysis that is based on cash flows does not take this potential benefit into account. By funding R&D, ATP is able to create a technological option in these potentially new and large investment areas. The investment in the new field would not have been possible or would have at least been delayed if the original investment by ATP had not been made.

Although it is usually viewed as strictly positive, accelerating R&D investments sometimes can have negative impacts. By postponing the R&D investment decision, new information can eliminate market or technical uncertainties. Firms may learn that the window of opportunity for a particular technical innovation is opening later or closing sooner than previously believed. Alternatively, a new innovation may occur that reduces the cost or changes the direction of the production process. The key benefit that emerges from postponement is the reduction in uncertainty that occurs with the passage of time.

Return Time on Investment—Different projects will reach commercial success at different rates depending on the amount of R&D that needs to be performed. Different funding sources also have different expectations about if and when the project will achieve commercial success and generate a financial return from their investment. This effect should not be confused with the project acceleration effect. Project acceleration focuses on how soon a project reaches the point of commercialization. Return time on investment focuses on how quickly the firm is able to recover its investment costs and achieve financial success. Different expectations about the return time on investment affect the quantity, timing, and intensity of R&D that the entrepreneur can perform.

Credibility—Funding from any source increases the amount of R&D that can be performed, but funding from some sources may increase the credibility of the company that receives the funding. For example, government agencies or investment companies that fund cutting-edge research may provide a stamp of approval for that company and that industry. By receiving approval from an objective third party, the company may have an easier time acquiring additional funding, entering into supply agreements, or marketing its product.

A recent study by Gompers and Lerner (1999) points out that high-tech firms are able to raise equity in two ways to finance their R&D: internally and externally. Internal funds consist of internal equity and retained earnings, and external funding consists of venture capital, bond and loan financing, equity financing, or expenditures by government. Firms need to have capital to invest in the necessary R&D to develop a product. If firms wish to raise external funding, they must convince financial institutions that they will develop a product that has commercial potential.

The problem is that firms know more about their product and the likelihood of commercial and technical success than potential investors or banks do. This situation is referred to as asymmetric information. Firms that are developing a product want to send a signal to the market that the product is marketable. Several benefits emerge if the firm can send a positive signal to the market. Banks and other financial institutions may be willing to lend to the firm at a lower rate or investors may be willing to invest more capital in the firm. However, the market cannot simply trust the firm. Rather, the market needs to receive a signal from a third party that indicates a high likelihood of success for the product.

ATP and its parent organization, NIST, are highly respected agencies that have significant experience in evaluating alternative R&D investments in technology. Financial institutions and investors may see the funding decisions made by ATP as positive signals. In effect, ATP funding creates a "halo" of credibility around the firm that receives the funding. The halo effect is more than a risk-sharing effect. If the funding had come from an unknown source, the next financial institution that lends funds to the firm would not have any additional information about the probability of technical success. The only change would be the decreased probability of failure because financial constraints were relaxed. ATP funding not only reduces the probability of financial failure, but it also sends a positive signal regarding the probability of technical success (Gompers and Lerner, 1999; Silber, 1996).

Probability of Success—Increasing funding will likely increase the probability of success. The entrepreneur has more resources that can now be spent on a project, so he/she engages in more or better R&D. Some funding sources may go beyond an increase in funding and have an additional nonmonetary impact on the probability of project success. For example, some funding sources may help eliminate technical difficulties a firm is facing. Other funding sources may have technical expertise that helps firms solve some project-specific issues, or they may have commercial expertise that helps a firm market its product.

Project Scope—Additional funding increases a project's potential applications. Within component-based software, increases in funding will increase the functionality of the components and architecture that are developed. By increasing the product's ability to work with multiple types of software, beneficial network externalities are created. Network externalities are the benefits that other users and developers receive by increasing the number of products that can interact with the given technology. Some funding sources may push firms to increase the functionality and compatibility of their technological innovations relative to others.

3.3 IMPACT OF ATP FUNDING

The entrepreneurs hoping to develop component-based software need to raise funds sufficient to maximize their expected return on the technology that they are developing, subject to constraints imposed by the funding source. ATP selection criteria have specific constraints that affect the type of projects for which companies may seek funding from ATP; these constraints also affect project and company performance and future financial viability. This section describes effects of ATP's funding on factors of importance, both to the firms that receive funding and the nation's innovation system.

The tables in this section are based on the business closeout reports or the most recent anniversary report of the ATP-funded firms. Because several firms have multiple projects and not all firms responded to all of the questions in the report, the total number of responses varies across questions. To make responses comparable across tables, the percentage of responses in each category is presented.

Additional information on the effects of ATP funding is based on the proposals that the firms submitted to ATP before they were awarded funding and started R&D. When available, the technical closeout reports were also used to increase our understanding of the effects of ATP funding on firms. Fifteen ATP-funded projects had submitted business closeout reports as of July 2000. Two remaining companies, Synquiry and Data Access, were still in the ATP-funded research phase. Multiple applications have emerged from each funded project. For example, Sagent has created five new applications that it intends to commercialize based on the technology it developed with funding it received from ATP.

Over 80 percent of the applications resulting from the research projects in the study are described as new products. Just over 15 percent of the applications are described as a new service, while the remaining applications are new manufacturing processes. Further reflecting ATP's desire to fund cutting-edge research is the fact that a majority (67 percent) of the applications of the ATP-funded projects are attempting to introduce brand-new products or production technologies, rather than improving an existing product or production technology. Of the 24 projects funded, 16 were in the process of commercializing their ATP-funded technologies at the time of the study. Many companies were selling a product or had multiple products.

3.3.1 Retention of Ownership

From the entrepreneur's perspective, one of the main advantages of ATP over other forms of funding is retention of ownership of the firm and the technology. From a social welfare perspective, the retention of ownership is a somewhat different issue. If the innovating firm is bought by an American company, then the benefits from domestic ownership still accrue. However, if the innovating firm or technology is bought by a foreign company, then American social welfare could be affected. Table 3-2 measures the impact of ATP funding on U.S. ownership of the company and the technology. Twenty responses were recorded for this question.

Based on the responses from funded CBSD firms, ATP increased the likelihood of U.S. retention of the company or allowed the company to do the research for roughly half of the firms responding. Most firms that submitted proposals to ATP did not believe that they would have to sell their firms; rather, they said they would have to partner with foreign firms or governments.

Table 3-2. Impact of ATP on U.S. Ownership

How has ATP funding affected these aspects of technology development?	No Project Without ATP	Increased	Decreased	Stayed the Same	Not Sure
Preservation of U.S. ownership of the company?	20%	25%	0%	45%	10%
Preservation of U.S. ownership of the technology?	20%	45%	0%	35%	0%

Note: Based on responses from 20 funded-project closeout reports.

3.3.2 Project Acceleration

For 62 percent of the applications resulting from the ATP-funded technology, respondents to business closeout reports believed that speed to market was critical to achieving project success. For an additional 33 percent of applications, respondents said that speed to market was an important factor in determining project success. Surprisingly, for 5 percent, speed to market was not important.

Further examination of the 5 percent reveals some possible explanations. A firm responded that speed was not an issue for one of their products but added that this product would not reach commercialization for a minimum of 5 years. If the technologies used in this product are so much further advanced than the current state-of-the-art technology, then having them rushed to market would not result in profitable sales because they would be incompatible with existing technologies. It is more difficult to explain the response of a second firm that speed-to-market was not important for two of their applications, despite a window of market opportunity of less than 1 year.

The project acceleration effect can be very significant in some cases. In its proposal, one small firm stated that it could conduct the same total amount of R&D with or without ATP funding. However, without ATP funding, the firm would expect to conduct the research over a 5- to 10-year period rather than a 1- to 3-year period.

Given that the respondents reported speed to market is an important criterion in determining the project's success for 95 percent of applications, understanding the impact of ATP on speed to market is important in estimating ATP's net benefit. Respondents were asked to estimate how much ATP funding will shorten time to market for each application. Table 3-3 presents these results.

Table 3-3. ATP Acceleration of Time to Market

Impact	Number of Applications	Percent
Not at all	7	12%
<1 Year	1	2%
1-2 Years	16	28%
2-3 Years	22	38%
3-4 Years	4	7%
5+ Years	1	2%
No project without ATP	7	12%

Note: Based on 58 applications included in the responses from 20 funded-project closeout reports.

Consistent with previous analysis of ATP-funded projects (Martin et al., 1998; Silber, 1996), for most applications, respondents thought that ATP-funded projects had a 1- to 3-year acceleration effect. Although this acceleration effect may seem minimal, windows of market opportunity are often very small in a new technology setting. For over 93 percent of the applications, respondents said that their market opportunity window would remain open only 2 years after completing the ATP project. The relatively brief market opportunity following project completion, combined with the acceleration effect, suggests that, in essence, ATP funding expands the project's window of market opportunity.

3.3.3 Return Time on Investment

The conventional wisdom, documented by one of the ATP project participants, is that venture capitalists would only fund projects that would generate a return in 18 to 24 months. The ATP participant's research project, like most ATP projects, was projected to take several years to complete and an additional 12 to 24 months to commercialize. The difference in the desired financial returns of venture capitalists and the realities of cutting-edge projects that ATP funds all but eliminates venture capital funding.

Aggregating data across all respondents, we found that for almost 75 percent of the applications, respondents anticipated revenue from their projects within 2 years following completion. This puts the revenue window at 36 to 60 months (project R&D time plus commercialization time). This timeframe is not when profitability occurs (which is what venture capitalists desire), but when the first income stream emerges. For several of these projects, profitability may not occur until well after 60 months. When the profitability delay is combined with a significant potential for technical failure, it is not surprising that most venture capitalists do not fund these projects.

ATP funding may not only help accelerate the current project, but also may influence future projects. Half of the respondents said that ATP funding increased their ability to conduct long-term research, and 40 percent said that they would never have been able to conduct research currently underway without ATP funding. Sixty-five percent of the respondents also stated that the ATP funding increased the speed at which they could conduct R&D projects. Because ATP funding not only affects the current project, but also other R&D projects that the company conducts in the future, it may be appropriate to evaluate ATP funding on a company-level basis, rather than at the project level.

3.3.4 Credibility

A major theoretical effect of ATP is that it increases the credibility of the company that receives the funding. ATP funding gives the company additional resources to use in R&D and indicates the project has gone through a rigorous peer-review process and successfully competed against other cutting-edge research projects. The ATP award is perceived to be a quality certification.

Some economists view ATP funding as a solution to an information problem (Lerner, 2000). Venture capitalists and other potential funders are unable to learn enough about the quality of a firm to agree to make an investment. Because of the uncertainty associated with the firm's quality, potential funders are forced to assume that the firm is of low quality—and do not invest. This problem is referred to as the "Lemons Problem" (Akerlof, 1970). By funding a firm, ATP solves this market failure by informing potential funders that the firm's R&D is of high quality.

To examine this impact, respondents were asked how the ATP award affected their credibility with various business entities. Table 3-4 reports these results. Most respondents stated that ATP increased their credibility in all business activities. Interestingly, although most companies said that the ATP award increased the credibility of their business activities, the extent of the use of the award as a marketing tool varied across companies. For example, one firm released several press announcements and posted an announcement on its Web site about the ATP project. Several other companies have not promoted the award on their Web sites.

Table 3-4. Impact of ATP Award on Firm Credibility

How has the ATP award affected your credibility with:	Increased	Decreased	Unchanged	Not Sure
Suppliers?	65%	0%	25%	10%
Customers?	90%	0%	0%	10%
Investors?	75%	5%	5%	15%
Internal management?	90%	0%	5%	5%

Note: Based on responses from 20 funded project closeout reports.

3.3.5 Probability of Success

Additional funding allows researchers to engage in additional amounts of R&D. Two theories exist about the impact of additional funding on the probability of R&D success. The first theory views R&D as a directed search process. Researchers try different promising approaches until they find success. Additional funding allows them to expand their search. The second theory views R&D as a cumulative process. Researchers apply their past experiences to move along a learning curve about the product or process. They must then reach a threshold level of knowledge along that learning curve before a breakthrough occurs. Within either theory, additional funding increases the probability that success is achieved.

There is no direct way to measure this impact of ATP on project success because no true counterfactual scenario exists. However, as described in Section 2, additional funding increases the speed of the project and the amount of R&D that can be performed. These results are consistent with the first theory regarding ATP. To test the second theory, we can see if firms said that ATP funding was effective at creating new knowledge. Sixty percent of the respondents stated that ATP funding increased their amount of useful knowledge to a great extent, and 40 percent stated that useful knowledge was increased to a moderate extent. No respondents indicated that there was just a slight increase in knowledge or none at all.

3.3.6 Project Scope

ATP funding can increase the amount and rate of R&D on a particular project and it may expand the scope of the R&D project. Expanding the scope of the project could include increasing the functionality of the tools and range of useful application domains for the tools and infrastructure.

When respondents were asked if ATP funding allowed them to expand the scope of their project, half responded affirmatively, and the other half stated that they would not have even conducted the R&D without ATP funding. One of the start-up firms provides an example of how ATP funding increased the scope of a project.

This firm was trying to apply object-oriented technology to the development of efficient, scalable, parallel-computing software and algorithms that could be incorporated easily into hardware systems-independent business applications. They envisioned three major economic benefits from their new technology: lower cost, increased capacity (more users and more applications), and more useful computing capacity (users could use more intensive applications). The company stated that with ATP funding they could achieve all three goals, but could only focus on achieving one or two of the goals without ATP funding.

3.4 CAUSES OF FAILURE

ATP has funded most projects in the CBSD program at just under $2 million. In addition to the funding cost, ATP expends resources on selecting each project, convening meetings, managing projects, and other necessary operations. Even with diligent efforts in project selection and monitoring, some projects may not deliver on their promise. Understanding the causes of these failures may help ATP in future project selection and monitoring.

Failures can occur at three separate points within the project: financing, development, or commercialization. All of the firms in this study were able to raise money to conduct R&D through ATP and other sources; this study can thus only examine causes and impacts at the R&D and commercialization stages. Failure at the R&D stage occurs when the project acquires funds to conduct research but cannot achieve its technical and scientific goals. Four of the funded firms experienced failure at this stage.

Failure at the commercialization stage occurs when the project reaches its technical and scientific goals but does not produce a product that can either be commercialized or used in-house to improve the firm's production process. Two additional firms in the ATP focused program failed to commercialize their technology, and in two of the projects we identified as successes the funded firms were selling a product or service substantially different than that initially intended.

Two of the failures at the R&D stage were due to an inability to form the planned joint venture (JV). One of these projects received a planning grant to organize a JV, but the firms involved never came together and eventually abandoned the effort. A second group was awarded funding to conduct a JV but could not reach final agreement with ATP about the details of the project. Although the specific causes of failure in these two JVs were apparently related to changes in business priorities of the participating firms, it is interesting to note that both halted work during the development of the JV rather than during the research or commercialization stage. In contrast, the ATP-funded Commerce One JV found no insurmountable barriers on the path to technical and commercial success.

The other two firms that did not complete the technology development phase experienced a more typical type of failure. They began working on developing the technology but were not able to meet the goals established at the project's inception or to modify the focus of the project toward an achievable technical goal while remaining consistent with the original proposal. One firm had difficulty hiring the technical experts originally envisioned in the proposal while the other firm underwent changing business priorities due to a merger.

Of the projects that achieved technical but not commercial success, it appears that mistakes in estimating future market conditions were the main explanation. One of these firms incorrectly estimated market demand, while a second experienced changes in a business unit that made pursuit of the ATP-funded technology unprofitable.

The technology pursued by one of the large firms was supposed to be implemented into another of the firm's new products. During the 3-year project, the development team was able to achieve technical success. However, continued development and commercialization were cancelled after the project's completion when the firm purchased a company that was already marketing its own portfolio of similar products. Although the large firm is still pursuing other internal product applications, it is unclear whether any of the firm's products will apply the ATP-funded technology.

One of the companies that achieved only partial commercial success created a solution to parallel computing problems, enabling rapid, error-free implementation of business applications on a wide variety of scalable, parallel computer servers. This company had hoped to work with the applications providers and reported that several firms were willing to adopt their technology to reduce costs. However, once prototypes were developed, the applications providers were unwilling to take the risk of making significant changes to their production processes, and the funded firm was unable to widely sell or license its technology. The firm changed its business model to a solutions approach, abandoning its initial product orientation, but a recent merger and refocusing on different business priorities has made the fate of the ATP-funded technology uncertain.

A second, partly successful project made significant initial progress in developing its component-based software technology during the 3-year project period. Unfortunately, one of the principal investigators left the firm at this point to pursue other opportunities, and the project languished. Business challenges resulting in cost-cutting and a series of layoffs effectively ended the second phase of the R&D process, leaving the firm with a product that was significantly more limited than originally envisioned. At this point, it is doubtful that the firm will be able to develop, market, and install the complementary technologies needed to make its core product effective, though the developed technologies may be used in other products.

4 Impact Measurement Methodology

In many respects, the firms engaged in software development are unlike companies typically envisioned in basic economics theory. Although they sell their products in competitive markets, their ability to differentiate their products gives them a good deal of market power in new, highly differentiated product areas and thus the ability to raise prices above marginal and average cost. Because of the nature of the R&D process, however, software development companies must commit significant funds far in advance of achieving either technical or commercial feasibility. These pre-commercial fixed costs, along with low marginal production costs and highly differentiated products, produce many of the features of natural monopoly markets. The rapid pace of technological development in software limits the scope and duration of the market power the innovating firms exert, however, as expanded product markets and competitor companies inevitably emerge around the new idea and product.

Many of the companies involved in component-based software are small start-ups that face the types of financing constraints discussed in Section 3.2.2. Because the bulk of their expenditures occur prior to their earning any revenues, indeed before technical feasibility has even been established, these firms have difficulty obtaining capital from loans or equity participation. Funding by ATP or another source of patient capital is thus the only way that these technology development projects can be undertaken.

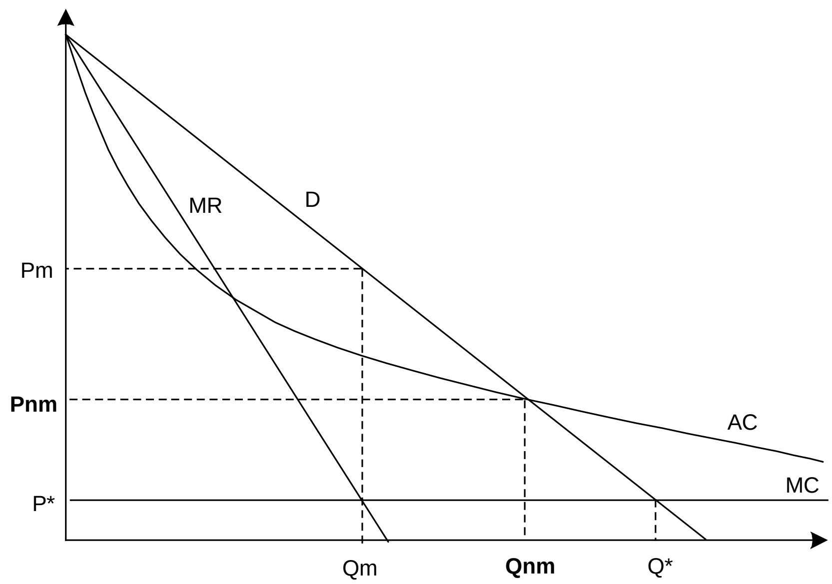
In this section, we present an economic model based on natural monopoly markets, building on the qualitative results from the first phase of this study. We then discuss how this general model was operationalized as a means to estimate the economic benefits from the ATP focused program. Benefits to customer/users of the products sold by the ATP-funded firms are estimated in addition to benefits to the ATP-funded firms. This methodology was used to analyze the economic contribution of the eight case studies of successful component-based software firms. Details of each of the case studies, including economic performance calculations, appear in Appendix B; an analysis of the entire ATP "portfolio" and with cross-cutting qualitative results comprises Section 5 of this report.

4.1 AN ECONOMIC MODEL OF SOFTWARE MARKETS

When companies develop technologies to be embodied in new software products, R&D costs are incurred well before any commercialization activities commence. Once products are released, firms hope to recoup these up-front R&D costs by charging a premium above the marginal cost of production. When the firms make the initial decision to invest, the expected value of their future profits must be sufficient to amortize the development costs. However, firms may also be concerned about the possibility of competitors entering the market. As a result, we expect that originating firms will price their products at a point that is high enough to cover the fixed and variable costs of production, but low enough to slow the entry of other firms into the market.

This type of market can be conveniently analyzed as a natural monopoly (i.e., a market characterized by high fixed costs of entry and low marginal costs of production). Figure 4-1 presents a diagram that illustrates this analysis. It shows the demand curve for component-based software products (D) and the associated marginal revenue curve (MR). A small, constant marginal cost of producing each unit of software is shown (MC), as is the average total cost of production (AC). Fixed costs per unit, which are downward sloping and approach zero at high levels of output, are omitted here for clarity. The AC curve shows the key characteristic of natural monopoly markets, that average costs fall with increasing production. This effect gives a large advantage to high-volume producers and favors those firms that are first into the market.

Figure 4-1. Market Equilibrium for Natural Monopoly Software Product



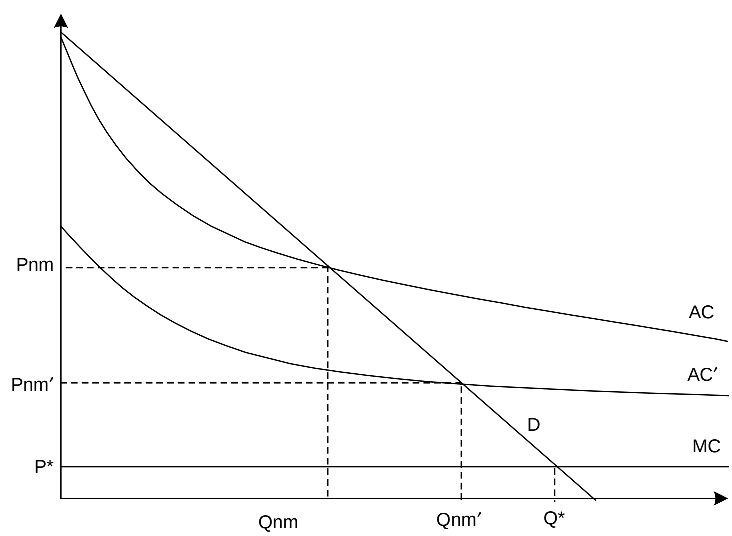
If a single firm could be assured of maintaining a monopoly, either through control of proprietary technology or by legal protection, it would set price at Pm and expect to sell Qm units of its product. In most software markets, however, this high price and profit level would attract profitable entry from other firms. The would-be monopolist will actually be worse off than if it had set a slightly lower price. Firms that had not incurred the R&D costs would have substantial incentive to enter and capture the market with a similar product at a lower price. To deter entry, the innovator firm will charge a price of Pnm and produce Qnm units of output. This price allows the innovator no profit, because price equals average cost but also permits no profitable entry.

Neither of these equilibrium points is socially optimal, however. Society would be better off if production were pushed out to Q* and the price charged were equal to marginal cost, denoted here as P*. Producers will never willingly operate at this level, however, because price would be well below average cost—the firms would not be able to recoup their fixed R&D costs. In the absence of total fixed cost subsidization by the government, this first-best outcome cannot be obtained in a market setting.

Several potential effects of ATP funding can be illustrated in this model. One effect of ATP's paying for a portion of technology development in the component-based software industry is a reduction in the fixed costs borne by the firms. As a result of ATP's support, average variable cost is reduced, while the marginal costs are unchanged.

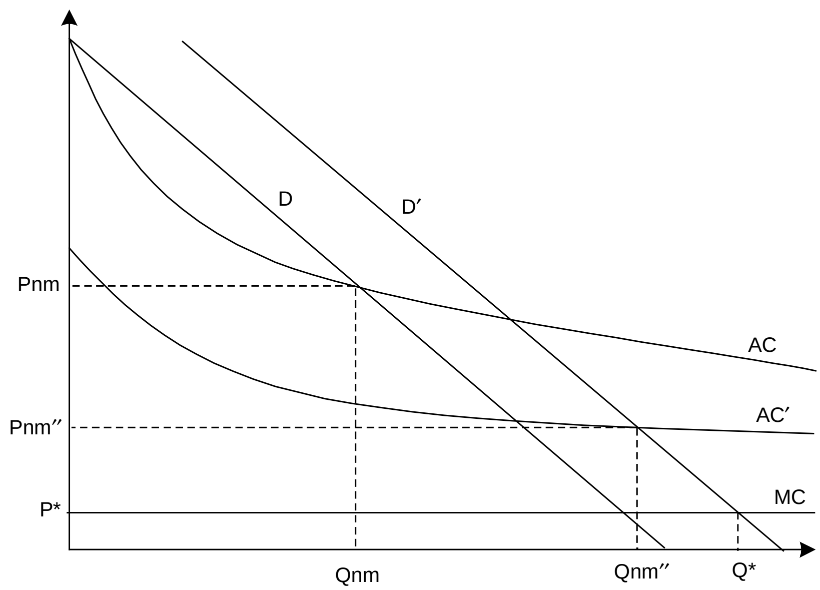
This concept is presented in Figure 4-2, with the reduction in fixed costs driving average costs down from AC to AC′. When this occurs, an entry-deterring firm will increase production to Qnm′ and allow the price to fall to Pnm′. Social surplus increase from the initial state, although quantity is lower and price higher than the social optimum because firms still need to cover their (lower) fixed costs of production.

Figure 4-2. Effect of R&D Subsidy on Software Product Equilibrium



ATP funding may produce an additional effect that augments the increase in social surplus. By promoting the development of a component-based market, demand for components may increase. This effect is presented in Figure 4-3, where demand shifts from D to D′ and average costs fall from AC to AC′. In this diagram, the shifting out of the demand curve is accompanied by an increase in the quantity produced, and price and output move even closer to the efficient level.

Figure 4-3. Effect of Demand Increase on Software Product Equilibrium

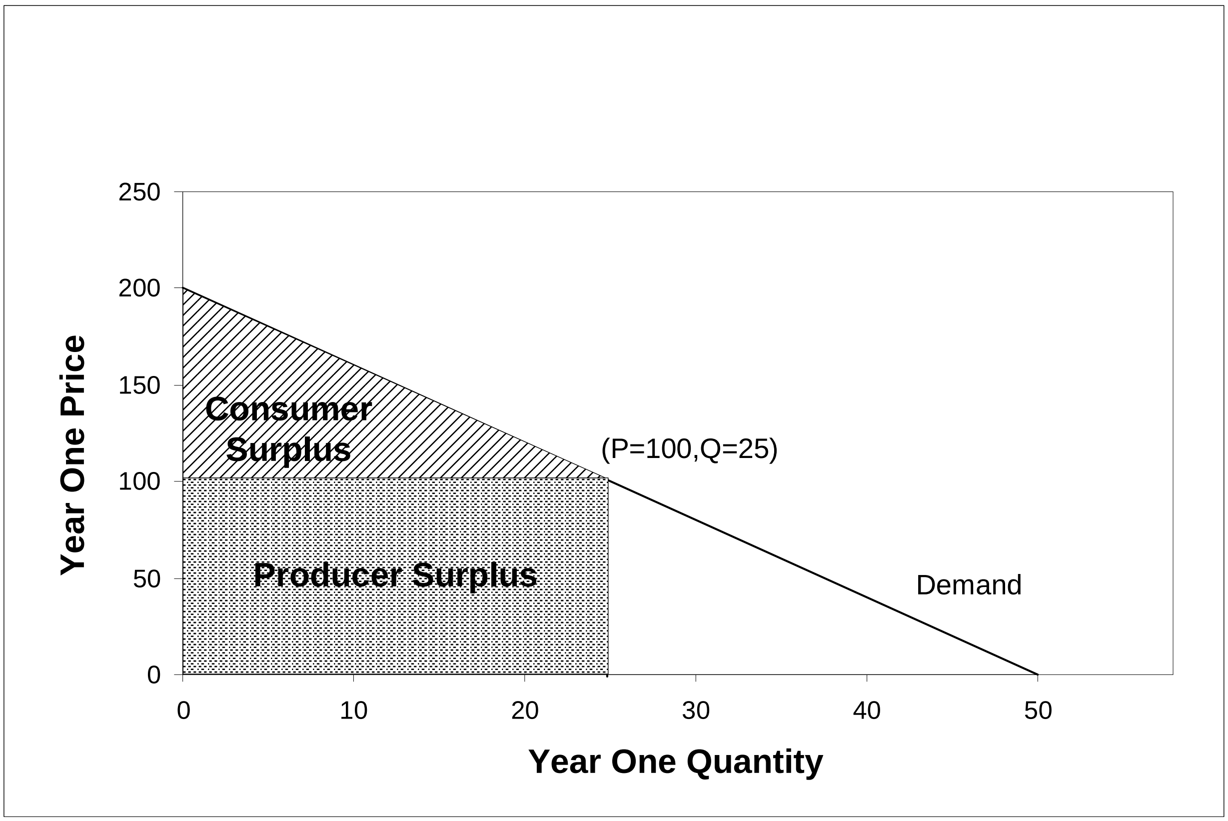


Previous work has suggested that ATP involvement may increase demand for component-based products and tools through network externality effects, by conferring scientific credibility on the new technology, or by demonstrating a commitment to continued financial support. For example, as the number of available components grows, each newly developed component makes existing components more valuable, a positive externality which is magnified by ATP's efforts. Information disseminated by ATP and project participants may convince applications developers of the inherent advantages of a component-based software design approach; the continued funding of new projects may, in turn, convince these same developers that the array of components and tools will continue to increase in the future. In the detailed case studies, RTI assessed the extent to which the funded firms have seen demand shift.

4.2 OPERATIONALIZING THE MODEL TO QUANTIFY ECONOMIC IMPACTS

When attempting to measure the economic benefits created by the ATP projects, we see that the natural monopoly in the product market ensures that the social benefits will be shared between the innovating firm and its customers. In each period that a product is sold, the component-based software firm earns a producer surplus, equal to the difference between price and marginal cost, multiplied by the number of units sold. Figure 4-4 shows producer surplus in a single year for a hypothetical component-software firm selling products at zero marginal cost.

Figure 4-4. Consumer and Producer Surplus Calculation for Component Software Product



By discounting the producer surplus for every period back to the project's start, along with the firm's R&D investments, measures of private return (profit) can also be calculated. To calculate benefits to users of the products sold by the ATP-funded firms, we need to capture two additional sets of financial flows: benefits that spill forward to the component firms' customers and public expenditures by ATP. In this section, we discuss how we measured each of these flows. For each project, the combined producer and consumer surplus comprise the total social benefit.

4.2.1 Measuring Benefits Captured by Customers

To estimate directly the benefits captured by the downstream firms, we needed to measure the impact of the software components on reducing customers' production costs. This is an empirically difficult proposition and could be approached in at least two ways. One method would be to interview customers of each of the products, as Mansfield did in his classic industry studies (Mansfield et al., 1977). A second approach, the one taken in this project, is to question the innovating firms directly about how much their customers would be willing to pay for each product using the ATP-funded technology.

The consumer surplus generated by sales in a market can be thought of as the benefit received by buyers of a product over and above the amount they must pay to purchase it. On a demand graph such as Figure 4-4, it can be measured as the area below the demand curve and above the price line. In standard public economics analysis, consumer surplus is not an exact measure of the social benefit from the product in question, because the buyers would enjoy some surplus from an alternate use of their money. The net social surplus would more properly be measured as the difference between the benefit received from consuming the product and that of the next best alternative.

In the present case, however, the "consumers" in question are actually firms that use software components in their own products. When we measure consumer surplus in such a market, it represents the cost savings incurred by purchasers of the components. Those firms that can reduce their software production costs the most will have the greatest willingness to pay (WTP) for the component product. Some will be willing to pay less. In combination, the different customer preferences define the product demand curve and enable us to compute consumer surplus.

Because all of the surplus is actually cost reduction, an estimate of consumer surplus is an accurate measure of the public benefit generated by the component. As a result, it is possible to extract the customer surplus information we require directly from the component producers, under the assumption that they are well informed about the value of their products to their potential and actual customers. A properly estimated demand curve for the component products contains all of the information about the social benefits that are passed along from innovator to customer firm to final consumer. This equivalence has been much discussed in the microeconomics literature, especially in the field of welfare economics. An accessible and cogent demonstration of the duality of surpluses appears in Just et al. (1982).

A second, even more compelling reason for obtaining the needed information directly from the component software firms is the wide range of customers for each of the component-based products and services. The logistics of contacting tens or hundreds of customers for each product would make this a prohibitively time-consuming and expensive proposition, beyond the scope of this project. The very nature of software components ensures that purchasers are likely to be diverse, enjoying quite different potential benefits from using the components. As a result, the most cost-effective source of data on customers' willingness to pay is the software producers themselves, who need to be aware of the valuation of potential customers to succeed financially.

4.2.2 Empirical Measurement of Derived Demand

Having decided to extract data on derived demand from the innovating firms, we faced the issue of the best way to obtain the valuations. This issue can be viewed as attempting to empirically construct the demand curve like that described in Section 4.1. Product prices and quantities sold in a given year provided the equilibrium point on the curve. By assuming a linear form for the demand curve, one additional piece of information allowed calculation of the producer and consumer surpluses. One method is to ask firms to estimate their customers' maximum WTP, thus establishing a "choke point," or the price at which the first unit will be demanded. An alternative approach is to estimate the quantity that would be sold if price were lowered by a specified percentage. We included questions in the case study interviews aimed at informing these two methods. The case study analyses in Appendix

B include estimates derived from one or the other approach, depending on which appeared more reliable for that case.

4.2.3 Measuring Producer Surplus and Profit

As we note above, producer surplus consists of the difference between marginal costs and price for each unit sold during a specified time period. With fixed costs of development incurred years before revenues are accrued, relatively large producer surpluses are to be expected in software product markets. By questioning firms about their prices, marginal costs, and production quantities, we expected to obtain the information needed to estimate surpluses in every year of the product's life. Marginal costs, consisting mostly of duplication and distribution expenditures, were assumed to be constant with increasing volume.

With each product being offered over a multiperiod time horizon, the producer surplus in each year, along with fixed cost R&D investments, can be discounted back to the project's inception to yield an expected profit for the producing firm. Each funded component software company was asked to report its R&D and related spending for product launch. Over the entire product life cycle, this profit must be expected to be positive or else the firm will not engage in costly R&D. In a natural monopoly market with entry-deterring pricing, the expected value of lifetime profits is zero. When measured empirically, however, we expect that estimated profits may be positive or negative, for a variety of reasons:

- Positive profits will result if the firm sets its price above the entry-deterring level. For instance, if the firm does not anticipate competition during the life of the product, it may choose a price that will yield positive profits. If the innovator is incorrect about potential competition, we would expect entry to occur over time. In several of the case studies discussed in the next section, we calculated positive profits for the component software firms.
- Negative profits might be found if the firm does not correctly account for its fixed development costs or if it is forced by competition to set a price below its average total cost. One of the firms in the case studies was found to earn a negative profit, although the total surplus generated by the component-based product was significant.
- In addition to these two non-optimal situations is the possibility that RTI might have employed a different discount rate than the component software company did, and its "true" expected profits are positive or zero. In the small

case study sample, we found firms had small positive profits, indicating close to optimal entry-deterring behavior.

4.2.4 Capturing ATP Expenditures

We used monthly payments data from ATP, which we then aggregated to annual flows in calculating gross and net social returns. For the smallest firms, ATP funds made up half or more of the capital spent on technology development, while for the larger firms and joint ventures, corporate resources accounted for a majority share.

4.2.5 Summary of Calculation Methodologies

The evaluation of the economic impact of the component-based software focused program involved calculating several performance measures, both at the firm level and for the entire portfolio of funded projects. Three of these measures, benefit-to-cost ratio (B/C), net present value (NPV), and internal rate of return (IRR), provide estimates of the net social surplus created by the combined public and private investment. A more in-depth description of each of the measures follows.

Benefit to Cost Ratio (B/C)

The annual time series of benefits and costs was assembled for each of the eight case study projects. Letting B_t be the net benefits accrued in year t and C_t the total funding for the project in year t by ATP and industry, then the benefit-cost ratio for the program is given by

$$(B/C) = \frac{\sum_{i=0}^{n} \frac{B_{(t+i)}}{(1+r)^i}}{\sum_{i=0}^{n} \frac{C_{(t+i)}}{(1+r)^i}}, \qquad (4.1)$$

where t is the first year in which benefits or costs occur, n is the number of years the benefits and/or costs occur, and r is the social rate of discount. In this study, r was set at 7 percent, the Office of Management and Budget (OMB) specified level. Because benefits and program costs may occur at different time periods, both are expressed in present-value terms before the ratio is calculated.

A similar method was used to evaluate the B/C ratio for the entire CBSD portfolio of 24 funded projects. For the overall program calculation, the results of which appear in Section 5, the sum of the benefits produced each year for the eight projects was included in the summation term in the numerator. The denominator included outlays of ATP and industry funding, both during and after ATP funding, provided to all 24 projects, a conservative assumption that is discussed in greater detail in Section 5.

Net Present Value (NPV)

The NPV of ATP's contributions to the CBSD projects was calculated as

$$NPV = \sum_{i=0}^{n} \left[\frac{B_{(t+i)}}{(1+r)^i} - \frac{C_{t+i}}{(1+r)^i} \right], \qquad (4.2)$$

where the terms have the same meanings as identified for the B/C determination. As before, the overall portfolio NPV was found by summing the discounted net benefits from the eight in-depth studies, less the total focused program expenditures. Any project that yields a positive NPV is considered to have been economically successful. It should be noted that the 7-percent real discount rate required by OMB is a rather high hurdle for project analysis, ensuring that the software projects that showed a positive NPV were quite socially advantageous.

Internal Rate of Return (IRR)

The IRR is the value of r that sets NPV equal to 0 in Eq. (4.2). Its value can be compared to conventional real rates of return for comparable or alternate investments. Risk-free capital investments such as government bonds can be expected to yield rates of return under 5 percent in real terms, while equities seldom return more than 10 percent over an extended period of time. In academic studies of the diffusion of new technologies, however, real rates of return of 100 percent or over have been found for significant advances. It should be noted that for projects in which costs exceed benefits, an IRR cannot be calculated.

5 Quantitative Analysis of Eight Case Studies

5.1 PROGRAM DESCRIPTION

ATP has pursued a focused program in component-based software. The focused program is unique in that ATP is investing in a broad range of projects with a common objective. Each project is pursuing different innovations in component-based software, but all have the same intent of bringing new technology to the software production process. Like all risky, high-tech research, the reality is that some of these projects will produce significant new innovations and others will fail.

The ATP program in component-based software was a 5-year effort to establish the technical foundations for fundamental change in the software industry. Drawing on current research in automated software design and production, the program supported projects to enable a market in broadly useful software components. The goal of the program was to make the use of software components as ubiquitous as the use of integrated circuits that make up today's component-based computer hardware. In the new component-based world, components will be automatically combined and configured by software composition tools to mesh into large applications without requiring the user to understand how the individual components operate. As development becomes easier, software engineers will be able to move from the mechanics of the software development process to the development of applications using automated tools to assemble and integrate independently produced components. ATP made a variety of investments to accelerate the development of the technology.

Because of differences in success rates, we evaluated the entire program as a portfolio of investments. By choosing this approach we are getting an accurate picture of the total program. If we only chose one of the companies to examine, we might drastically under- or overestimate the benefit from ATP. For example, if we were to examine a single project that was highly successful with an internal rate of return of over 100 percent, we might conclude that all of the funded companies were highly successful. Alternatively, if we were to examine a project which produced a negative internal rate of return, we could assume that the entire ATP program was a failure. A portfolio approach provides the best picture of the entire program. Table 5-1 provides a profile of the projects' ATP funding.

Table 5-1. ATP Funding of CBSD Program, by Year

Start date	1994
1994 outlays	$72,727
1995 outlays	$6,165,483
1996 outlays	$8,828,986
1997 outlays	$9,983,280
1998 outlays	$8,958,650
1999 outlays	$6,223,988
2000 outlays	$1,827,848
Total ATP outlays	$42,060,962

Source: ATP monthly project payments data.

Section 5.2 describes the actual returns and presents quantitative measures of the economic performance of the CBSD focused program. Perhaps more interestingly, Section 5.3 presents the lessons learned from the ATP program and describes the non-market benefits of ATP funding not directly visible in the measures of economic performance.

5.2 PORTFOLIO PERFORMANCE

As described in Section 1.3, eight of the 24 funded projects were selected for quantitative economic analysis. At first it might appear that RTI used a "cherry-picking" approach by selecting only the best projects to include in the portfolio analysis. However, this is not the case. We estimated the benefits from the eight projects that we believed were a priori the most successful and included costs experienced for the entire set of 24 projects. We assumed that the remaining 16 projects generated zero economic benefit. This is an extremely conservative approach. By assuming that these projects had no benefits and only costs, we are intentionally biasing our results downward. If the benefits from the few selected projects were able to overcome this severe, intentional, built-in bias, we could have strong confidence that the entire focused program was successful.

5.2.1 Overall Focused Program Performance

From each of the eight in-depth case studies, we estimated the total social benefits generated by actual and projected sales of products based on the ATP technology. This stream of benefits was then compared to the total costs of the project to generate a portfolio net benefit estimate and a portfolio internal rate of return. Table 5-2 shows ATP and private expenditures in the component-based program and the stream of benefits realized by the program. Based on the data from Table 5-2, we calculated an overall net present value, benefit-cost ratio, and an internal rate of return. These are presented in Table 5-3.

The ATP component-based program was successful; based on a discounted investment of $34 million in ATP funds and $55 million by the private firms, the program was able to generate estimated discounted benefits in excess of $850 million. Even this return is an underestimate of the total benefits of these investments. It only includes benefits from eight of the projects, limits the life span of the projects, and is based on conservative estimates of the benefits from each of the individual technologies that ATP funded.

Table 5-2. Total Expenditures and Benefits (thousands of 2000 dollars)

Year	ATP Expenditures	Private Expenditures	Total Benefits	Net Benefits
1994	$100	$700	$100	–$600
1995	$6,800	$5,100	$100	–$11,800
1996	$9,500	$19,700	$100	–$29,100
1997	$10,500	$8,500	$100	–$18,900
1998	$9,300	$9,400	$5,600	–$13,100
1999	$6,300	$23,100	$41,700	$12,300
2000	$1,800	$3,800	$204,300	$198,600
2001		$2,900	$459,700	$456,800
2002		$500	$469,700	$469,200
2003		$500	$464,500	$464,000
2004		$300	$21,200	$20,900
Total	$44,300	$74,500	$1,667,100	$1,548,300
In NPV Terms	$33,800	$54,500	$929,000	$840,000

Note: Total expenditure and benefit amounts have been converted into real 2000 dollars. Private Expenditures include both industry cost-share on ATP projects and subsequent industry funding for product development and marketing. The NPV calculations discount these amounts expressed in 2000 dollars back to the start of the CSBD program in 1994.

Table 5-3. Measures of Performance of ATP-Funded CBSD Program

Net present value (in 2000 dollars)	$840 million
Benefit-cost ratio	10.5
Internal rate of return	80%
Total producer surplus (in 2000 dollars)	$538 million
Total consumer surplus (in 2000 dollars)	$1,129 million

5.2.2 Individual Project Performance

The performance of the portfolio of projects funded in the focused program follows expectations for risky investment projects. Some of the projects failed and barely generated any social returns, while others were quite successful. Three of the projects—the Commerce

One JV and the Tom Sawyer and Intermetrics single-company projects—generated enough returns independently to cover the entire cost of the focused program. The benefits generated from these three projects compared to the entire investment costs generates a benefit-cost ratio of over 10. This result is very consistent with high-risk, high-return investment. Given the high-risk nature of new software and Internet companies during the period that ATP invested in these companies and the time when they started to release new products, the returns generated by the ATP focused program are significant. Table 5-4 lists the estimated project returns for the eight in-depth case studies.

Table 5-4. Individual Project Performance

Project	Net Present Value	Internal Rate of Return	Benefit-Cost Ratio
Aesthetic Solutions	–$1.2	N/M	0.4
Commerce One JV	$789	363%	39.0
Extempo Systems	–$1.2	N/M	0.6
Intermetrics	$29.6	103%	9.6
Real-Time Innovations	$2.0	31%	1.8
SciComp	$21	51%	7.6
Tom Sawyer Software	$51	137%	18.0
Xerox PARC	$1.2	13%	1.2
Overall Portfolio	$840	80%	10.5

Note: NPV represents real (2000 dollars) net benefits discounted to beginning of CSBD program in 1994. Overall portfolio includes expenditures for all 24 projects.
N/M = not meaningful.

5.2.3 Distribution of Benefits

As the methodology and measurement portions of this report detail, economic benefits from the technologies funded by ATP's focused program were shared between the funded companies and the customers of their component-based products. The funded firms received the capital needed to develop products they could sell profitably in the market. Their customers were able to purchase and use component technology to lower their costs of developing software systems and information products and services.

As a part of our in-depth financial analysis of the eight CBSD firms profiled in this section, we estimated the value of both producer and consumer (customer) surplus over the relevant time horizon. Table 5-5 presents a breakdown of these performance measures for the eight case studies. We also briefly investigated the profitability of these firms, taking into account estimated producer surplus and their internal investments. We found that one firm incurred a net loss, two earned small economic profits, and five made significant profits. We would expect that the last five firms might experience competitive entry into their markets in the near future, with resulting downward pressure on their margins and net profits.

Table 5-5. Distribution of Benefits (in millions of 2000 dollars)

	Accrued to 12/2000	Projected in Future
Total consumer surplus	$247	$882
Total producer surplus	$5	$533
NPV of net benefits	$73	$767

Our evaluation of this ATP program was performed very early in the product life-cycle of many of the component-based products offered by the funded firms. Because of this advanced timing, many of the quantities and prices were future projections made by the principals we interviewed, rather than results of actual sales data. Reference to Table 5-2 shows that a large fraction of the total consumer and producer surplus is expected to accrue in the 2001 to 2004 time period. For this reason, we have broken down the portfolio's estimated surpluses into those already incurred and those projected into the future; these totals appear in Table 5-5.

5.3 ADDITIONAL CONTRIBUTIONS OF ATP

In addition to the social benefits discussed above, the funded firms reported that ATP made several contributions to their success that are more difficult to quantify. Standardized requirements for proposal writing, record-keeping, and progress reporting caused firms to be more thorough in their project planning and execution. The funding environment was well suited to the unpredictable nature of R&D, as both patience and flexibility on the part of ATP supported the firms' efforts. A "halo effect" increased credibility and lowered barriers to commercialization of the technologies developed.

This section integrates information from all eight of the companies that RTI studied. Although each company had a unique experience in developing its technology and in determining what applications and products to pursue, all of the companies shared numerous common experiences.

Proposal Writing and Organization

Even before companies received funding from ATP or started working on developing new technologies and products, they had already benefited from ATP's program. The application process is so rigorous and thorough that companies invested significant resources in determining where the current technological opportunities were, identifying potential solutions, and envisioning a marketing plan to fill a particular market segment. One firm stated that, by forcing companies to write effective proposals, ATP ensured that their firm put more thought into the industry, thus improving the research done in that sector.

The companies also said that the funding process was very fair. Some companies noted that, for most federal funding processes, the company seeking the funding needs to have a personal connection to the funding agency. Companies have to know someone at the agency, have worked with them in the past, or otherwise have had a connection. Respondents said that ATP made their funding decisions solely on the merits of the proposal, leading to better project selection and greater returns to the program.

ATP's Understanding of the Timing and Risks of Research

Years often elapse between the time an idea occurs and when a product can reach the marketplace. This is especially true when an entrepreneur is pursuing a groundbreaking line of research. However, most venture capitalists want to see a return on their investment within 2 years. In addition, they want a high degree of certainty about the probability of success of the research that they are funding.

In this competitive world of funding R&D of new technologies, projects that are high-risk/high-return but several years out are rarely funded by venture capital. ATP understands this problem. ATP's funding of the risky phases of technology development, where products are many years off, gives companies a chance to pursue ideas that the private sector will not fund. Tom Sawyer Software is an example of how ATP's patience generates returns. ATP first started funding Tom Sawyer in 1995, and they have only recently brought a product to market. However, the expected returns on this ATP project are so large that the benefit-cost ratio is projected to be 18. Without ATP's patient capital, a company like Tom Sawyer might never have been able to develop the unique technology embodied in their product.

ATP is able to circumvent the problems associated with the risk and uncertainty of a product by investing in a portfolio of projects. This investment approach allows them to pick from the best projects and not worry about the timing or risk that is associated with individual projects. ATP closely examines risk and uncertainty but, unlike a venture capitalist, does not eliminate a project just because it is risky.

Flexibility

ATP also understands the flexible nature of R&D. A company might be pursuing a strain of research that appears to be promising. At one point the company might realize that it needs to change the focus of the research in a new but related direction. ATP gives companies the flexibility to change their research midstream as conditions merit, with oversight from ATP's project management team and maintaining consistency with the original project goals.

For example, one of the firms that achieved technical success with its technology did not achieve commercial success with an early spin-out product. ATP allowed the firm to use the technical knowledge gained from the failed spin-out product to develop new project tasks, equivalent in technical and business merit and faithful to the overall project goals. The firm has yet to develop an effective commercial product, but it is still engaging in cutting-edge research that may reap dividends in the future.

Acceleration versus Feasibility

The companies that RTI examined all said that the ATP funding was instrumental in accomplishing their projects. SciComp, which developed a successful product called SciFinance, strongly felt that without ATP, its technology never would have developed in any form. Venture capitalists were unwilling to wait the seven years required for development, and no other institutions were willing to invest in SciComp. This firm now expects to generate revenues of over $10 million per year in the near future.

Other companies said that without the ATP funding they probably would have developed very different products, if any at all. For example, without the ATP funding one existing firm would have continued to sell a small piece of conventional software used by a handful of programmers. With the funding, this company has used component technology to develop a sophisticated and elaborate product that is used to control the pre-launch sequence for the space shuttle. Enabling technology that makes new product ideas feasible and possible, rather than simply accelerating the adoption date of one product, greatly increases the benefits attributable to ATP.

Halo Effect

The research community often views ATP as being an impartial judge of quality. When ATP funds a particular company it is vouching for the quality of the company's research ideas and their business and economic potential. When potential customers are making purchase decisions they may consider this information. This "halo effect" should translate to more sales and more opportunities for the funded companies.

Several companies said that the ATP funding generated a halo effect. The halo effect can emerge in different ways. One firm, which was a start-up company when it received the ATP grant, said that the ATP grant gave them credibility with venture capitalists and other financing mechanisms. A second start-up found out that ATP gave them enough credibility with their customers that they were able to charge a price premium for their higher-quality products. An established firm noted that the ATP funding gave them internal credibility with which they were able to generate more funds to conduct their R&D and expand the scope of the project.

5.4 CONCLUSION

The quantitative and qualitative analyses presented in this report demonstrate the significant impact of ATP's investment in component-based software technology. A large fraction (two-thirds) of the funded projects achieved their technical objectives; many of these firms and one notable JV proceeded to develop successful commercial products based on the technologies developed. ATP was credited by the firms with *enabling* the R&D efforts of many of these firms, *accelerating* the technology development process, and *increasing the probability* of technical and commercial success.

When the 24 funded projects are viewed as an investment portfolio, the social returns exceed any reasonable benchmarks for public or private investment. The estimated 80 percent internal rate of return reflects a substantial benefit to the nation in excess of the return to the companies funded and is an indication of inefficient capital markets for projects with high technical risks. The calculated net present value (NPV) of $840 million and benefit to cost ratio (B/C) of 10.5 suggest that the funds were a worthwhile expenditure of public funds.

References

Abramovitz, Moses. 1956. "Resources and Output Trends in the United States Since 1870." *American Economic Review, Papers and Proceedings* 46(2):5–23

Advanced Technology Program (ATP). 1997. <http://www.atp.nist.gov/www/press/9706cbs.htm>.

Advanced Technology Program (ATP). 1999. "ATP Focused Program: Component-Based Software." <http://www.atp.nist.gov/atp/focus/cbs.htm>. As obtained on December 23, 1999.

Akerlof, G. 1970. "The Market for Lemons: Quality Uncertainty and the Market Mechanism." *Quarterly Journal of Economics* 89:488–500.

Andersson, M., and J. Bergstrand. 1995. "Formalizing Use Cases with Message Sequence Charts." Unpublished Master's Thesis. Lund Institute of Technology, Sweden.

Aoyama, M. 1998. "New Age of Software Development: How Component-Based Software Engineering Changes the Way of Software Development." Paper presented at the 1998 International Workshop on Component-Based Software Engineering, Kyoto, Japan, April 25–26.

Austin, David, and Molly Macauley. 2000. *Estimating Future Consumer Benefits from ATP-Funded Innovation: The Case of Digital Data Storage*. NIST 00-790. Gaithersburg, MD: NIST.

Brown, A., and K. Wallnau. 1998a. "An Examination of the Current State of CBSE: A Report on the ICSE Workshop on Component-Based Software Engineering." Presented at the 20th International Conference on Software Engineering (ICSE), Kyoto, Japan, April 19–25.

Brown, A., and K. Wallnau. 1998b. "The Current State of Component-Based Software Engineering (CBSE)." *IEEE Software* September:37–47.

Brown, A., and K. Wallnau. 1996. "Engineering of Component-Based Systems." In *Proceedings of the Second International IEEE Conference on Engineering of Complex Computer Systems*, Montreal, Canada.

Brown, A. 1999. "Building Systems from Pieces with Component-Based Software Engineering." In *Constructing Superior Software*, Paul Clemens, ed. London: MacMillan Technical Publishing. Bachman, F., Bass, L., Buhman, C., Comella-Dorda, S., Long, F., Robert, J., Seacord, R., Wallnau, K., Volume II: Technical Concepts of Component-Based Software Engineering, Technical Report, CMU/SEI-2000-TR-008, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. <http://www.sei.cmu.edu/publications/documents/00.reports/00tr008/00tr008title.html>.

Cheesman, John, and John Daniels, 2000. *UML Components: A Simple Process for Specifying Component-Based Software*, Addison Wesley, October 2000.

Components Source. 2000. <www.componentsource.com>. Accessed on May 9, 2000.

CONSAD Research Corporation. 1997. *The Development of Advanced Technologies and Systems for Controlling Dimensional Variation in Automobile Body Manufacturing*. NISTIR GCR 97-709. Gaithersburg, MD: NIST.

Cuthill, B. 1997. "Advanced Technology Program—Focused Program in Component-Based Software: Supplemental Information for Program Competition 97-06." <http://www.atp.nist.gov/www/press/9706cbs.htm>. Accessed on December 23, 1999.

Dixit, Avinash K., and Robert S. Pindyck. 1994. *Investment Under Uncertainty*. Princeton, NJ: Princeton University Press.

Egan, E. 1999. "Application Districts: An Emerging Spatial Form in the Computer Software Industry." Paper presented at the Fourth Regional Science Policy Research Symposium, Chapel Hill, NC.

Egan, T. 1998. "Structural Change and Spatial Dynamics of the U.S. Software Industry." Paper presented at the Sloan Foundation Globalization Workshop, Durham, NC, April 15.

Ehlen, Mark A. 1999. *Economic Impacts of Flow-Control Machining Technologies: Early Applications in the Automobile Industry*. NISTIR 6373. Gaithersburg, MD: NIST.

Gallaher, M., B. Kropp, and T. Bingham. 2000. "Government's Role in Supporting the Development of Standards and Infratechnologies to Promote the Security of Information Systems." Research Triangle Park, NC: Research Triangle Institute.

Gompers, Paul, and Josh Lerner. 1999. "An Analysis of Compensation in the U.S. Venture Capital Partnership." *Journal of Financial Economics* 51(1):3–44.

Haines, G., D. Carney, and J. Foreman. 1998. "Component-Based Software Development/COTS Integration." *Software Technology Review*. <http://www.sei.cmu.edu/str/descriptions/cbsd_body.htm>. Accessed on December 16, 1999.

Jézéquel, J-M., and B. Meyer. 1997. "Design by Contract: The Lessons of Ariane." *IEEE Computer* 30(1):129–130.

Just, Richard E., Darrell L. Hueth, and Andrew Schmitz. 1982. *Applied Welfare Economics and Public Policy*, Chapter 4 and Appendix A. Englewood Cliffs, NJ: Prentice-Hall.

Kogut, P., and P. Clemens. 1999. "The Software Architecture Renaissance." <http://www.sei.cmu.edu/publications/articles/sw-arch-renaiss.html>. Accessed on May 16, 2000.

Lerner, Josh. 2000. "When Bureaucrats Meets Entrepreneurs: The Design of Effective 'Public Venture Capital' Programs." In *Managing Technical Risk: Understanding Private Sector Decision Making on Early Stage, Technology-based Projects.* Prepared for the U.S. Department of Commerce. Washington, DC: National Institute of Standards and Technology.

Mansfield, Edwin, J. Rapoport, A. Romeo, S. Wagner, and G. Beardsley. 1977. "Social and Private Rates of Return from Industrial Innovations." *Quarterly Journal of Economics* 91:221–240.

Martin, S.A., A.E. Kenyon, M.V. Bala, and T.H. Bingham. 1998. "A Framework for Estimating the National Economic Benefits of ATP Funding of Medical Technologies: Preliminary Applications to Tissue Engineering Projects Funded from 1990 to 1996." Prepared for the U.S. Department of Commerce, National Institute of Standards and Technology. Research Triangle Park, NC: Research Triangle Institute.

Meyer, B. 1999. "Beyond Objects: The Significance of Components." *Software Development Magazine*. <http://www.sdmagazine.com/ulm/beyondobjects/s9911bol.shtm>. Accessed on May 16, 2000.

Meyers, C., and T. Oberndorf. 1997. Open Systems: The Promises and the Pitfalls." Paper presented at the Fifth International Symposium on Assessment of Software Tools and Technologies, Pittsburgh, PA, June 3–5.

Morgenthaler, David. 2000. "Assessing Technical Risk." In *Managing Technical Risk: Understanding Private Sector Decision Making on Early Stage, Technology-Based Projects.* Prepared for the U.S. Department of Commerce. Washington, DC: National Institute of Standards and Technology.

Ning, J.Q. 1999. "A Component Model Proposal." Paper presented at the 1999 International Workshop on Component-Based Software Engineering, Los Angeles, CA, May 17–18.

Parrish, A., B. Dixon, and D. Hale. 1999. "Component Based Software Engineering: A Broad Based Model is Needed." Paper presented at the 1999 International Workshop on Component-Based Software Engineering, Los Angeles, CA, May 17–18.

Scherer, F.M. 1999. *New Perspectives on Economic Growth and Technological Innovation*. Washington, DC: Brookings Institution Press.

Silber, Bohne. 1996. *Survey of Advanced Technology Program 1990–1992 Awardees Company Opinion about the ATP and Its Early Effects*. Gaithersburg, MD: ATP.

Solow, Robert M. 1957. "Technical Change and the Aggregate Production Function." *Review of Economics and Statistics* 39(3):312–20.

Szyperski, Clemens. 1998. *Component Software Beyond Object-Oriented Programming*. Boston: Addison-Wesley.

Tassey, G. 1997. *The Economics of R&D Policy*. Westport, CT: Quorum Books.

Vonortas, Nicholas S., and Henry R. Hertzfeld. 1998. "Research and Development Project Selection in the Public Sector." *Journal of Policy Analysis and Management* 17(4):639–57.

Yin, Robert, K. 1984. *Case Study Research: Design and Methods.* pp. 47–54. Beverly Hills, CA: Sage Publications.

Appendix A: Methodology for Selecting Detailed Case Studies

Central to using case studies to gather data on the economic impact of ATP's support of CBSD projects was the need to ensure that the process used to select the projects for inclusion was rule-based rather than outcome-based. Establishing selection rules that are strictly followed during sample selection increases the external validity of the study (Yin, 1984).

Eight case studies were planned from the beginning. One approach to selecting the eight case study candidates would be a random selection from the 24 funded projects. Results from this approach would be the most statistically generalizable; however, by engaging in targeted sampling from the 24 projects more information could be extracted from the case studies.

Specifically, several projects that we knew a priori would not be fruitful could be eliminated from the set of 24. We used several screening steps to narrow the list of projects, thus improving the selection of the final case studies.

- ➤ The first screening step in selection of projects for the case studies was to eliminate all projects that were not technically successful. Four projects were eliminated from consideration because they failed to achieve technical completion based on information provided to ATP. It might seem that by doing this we would be placing a positive bias in our results. However, this is not the case. Projects that were failures are assumed to have a zero social benefit. The estimates for the entire focused program are made more conservative by analyzing the benefits and costs of the more successful projects in the case studies while also including the costs of the failures in the portfolio analysis.

- A second screen was used to eliminate projects that have not yielded commercial products or services. According to the evaluation methodology detailed in the text, a project does not yield tangible economic benefits unless the technology is commercialized. Within the CBSD focused program, two of the firms that achieved technical success elected not to commercialize their technology, and are thus effectively eliminated from the case studies.
- Two additional projects from the 1997 funding round were just completing their ATP-funded R&D phase as of December 2000, and, as such, had not reached commercialization. These projects are also eliminated from the case-study analysis because estimating their impact would rely solely on forecasts of market size and market potential rather than on observed data.

Table A-1 summarizes the screens and the number of projects dropped during each step.

Table A-1. Projects Dropped from Case Study List

Screen	Explanation	Projects Dropped
Technical failures	The project failed to complete its technological objective	Cubicon, Kestrel, Unisys, Sterling
Commercial failures	The project succeeded technically, but failed to reach commercialization	Lucent, Andersen
Still in development	In the ATP-funded R&D phase; has not yet yielded products or services	Data Access, Synquiry

After eliminating the eight projects listed in Table A–1, we were left with a potential list of 16 projects from which to select the eight individual benefit-cost case studies.

To take advantage of the heterogeneity across projects, we wished to make selections that capture this variability. Three major types of variations exist across the projects: year of funding, project type, and company size. To maximize the external validity of the study, we wished to select projects from all three rounds of funding (1994, 1995, and 1997) and all types of projects and company sizes represented in the program (joint ventures, single-company projects, projects involving start-up, and other small, medium, and large companies).

From the list of 16 available projects, only one was led by a medium-sized company (Intermetrics), one was a joint venture (led by Commerce One, a small company at the time of the ATP award) and one was led by a large company (Xerox); as a result, all three were included in the case study group. This selection left five projects to select from among the two classes of small firms, most of which were in the 1994 and 1995 funding rounds. We chose to select three start-ups and two other small firms to study, in line with their representation in the total program.

To balance the rest of the case study panel, then, three decisions had to be made. First, we wanted to choose two of the four 1994 start-ups with commercial products, which included SciComp, Aesthetic Solutions, APT, and Continuum. Secondly, we opted to select two of the four small firms funded in 1995, which were Analogy, Reliable Software Technologies, Tom Sawyer Software, and Real-Time Innovations. Finally, one of three 1995 start-ups would be appropriate, chosen from the group that included Extempo Systems, Hybrithms, and Semantic Design. RTI made the final selections based on conversations with ATP about which companies would be the most willing to share information regarding their product. Table A-2 presents the eight projects selected for inclusion in the case study analysis.

The final set of selected projects meets objectives of representation and heterogeneity, as is illustrated by the profiles presented in Tables A-3 and A-4. These tables show the number of projects selected out of the total number of projects in that category, and those that have reached commercialization.

Table A-2. Eight Projects Selected

Project	Study Title	Company Size	Project Budget ($000)	Product
Aesthetic Solutions Inc.	A Component Technology for Virtual Reality Based Applications	Start-Up	2,277	World Visions
Commerce One JV[a]	Component-Based Commerce: The Interoperable Future	Led by Start-Up	10,156	MarketSite
Extempo Systems Inc.	Component-Based Software for Advanced Interactive Systems in Entertainment and Education	Start–Up	2,537	Imp Characters
Intermetrics Inc.[b]	Debugging Component-Based Software for Enterprise Systems	Medium	2,670	Technology is licensed to Mercury Interactive
Real-Time Innovations	Component-Based Software Tools for Real-Time Systems	Small	2,618	ControlShell v6.0
SciComp Inc.	Automatic Generation of Mathematical Modeling Components	Start-Up	2,235	SciFinance
Tom Sawyer Software	Graph Visualization Technology	Small	2,919	Graph layout tool kit, graph editor tool kit
Xerox, Palo Alto Research Center	Reusable Performance-Critical Software Components Using Separation of Implementation Issues	Large	3,141	AspectJ

[a]Funded as a joint venture led by start-up firm CNgroup (later renamed VEO Systems), with partners CommerceNet, Business Bots, and Tesserae Information Systems, the last two also start-ups. CommerceNet is an industry trade group. CommerceOne later acquired VEO Systems, inheriting leadership of the ATP project.

[b]Later merged into Averstar, now part of Hewlett-Packard.

Source: ATP project data, including company close-out reports.

Table A-3. Number of Projects Selected from Each Funding Year Group

Year	Funded	Reached Commercialization	Selected
1994	11	6	3
1995	7	7	3
1997	6	3	2
Totals	24	16	8

Table A-4. Number of Projects Selected by Company Size or Project Type

Company Size or Project Type	Funded	Reached Commercialization	Selected
Start-Up	10	8	3
Small	6	5	2
Medium	2	1	1
Large	3	1	1
Joint Venture[a]	3	1	1
Totals	24	16	8

[a] The selected JV was led by a start-up firm. Of the two JVs not selected, one was led by a small firm, the other by a large corporation. Neither of these JVs completed their technical work.

Appendix B: Case Studies

B.1 AESTHETIC SOLUTIONS INC.

B.1.1 Company Description

Aesthetic Solutions is a small company located in Laguna Niguel, California. Although the main technological ideas developed by Gary Falacara, principal investigator at Aesthetic Solutions, existed before receiving the ATP grant, the company did not. After receiving funding from ATP, Aesthetic Solutions was able to conduct the R&D needed to accomplish the technical goals it put forth in its ATP proposal.

Company Demographics

Type of Firm	Start-up, private
Founding Date	1995
Company Size at Time of Award	4 employees
Headquarters Location	Laguna Niguel, CA

B.1.2 Technology Description

Goals of the ATP Project

Aesthetic Solutions hoped to build a component-based technology for virtual reality that would allow non-technical users to incorporate 3-D virtual reality images within their downstream applications. Aesthetic Solutions believed that current virtual reality programs were too expensive and only oriented to programmers, not users. Aesthetic Solutions hoped to lower the barriers to virtual reality use and, in the long term, turn virtual reality technology into a mass-market product.

At the start of the ATP project, Aesthetic expected that it would be able to achieve all of the technical goals that were needed for users to implement virtual reality into downstream applications. The potential applications of this technology would be numerous, including medical, training, and industrial applications. Ideally, end users could use the technology developed by Aesthetic and incorporate it into their applications. By incorporating this technology into their applications, end-users' productivity was expected to increase. For example, by including a 3-D interface within a medical training application, doctors would understand more clearly how the human body works and the implications of various medical procedures.

Technical Accomplishments

Aesthetic Solutions accomplished all of the technical goals set forth in their proposal to ATP. Their goals were achieved within the time frame they expected and reached the level of quality required. Aesthetic Solutions developed demonstrations of their technology, codes, and technology libraries that end users could immediately use within the applications they are developing. However, technical success has not led to commercial and economic success.

Aesthetic Solutions gives four reasons why they have been unable to achieve commercial success to date. Aesthetic Solutions thought that all of these barriers could be overcome if they could have developed an effective sales force. However, they lacked the financial resources to develop such a sales force.

First, the company was not able to convince businesses to adopt their technology. For several reasons, some of which were beyond their control, Aesthetic Solutions used a programming language called Virtual Reality Markup Language (VRML) within their technology. Consequently, potential end users would have had to change how they produced their own software to incorporate Aesthetic's new technology. Most end users were afraid to take this risk. Second, the market for 3-D technologies unexpectedly collapsed during this time period, which drastically reduced the number of potential customers. Third, because of changes to the market and the programming language used, venture capitalists were unwilling to invest additional funds in developing custom applications of the technology developed by Aesthetic. Fourth, the technology is ahead of its market. Although Aesthetic Solutions was able to develop the technology, end users lacked applications that could incorporate it; thus, sales of the product have been limited.

Products

Product	How Product is Sold
World Visions	Per unit
Component Visions	Never commercialized
User Visions	Never commercialized

Customer Use of the Product

Aesthetic Solutions developed three separate products, World Visions, Component Visions, and User Visions. World Visions is an assembly tool that consists of a point-and-click library of reusable 3D components. Although the product is currently being sold, Aesthetic Solutions does not expect that it will stay on the market much longer. Component Visions was a prototype development tool for developers to build their own components that never reached the market. The company had hoped this product would display the technology that it had developed. User Visions was expected to be the product that customers could use to apply the 3-D libraries developed by Aesthetic Solutions. However, this product has not reached the market and will not be fully developed.

Future Products

Aesthetic Solutions' future products using the ATP-funded technology are not clear. They continue to pursue 3-D technology applications that end users can incorporate into their applications, but they have not achieved great commercial success with the ATP-funded technology. Ideas and information that Aesthetic Solutions developed based on the R&D funded by ATP may not be included in future products.

B.1.3 Project Performance

In this section, we estimate the economic performance of the ATP-funded Aesthetic Solutions project. RTI conducted a structured interview with the principal investigator, Gary Falacara, and gathered information on product life spans, actual and projected sales, and total project costs. From this information, we derived demand curves for all products sold during their expected lives, and used these to estimate consumer and producer surplus and R&D expenditures. Based on these estimates, we calculated the net present value, benefit-cost ratio, and internal rate of return for the component software project.

Product Life Spans

Product	Life Span
World Visions	4 to 5 years
Component Visions	Never commercialized
User Visions	Never commercialized

R&D Expenditures for the Technology

Aesthetic Solutions first started working on their technology in 1994. Since then, they have spent $500,000 of non-ATP resources on developing the technology. ATP's actual payments to Aesthetic Solutions totaled $1,715,456 over the 1995 to 1998 time period. The combined ATP and company expenditures were set against the estimated consumer and producer surpluses in calculating net benefits.

Estimation of Performance Measures

To estimate the consumer and producer surplus benefits generated by Aesthetic Solutions’ product World Visions, we assumed that the marginal costs of reproducing the software are so close to zero that they are negligible, and the only costs that are important are the investments in R&D. This assumption is consistent with comments made by Aesthetic Solutions staff.

We first generated a per-year consumer and producer surplus benefit from the technology. Because the product is not expected to continue sales into the future, we maintained our conservative bias and did not include any benefits that might emerge after 2001. R&D expenditures and benefits were adjusted to 2000 dollars to remove effects of inflation. We estimated the benefit-cost ratio, net present value, and internal rate of return and present them in the table below.

Measures of Performance

Project Benefit-Cost Ratio	0.37
NPV of ATP and Private Investment	–$1,200,000
Internal Rate of Return	N/M
Producer Surplus	$600,000
Consumer Surplus	$50,000

N/M = not meaningful.

B.1.4 Qualitative Benefits Attributable to ATP

General Impacts of ATP Funding on the Company

ATP funding had several qualitative impacts. Aesthetics Solutions takes advantage of the fact that ATP funded their project in communicating to potential customers and partners. For example, companies like Apple, Microsoft, and Motorola became interested in Aesthetic Solutions after learning about the ATP project. This halo effect built the credibility of Aesthetic Solutions as an effective company.

Secondly, the ATP grant created important business connections that Aesthetic Solutions used to improve its business operations. Finally, Aesthetic Solutions believes that it would have taken 10 years for them to reach market without support from ATP.

General Impacts of ATP Funding on the Market

Aesthetic Solutions believes the entire market for virtual reality programs has been accelerated and that with improved virtual reality programming, everyone will be able to incorporate virtual reality into programs.

B.2 COMMERCE ONE JOINT VENTURE

Determining the economic benefits from the ATP-funded Commerce One Joint Venture was a challenge. The firms involved have gone through several organizational changes since the inception of the project, and the leading firm, Commerce One, went public in 1999 and was subsequently caught up in the Internet bubble and its aftermath. The following sections contain RTI's valuation methodology for the products created from the technology produced; several other possible approaches could also be justified.

A brief history of the joint venture shows some of the changes that the companies have experienced:

- The joint venture was formed in 1997 by an e-commerce trade association, CommerceNet, and three start-up firms, VEO Systems, Tesserae Information Systems, and BusinessBots. VEO, initially called CNGroup, was formed from the R&D arm of the trade association and became the joint-venture lead.
- From October 1997 through December 1999, the joint venture led by VEO Systems received about $4.8 million in matching funds through the ATP program.
- From January 1997, when VEO Systems was founded, through January 1999, when it was acquired by Commerce One, VEO Systems raised roughly $5 million in other sources of funds, mostly through investment by strategic corporate partners. These funds provided the “matching funds” share for VEO Systems in the ATP program.
- In January 1999, VEO Systems was acquired by Commerce One in a stock swap that valued VEO Systems at $22 million. Based on interviews with Commerce One, most of the value of VEO Systems in the Commerce One acquisition was for technology developed under the ATP project.
- In July 1999, Commerce One went public at $21 per share.
- Tesserae, which had been renamed Cadabra, Inc., was purchased by GoTo.com on January 31, 2000. The purchase price was $8 million in cash and shares of GoTo.com then valued at $250 million.
- In January 2000, Commerce One briefly traded at a split-adjusted price of about $700 per share.
- As of November 2001, the stock price of Commerce One was $2.45, and the future of the company was uncertain.

- GoTo.com, now called Overture Services, wrote off its investment in Cadabra's technology during 2001. BusinessBots appears to have stopped operating as well.

B.2.1 Demographics of Joint Venture Partners

Name of Firm	VEO Systems	BusinessBots	Tesserae	CommerceNet
Type of Firm	Start-up	Start-up	Start-up	Industry Trade Association
Founding Date	1997	1997	1997	1994
Company Size	(See notes)	Small	(See notes)	Small
HQ Location	Mountain View, CA	San Francisco, CA	San Mateo, CA	Palo Alto, CA

Notes: VEO Systems was purchased by Commerce One in January, 1999.
Tesserae, renamed Cadabra, was purchased by GoTo.com in January, 2000.

B.2.2 Technology Description

Goals of the ATP Project

The overall goal for the four joint-venture partners that participated in the ATP project was to develop new architectures and technologies for building Internet marketplaces. The firms hoped to develop a product that would let businesses build on each other's Internet services, using them as components to create innovative virtual companies, marketplaces, and trading communities. The fundamental problem was how to combine heterogeneous documents and other sources of information coming from a wide variety of Web sites, catalogs, databases, ERP systems, news feeds, and other applications. This information invariably exists in incompatible formats and semantic models, which means that the information one business provides to another cannot be used directly by the recipient's computing applications.

Technical Accomplishments

The Commerce One JV achieved significant technical breakthroughs to transform new ideas into complementary, scaleable, and commercially viable tools and software platforms for creating Internet marketplaces. The key achievements include the following:

- XML-based Common Business Library (xCBL);

- “intelligent” tools for transforming Web content into XML and other structured formats that can be processed by computers to support application integration, comparison shopping, and agent-based process invocation; and
- architectures and concepts for describing Internet marketplaces, their participants, the services they offered, and the information needed to operate them (the eCo Framework).

Products

Product	How Product is Sold
MarketSite (marketplace platform)	Software license; highly customized product

Customer Use of the Product

As a result of VEO System's leadership and its subsequent incorporation into Commerce One, Commerce One was the firm to commercialize the primary technical results produced by the JV. Customers would purchase the virtual market products from Commerce One and then develop trading markets where different suppliers and customers would be able to purchase products from anyone in the virtual market. Commerce One’s XML-based MarketSite platform is supporting many of the largest business-to-business procurement communities in operation today, including Exostar (aerospace industry), Covisint (automotive), Forest Express (paper products), Quadrem (mining and metals), and Trade Ranger (energy and petrochemicals). Because MarketSite is Commerce One’s main product, we used it to estimate the economic performance of the project

Future Products

Currently Commerce One is not planning on developing any new products. Rather, they are planning on refining and improving their existing suite of products.

B.2.3 Project Performance

In this section, we estimate the economic performance measures for the ATP-funded Commerce One joint venture project. In our effort to obtain the information needed to perform the valuation, RTI conducted a structured interview with Robert Glushko, the principal investigator from the lead firm. We asked questions about product life spans, actual and projected sales, and total project costs. Because GoTo.com has written off their JV-developed technology and BusinessBots is no longer operating, we did not attempt to value additional products from the JV.

Although we obtained less information from the interview than we would have liked, Commerce One's status as a publicly-traded corporation meant that a great deal of additional data was available from public sources. The information we obtained was sufficient to derive demand curves for the products sold, and to estimate consumer and producer surplus and R&D expenditures. Based on these estimates, we calculated the net present value, benefit-cost ratio, and the internal rate of return for the component software joint venture.

Product Life Spans

Although other companies tried to develop similar products based on Commerce One's success, Commerce One is clearly the leader in this area. However, because other companies are starting to develop competing projects, we limited our analysis through the year 2003. After this time, we conservatively assume that competing products will have emerged and no longer include benefits from Commerce One.

Product	Life Span
MarketSite	Until 2003

Commerce One's MarketSite product differs from other component-based products because of the significant amount of customization that is required to develop a "marketplace." This reality dictates the approach that is used to determine the price of Commerce One's products. Each marketplace that they establish is a unique entity; although the technology overlaps across the products, a significant amount of customization must occur, which makes determining a marginal cost impossible. According to data from annual reports for Commerce One, the average cost of developing a marketplace is just over $1 million. This is a one-time cost that must be netted out of the total benefits generated during the year that a new marketplace is developed.

R&D Expenditures for the Technology

All of the JV partners provided funds to the technology development effort. A substantial share of the amount VEO Systems received as a result of its acquisition by Commerce One was also invested in the ATP-funded technology.

Year	Non-ATP Expenditures	ATP Expenditures
1997	$1,936,815	$235,185
1998	$4,801,945	$2,037,055
1999	$17,968,240	$2,527,760
Total	$24,707,000	$4,800,000

Estimation of Performance Measures

Commerce One is generating large public and private benefits from its MarketSite product. From estimates of the number of units sold and price, we estimate that in 2000 alone a consumer surplus of over $150 million was generated; for 2001, we project that surplus to rise to $200 million. Even with substantial reductions in new implementations forecast for the next two years, the cumulative benefits by the end of 2003 will exceed $800 million in inflation-adjusted (2000) dollars.

Measures of Performance

Benefit-Cost Ratio	39
NPV of ATP and Private Investment	$789 million
Internal Rate of Return	363%
Producer Surplus	$220 million
Consumer Surplus	$615 million

B.3 EXTEMPO SYSTEMS, INC.

B.3.1 Company Description

Extempo makes smart, friendly, helpful interactive characters providing personalized customer care on the Internet. Their self-defined market niche was originally edutainment, but has changed to e-commerce. Serving business-to-consumer and business-to-business companies, Extempo offers two solution packages. Web Staff Characters receive and serve customers who visit a business' Web site, kiosk, or other destination site. Interactive Messenger Characters visit and serve a business' customers through electronic links carried in e-mail or other proactive channels. For each of these solution packages, Extempo offers custom characters built to specification, a growing roster of market-targeted characters that can be customized to particular applications, and turnkey character solutions.

Company Demographics

Type of Firm	Start-up, private
Founding Date	1995
Company Size at Time of Award	2 employees
Headquarters Location	Redwood City, CA

B.3.2 Technology Description

Goals of the ATP Project

The major goal of the Extempo project was to develop an interactive character that customers could use to develop their Web pages. Ideally, Extempo wanted to develop a fully interactive character that can be applied in numerous different situations.

Technical Accomplishments

Extempo was successful in developing a software architecture model and initial examples of plug-and-play characters. These characters have been implemented by numerous different Web sites, ranging from sites produced by large industrial companies such as Proctor and Gamble to small Internet startups such as

Petopia. In addition to developing the characters, Extempo also evaluated user experiences to improve the characters' functionality and usability.

Products

Product	How Product is Sold
Web Staff	Sold on a subscription basis
Messenger	Sold on a subscription basis

Customer Use of the Products

Extempo's interactive characters are developed for or licensed to e-commerce companies for use on their Web sites. The characters non-intrusively solicit customers for required information or provide information on a specialized product or service.

Extempo's characters communicate easily through conversation and gesture. They inform and assist customers, answer their questions, and make helpful suggestions. Extempo-powered characters can provide better services than their human counterparts in some situations. The characters offer immediate, unlimited time and attention to each individual customer. They learn about their customers and remember them, improving their personalized services on every visit. Extempo-powered characters do not vary in their presentation to customers and are always friendly, cheerful, and entertaining. As a result, customers may visit a site more often, stay longer, and complete more transactions. In theory, by building bonds of loyalty and trust based on a shared history of service and social exchange, Extempo-powered characters build relationships with their customers over time the same way that humans do.

Several potential improvements to a user's product may occur from using Extempo characters. Users may become more comfortable using the Internet, which would increase the frequency of transactions they conduct over the Internet. Additionally, visitors to Web sites will be able to use them more efficiently because they will not have to find their way around a Web site unaided and will have input on products that best serve their needs. This increased automation may lower total transactions costs.

Future Products

Although Extempo is planning on increasing the functionality, variety, and uses of the characters they have created, they have no plans for new products. Extempo has recently been able to attract new venture capital funding, which might create new product opportunities.

B.3.3 Project Performance

In this section, we estimate economic performance measures for the ATP-funded Extempo project. RTI conducted a structured interview with Barbara Hayes-Roth, the principal investigator from the start-up firm. We asked questions about product life spans, actual and projected sales, and total project costs. From this information, we derived demand curves for all products sold during their expected lives, and used these to estimate consumer and producer surplus and R&D expenditures. Based on these estimates, we calculated net present value, the benefit-cost ratio, and the internal rate of return for the component software project.

Although Extempo produced two products using the ATP-funded technology, Web Staff and Messenger, we focused on estimating the benefits from Web Staff only. Web Staff is the main product that Extempo sells. It costs substantially more than Messenger, and Messenger has numerous competing products that could provide very similar services. Hence the unique benefit of the ATP investment in Extempo is the product Web Staff. Throughout this analysis we provide additional information regarding sales and price information for Messenger.

Product Life Spans

Extempo has such a technical lead in this market that they do not expect any competition for their products until 2006. Even though the current products could stay on the market for several years beyond 2006, we maintained a conservative bias for this analysis and assumed that the product will stay on the market for only 3 years.

Product	Life Span
Web Staff	Until 2006
Messenger	Until 2006

Competing Technologies

Two companies have the potential to produce similar products. Both Artificial Life, located in Boston, MA, and eGain Communications, located in Sunnyvale, CA, are conducting R&D to develop new artificial intelligence components that designers of Web pages can use in their products. However, according to Extempo, both of these companies are currently lagging in their technology development. Extempo believes they will maintain technical superiority until 2006 at a minimum.

R&D Expenditures for the Technology

Extempo spent a total of $550,000 of its internal funds during the project time frame, and received almost $2,000,000 from ATP. Extempo has continued to pursue R&D and commercialization since 1999. Their additional internal funding of $300,000 and external venture capital financing of $1.3 million are included in the cost calculations because they were used to support commercial launch of their products. We assumed even annual spending of internally generated funds during the project, as the summary table below illustrates.

Year	Non-ATP Expenditures	ATP Expenditures
1995		$351,491.62
1996	$137,500	$481,010.18
1997	$137,500	$862,957.83
1998	$437,500	$294,824.37
1999	$1,237,500	
2000	$200,000	
Total	$2,150,000	$1,990,284.00

Estimation of Performance Measures

To estimate the consumer and producer surplus benefits from the ATP investment in Extempo, we again assumed that the marginal costs of reproducing the software are negligible, and the only costs that are important are the investments in R&D. To estimate the benefits from Web Staff, we first generated a per-year benefit from the technology. As is discussed above, we carried the benefits from sales out for three years past the launch date, a conservative assumption. All R&D expenditures and benefits were first adjusted to 2000 dollars to remove effects of inflation. We estimated the benefit-cost ratio, net present value, and internal rate of return and present them in the table below.

Measures of Performance

Benefit-Cost Ratio	0.63
NPV of ATP and Private Investment	–$1,220,000
Internal Rate of Return	N/M
Producer Surplus	$1,375,000
Consumer Surplus	$690,000

N/M = not meaningful.

B.3.4 Qualitative Benefits Attributable to ATP

General Impacts of ATP Funding on the Company

Extempo did not exist as a company until the ATP award. After receiving the ATP award, the company began their R&D process and created the Web Staff product. The biggest additional benefit that Extempo has received from the ATP funding is the attraction of a significant inflow of venture capital dollars. Although this funding may allow them to develop additional products in the future, the uncertain nature of the ATP R&D makes it impossible to value additional economic benefits for this study.

General Impacts of ATP Funding on the Market

According to Extempo, ATP's investment in their technology was critical in developing the component-based market for intelligent characters. Without ATP, these products may have been developed eventually but not until 2007 at the earliest.

B.4 INTERMETRICS, INC.

B.4.1 Company Description

At the beginning of its ATP project, Intermetrics, Inc. was an established, privately held firm, founded in 1969 in Burlington, MA. In February 1998 Intermetrics merged with Pacer Infotec to form AverStar, Inc. AverStar was purchased by Titan Corporation in June of 2000. As of February 1, 2001, Hewlett-Packard entered into a 1-year agreement with Titan to become the sole licensee of the technology that was developed using ATP funds. At the end of the 1-year agreement, both parties have the option to continue the agreement or dissolve the relationship. Interestingly, given all of the changes in corporate ownership, the research team that received the ATP award has stayed relatively intact with the key group of 8 to 12 Intermetrics employees currently working at Hewlett-Packard.

Company Demographics

Type of Firm	Established, private
Founding Date	1969
Company Size	Medium ($54M in 1997 sales)
Headquarters Location	Burlington, MA

B.4.2 Technology Description

Goals of the ATP Project

At the start of the ATP project, Intermetrics hoped to accomplish two goals. First, the company wanted to develop a diagnostic tool that would capture and display all of the software component interactions that occur within an e-business transaction performed across a system containing multiple platforms (for example, COM and CORBA). This tool would let the company know where any bottlenecks, slowdowns, or faults were within the system. This information would improve transaction time for existing transactions and let the company identify problems and bring its applications back online as soon as possible following a system crash. Second, the company wanted to integrate the diagnostic tool with evaluation capabilities. This achievement would have created a product that could provide more diagnostic feedback regarding the operations of an e-business transaction.

Once their resulting product had been developed, Intermetrics had planned on licensing the product to an original equipment manufacturer (OEM) that would in turn sell it within that company's product. Intermetrics lacked a large enough distribution and sales force to effectively sell the product on its own, so they thought working in a consulting role rather than as a direct seller would generate greater returns.

Technical Accomplishments

Intermetrics effectively accomplished the first goal: they developed the technology supporting two products. In a licensing agreement with Mercury, Intermetrics released a product called eWatch in 2000. After that relationship ended, and with the new relationship with Hewlett-Packard, the ATP-funded technology is expected to be incorporated into a product called OpenView. Approximately one-third of the technology used in OpenView is the ATP technology; the rest is technology previously developed by Hewlett-Packard. Consequently, we considered two potential analysis scenarios. If the ATP technology was only responsible for one-third of the benefit from the products, only one-third of the benefits should be considered in the economic returns of the ATP project. Alternatively, the products might not have existed without ATP's help; if this is true, all of the benefits from the technology should be considered. Maintaining our bias towards conservative estimation, we attributed only one-third of the benefits of the products to the ATP project.

Intermetrics did not achieve the second goal of integrating their tool with evaluation capabilities, and they do not anticipate achieving it in the future. The Intermetrics group working on the technology decided that the challenges of accomplishing the second goal were too great and that they needed to focus on turning the technological accomplishments that they had achieved into a marketable product.

Products

Product	How Product is Sold
eWatch	Per unit
OpenView	Per number of URLs

Customer Use of the Products

The ATP-funded technology was incorporated into eWatch, a product for Mercury and is expected to be incorporated into OpenView, a Hewlett-Packard product, very soon. Once a user has purchased eWatch, two options are available. First, the users of the product can run tests on an entire e-business transaction application (e.g., the software, the routers, the switches) and determine where the bottlenecks are within the application. This testing allows the user of the technology to create faster, better, and cheaper e-business applications.

Second, if an e-business application crashes, system programmers could apply the product to the application to pinpoint the exact location of the error or fault. This approach would significantly decrease the amount of time required for a company to get their e-business application up and running again. This benefit could range from simply saving a few transactions that would have been lost to saving an entire business that cannot afford to lose any transactions.

Future Products

If the relationship between the Intermetrics team developing the technology and Hewlett-Packard is successful, Hewlett-Packard is expected to incorporate the ATP-funded technology into future versions of OpenView.

B.4.3 Project Performance

In this section, we estimate economic performance measures for the ATP-funded Intermetrics project. We gathered information on product life spans, actual and projected sales, and total project costs. Based on these estimates, we calculated the net present value, benefit-cost ratio, and internal rate of return for the ATP-funded project.

Competing Technologies

Based on discussions with Jeff Rees, principal investigator at Intermetrics, no competing technologies that exist now are similar to the ATP-funded technology. However, he believes that a similar technology would have been developed within the next three years if ATP had not invested in Intermetrics. Based on this information, the effect of the ATP project will be a 3-year acceleration of the benefits from the products that incorporate the technology. Again maintaining our bias towards conservatism, we assumed the window of market opportunity is constant at three years, and the only effect is an acceleration of the benefits by three years.

The closest competing technology is an application response time metric (ARM). This intrusive technology gives software users the ability to estimate the total time an e-business transaction takes, but it fails to provide any of the necessary information to improve the transaction or provide any information to get a Web page application functioning again.

R&D Expenditures for the Technology

From the project's inception in 1997, Intermetrics spent close to $3,000,000 of internally-derived capital. ATP provided about $1,650,000 in funding, mostly in calendar years 1998 and 1999. We used the schedule of payments provided to us by ATP in our benefits calculations, and estimated smooth spending of the internal funds from 1997 through 2001.

Estimation of Performance Measures

To estimate the consumer and producer surplus from eWatch and OpenView, we assumed that the marginal costs of reproducing the software are so close to zero that they are negligible, and the only costs that are important are the investments in R&D. Because R&D costs cannot be separated into investments in eWatch and investments in OpenView, the individual product benefits were also combined. The table below presents the various performance measures calculated.

Measures of Performance

Benefit-Cost Ratio	9.6
NPV of ATP and Private Investment	$29.6 million
Internal Rate of Return	103%
Producer Surplus	$31 million
Consumer Surplus	$3.7 million

B.4.4 Qualitative Benefits Attributable to ATP

General Impacts of ATP Funding on the Company

ATP funding had several qualitative impacts. Intermetrics frequently uses the fact that ATP funded them as a way to establish the company's credibility in marketing its products. Intermetrics also said that one of the major benefits from ATP funding was that ATP truly understands the research process. ATP was flexible when the company's goals changed based on the information they discovered during the R&D process. This flexibility to refocus the enabling technology in a way consistent with ATP criteria allowed Intermetrics to come up with better technology faster.

General Impacts of ATP Funding on the Market

In addition to accelerating the commercialization date for OpenView, Intermetrics believes that the ATP grant effectively accelerated the entire market for similar products by at least 1 year. The potential surplus generated from this acceleration could be very significant.

B.5 REAL-TIME INNOVATIONS, INC.

B.5.1 Company Description

Real-Time Innovations, Inc. was a spin-off from Stanford University and has developed into a leading developer of new tools and architectures for the growing real-time software market. The main technology that the company has developed is called Control Shell. Control Shell was originally conceived by Stanley Schneider while he was a graduate student at Stanford. Since that time the technology has been incorporated in seven major product versions; version 6.0 was released in 1998, and version 7.0 was released in 2001.

Company Demographics

Type of Firm	Small, private
Founding Date	1991
Company Size at Time of Award	5 employees
Headquarters Location	Sunnyvale, CA

B.5.2 Technology Description

Goals of the ATP Project

The goal of the Control Shell project was to develop a tool for building complex real-time systems. The company hoped to develop a library of tools focused on vertical markets; for example, building a complex real-time system for robotics, aerospace, or vehicles. Control Shell is a program with a graphical interface that lets users develop and manipulate their real-time systems.

Technical Accomplishments

Although earlier versions of Control Shell existed, the product that was released based on the technology developed during the ATP project was such a dramatic improvement that it should be considered a different product. Until version 6.0, Control Shell was a platform for building monolithic applications rather than for building separate components which could be configured into systems. Version 6.0 was the first product that was commercially released, the first post-ATP product, and the product that we analyzed for this case study. Real-Time Innovations has produced several other products, but they were not based on the ATP-funded technology and are excluded from this analysis.

Control Shell Version 6 controls real-time systems. The technology allows users to have a graphical interface they can use to control all of the real-time systems that exist within the application of interest.

The graphical interface lets them monitor, adjust, or alter their real-time system.

Products

Control Shell can be sold in two different forms, either per seat or per run-time.

Product	How Product is Sold
Control Shell 6.0	Per seat and per run-time
Control Shell 7.0	Per seat and per run-time
Control Shell 8.0	Per seat and per run-time

Customer Use of the Products

Customers in numerous industries have adopted Control Shell. Once they purchase Control Shell and the corresponding library of information they need, they are able to implement the entire system. The customer purchases the product and then develops the specific real-time applications around Control Shell. Customers have ranged from Lockheed Martin to hospitals. Use of Control Shell has ranged from controlling some of the prelaunch activities of the Space Shuttle to developing testing systems for medical students.

B.5.3 Project Performance

In this section, we estimate the economic performance measures for the ATP-funded Real-Time Innovations project. We conducted a structured interview with the principal investigator from this small firm. We asked questions about product life spans, actual and projected sales, and total project costs. From this information, we derived demand curves for all products sold during their expected lives, and used these to estimate consumer and producer surplus and R&D expenditures. Based on these estimates, we calculated the net present value, benefit-cost ratio, and internal rate of return for the component software project.

In the desire to create a conservative estimate, we included benefits only from Control Shell 6.0, which is currently on the market. Real-Time Innovations expects to release new versions of Control Shell well into the future. Qualitative information that was discussed regarding future products is presented below.

Product Life Spans

Product	Life Span
Control Shell 6.0	4 years
Control Shell 7.0	4 years
Control Shell 8.0	4 years

Competing Technologies

Currently, there are no technologies that compete directly with Control Shell. However, Real-Time thinks that in 2 to 3 years new competitors will emerge.

R&D Expenditures for the Technology

Real-Time estimates that it contributed $2,000,000 to the development effort during the ATP-funded project, while ATP's contribution was $1,910,000. During this time, the firm was creating products that embodied the new technology, and releasing them as upgraded versions of Control Shell. For this reason, Real-Time Innovation’s R&D expenditures included in our analysis include costs for both technology development and product development beyond the ATP project.

Estimation of Performance Measures

To estimate the consumer and producer surplus benefits generated by Control Shell 6.0, we assumed that the marginal costs of reproducing the software are so close to zero that they are negligible, and the only costs that are important are the investments in R&D. According to Real-Time Innovations, Control Shell 7.0 never would have been successfully released without ATP funding as a component-based framework. In addition, the company believes no other companies could release a similar product. Therefore, the total benefit from Control Shell is attributable to the ATP-funded project. R&D expenditures and benefits were adjusted to 2000 dollars to remove effects of inflation. We estimated the benefit-cost ratio, net present value, and internal rate of return and present them in the table below.

Measures of Performance

Benefit-Cost Ratio	1.8
NPV of ATP and Private Investment	$2,060,000
Internal Rate of Return	31%
Producer Surplus	$3.9 million
Consumer Surplus	$1.0 million

B.5.4 Qualitative Benefits Attributable to ATP

General Impacts of ATP Funding on the Company

As noted above, without ATP support, Control Shell 7.0 technology never would have resulted in a true product. Rather, it would have continued to be a highly specialized tool that only a few programmers would have used. ATP gave the company the “guts to take a risk.” ATP increased their confidence that their product could be technically successful and ensured a steady stream of funds to allow them to conduct their research.

General Impacts of ATP Funding on the Market

According to Real-Time Innovations, one of the biggest benefits of ATP funding a particular sector of the economy occurs during the proposal writing process. They cite the ATP proposal process as the best in the research world because of its honest approach to evaluating proposals. The company notes that each company is treated equally and fairly, which is far different from the rest of the federal funding world. By forcing companies to write effective proposals, more thought is put into the R&D planning, which improves the research done.

B.6 SCICOMP INC.

B.6.1 Company Description

SciComp, a start-up firm in Austin, TX, creates gains in productivity and value for its customers by directly harnessing the power of mathematics and scientific computing through automation. The company provides software synthesis tools that empower their customers to produce complex mathematical software that satisfies their needs without manual computer programming.

SciComp's proprietary software synthesis technology allows engineers and scientists in a wide variety of application areas to state their problems in a concise and easy-to-learn language called ASPEN (algorithm specification notation). From their brief ASPEN specifications, SciComp's synthesis technology tools generate thousands of lines of C or Fortran code. SciComp has incorporated this technology into SciFinance. SciFinance is a software system that automates the pricing of complex derivative and option structures. Users briefly specify their models in precise financial and mathematical terms. Then SciFinance generates the C code that implements the user's specifications. Essentially, it automates various parts of a traditionally labor-intensive production process. SciComp also supports strategic partnering to develop new distribution channels, markets, and application areas.

Company Demographics

Type of Firm	Start-up, private
Founding Date	1994
Company Size at Time of Award	1 employee
Headquarters Location	Austin, TX

B.6.2 Technology Description

Goals of the ATP Project

Based on the funding that they received from the ATP, SciComp hoped to develop a technology to automatically synthesize computer code for scientific applications from high-level specifications. The main focus of the ATP project was to make the development of difficult but routine pieces of code commonplace and simple.

Technical Accomplishments

SciComp was able to develop a high-level specification language based on the research that they conducted under the ATP award. SciComp developed the methods, interfaces, and performance templates needed for the algorithms used in their products. They created an extensible specification language and developed a practical synthesis technology that worked more quickly and improved productivity ten-fold for companies using their products. All of the innovations developed by SciComp under the ATP award are incorporated in the company's SciFinance product.

Products

Product	How Product is Sold
SciFinance v1.x	SciFinance v1.x was licensed on a per-seat basis. Each license covered a minimum of a 1-year time period. Access to SciFinance v1.x was password-enabled. Code generated by SciFinance was licensed for use by the customer for internal purposes only, not for resale.
SciFinance v2.x SciPDE Module SciMC Module	SciFinance is licensed on a per-seat basis and is available with either or both modules. Each license covers a specific time-period on a subscription basis, either single or multiple months. Access to SciFinance is password-enabled. Code generated by SciFinance is licensed for use by the customer for internal purposes only, not for resale.
SciXL	SciXL has a multiple license structure: ➤ Developer license—a permanent, password-enabled license issued on a per-seat basis, which can be used to generate Microsoft Excel add-ins. ➤ User license—a license that allows nondeveloper license seat holders to use SciXL-generated Microsoft Excel add-ins.

Customer Use of the Products

After purchasing SciFinance, customers can specify the type of code they need to solve a particular problem. They enter the specifications (e.g., what needs to be solved, what computer language, how many iterations), and SciFinance automatically generates the code to fit into their existing computer system. This code is essentially cut and pasted into their existing software source code and is ready to compile and use. SciXL is another product that incorporates ATP-funded technology. It is similar to SciFinance, but it has a different licensing structure because it was developed with a third party. Thus, we treat it separately in our benefits estimation.

Customers using these products can now quickly develop complex analytical models so that staff can work on other critical analytic tasks. Customers also benefit by receiving consistent, high-quality code in a fraction of the time it would have taken to develop their code independently.

Future Products

SciComp is planning on expanding the technology that they have developed to numerous other potential applications. Any application that requires in-depth and intense numerical modeling is a potential market for SciComp. For example, automobile design, development of electronics, chemistry, civil engineering, mining, physics, education, and the military are all activities and sectors that could benefit from using the technology. However, SciComp has yet to develop specific applications for these sectors.

B.6.3 Project Performance

In this section, we estimate economic performance measures for the ATP-funded SciComp project. We gathered information from Dr. Elaine Kant, the firm's founder and project principal investigator, on product life spans, actual and projected sales, and total project costs. From this information, we derived demand curves for each product sold during their expected lives, and used these to estimate consumer and producer surplus and R&D expenditures. Based on these estimates, we calculated the net present value, the benefit-cost ratio, and the internal rate of return for the software project.

Product Life Spans

Product	Life Span
SciFinance v1.x	2 years
SciFinance v2.x	4 years (until 2004)
SciXL	4 years (until 2004)

Competing Technologies

According to SciComp, there are no direct competitors to the suite of products they have developed based on the ATP technology. Although there are numerous ways to develop code for a specific project or application, few of them have been automated, and none in the application area that SciComp is addressing. Because of its automation, the suite of SciComp products is a radical departure from previous products.

R&D Expenditures for the Technology

ATP contributed a little more than $1,900,000 during the 1995 to 1997 period. SciComp, as a small firm, could only provide about $255,000 during that same time frame. Once the commercial potential of their technology became apparent, SciComp was able to attract external funding of $1,000,000, which aided them in launching the products discussed above.

Estimation of Performance Measures

To estimate the consumer and producer surplus benefits, we assumed that the marginal costs are zero, and the only costs that are important are the R&D expenditures. SciFinance is downloaded from a Web site, so there are very little additional marginal costs associated with getting the product to the consumer.

We first generated a per-year consumer and producer surplus benefit from the technology. R&D expenditures and benefits were adjusted to 2000 dollars to remove effects of inflation. We estimated the benefit-cost ratio, net present value, and internal rate of return and present them in the table below.

Measures of Performance

Benefit-Cost Ratio	7.6
NPV of ATP and Private Investment	$21 million
Internal Rate of Return	51%
Producer Surplus	$17 million
Consumer Surplus	$8.6 million

B.6.4 Qualitative Benefits Attributable to ATP

General Impacts of ATP Funding on the Company

SciComp states that without the ATP funds, it would have been impossible for the company to develop its technology and produce these products. The company simply lacked the resources and was unable to raise venture capital due to the 7-year lag between initial investment in the technology and the time they would be able to generate positive economic returns from investment.

General Impacts of ATP Funding on the Market

SciComp believes the CBSD market would have developed without the ATP investment, but would have been substantially delayed. SciComp was uncertain about the extent of the delay but thought the delay in developing the overall component-based market would have been substantial.

B.7 TOM SAWYER SOFTWARE

B.7.1 Company Description

Graph-drawing theory is a rapidly growing academic discipline. More companies are using algorithms and other analytical techniques to display large amounts of data as graphs for easier interpretation. Tom Sawyer Software, a small company located in Berkeley, CA, was funded by ATP to develop component-based graphing tools. Tom Sawyer develops information systems and software applications that companies can use to manage the flow of information throughout their company. Specifically, Tom Sawyer focuses on developing graphs that biochemical, engineering, and financial companies can use to more easily and efficiently display information about their workflow, production process, or other flows of complex information.

Company Demographics

Type of Firm	Small, Private
Founding Date	1991
Company Size at Time of Award	Approximately 10 employees
Headquarters Location	Berkeley, CA

B.7.2 Technology Description

Goals of the ATP Project

The goal of the ATP project was to create automated graph layout technology that could become a "standard" component of large systems. Tom Sawyer hoped to advance the state of the art in relational graph visualization. Specifically, they hoped to set the standard for three types of graphing techniques: constraint-based graphs, incremental layouts, and drawing of very large graphs (i.e., graphs that rapidly change or contain numerous items). By accomplishing these technological improvements they would be able to advance the level of graphing capabilities, enabling improved communication of complex ideas.

Technical Accomplishments

Tom Sawyer has developed and marketed its first product, a combined graph toolkit and layout toolkit. It took 6 years to develop the first product rather than the 3 years that Tom Sawyer initially thought it would take. As a result, the current generation of the Graph Toolkit only incorporates some of the ATP-funded technology. New product versions released this fall will fully incorporate ATP-funded technology. The new generation of the technology is geared towards specialty markets such as biotechnology and other specific applications within industry. In addition, Tom Sawyer has been able to develop Java and Microsoft (MFC and ActiveX) versions of their technology over the last 3 years to increase the number of market opportunities available for their products.

Products

Product	How Product is Sold
Graph Toolkit	Per seat

Customer Use of the Product

Tom Sawyer's customers can use the Graph Toolkit in two ways. First, other companies can purchase the Tom Sawyer software and incorporate it into the visual displays that are generated from their products and systems. For example, Cisco uses Tom Sawyer products to display how information is moving over a network system. Users of the Cisco product are then able to examine their networks in an easily interpretable manner to get a better understanding of the efficiency of their network. Other companies, such as banks, can use the Tom Sawyer software to better understand their internal workflow and learn what part of their operation is creating bottlenecks or inefficiencies. In either case, customers can reduce cost by using the Tom Sawyer product rather than employing software engineers to generate custom programs for analyzing firm workflow.

Future Products

Tom Sawyer expects to continue using the ATP-funded technology in future products. Their current expectation is to create more specialized versions of their product for specific industries.

B.7.3 Project Performance

In this section, we estimate the economic performance measures for the ATP-funded Tom Sawyer Software project. RTI conducted a structured interview with the principal investigator, Brendan Madden, and gathered information on product life spans, actual and projected sales, and total project costs. From this information, we derived demand curves for all products sold during their expected lives, and used these curves to estimate consumer and producer surplus and R&D expenditures. Based on these estimates, we calculated the net present value, benefit-cost ratio, and internal rate of return for the component software project.

Product Life Spans

The first generation of Tom Sawyer products was released in 1998 and had a life span of three years. The next generation of products is to be released this fall with an expected life span of an additional three years. Because the first-generation product was roughly only half dependent on the ATP technology, we included only half of the estimated benefits of the product in our estimation. However, the second-generation product is totally reliant on the ATP project, so the total benefits from the product are considered in our analysis.

Competing Technologies

Two French companies, Ilog and Loox, are trying to produce similar component-based products to Tom Sawyer's. These companies have not reached the level of technical accomplishment that Tom Sawyer has and are not effectively competing against Tom Sawyer.

R&D Expenditures for the Technology

Tom Sawyer has spent about $180,000 per year on developing the component-based software technologies and commercializing products for sale. They expect this level of spending to continue for another two years at a minimum. ATP contributed $2,000,000 of funds for technology development during the project's 1995–1998 time period. As a result, about $3.4 million in R&D expenditures must be set against the economic benefits generated by sales of the graph development products.

Estimation of Performance Measures

To estimate the consumer and producer surplus benefits generated by Tom Sawyer's Graph Toolkit product, we assumed that the marginal costs of reproducing the software are so close to zero that they are negligible, and the only costs that are important are the investments in R&D.

We first generated a per-year consumer and producer surplus benefits from the technology. R&D expenditures and benefits were adjusted to 2000 dollars to remove effects of inflation. We estimated the benefit-cost ratio, net present value, and internal rate of return and present them in the table below. As these results indicate, this was one of the most successful of the projects funded in the ATP component-based software focused program.

Measures of Performance

Benefit-Cost Ratio	18
NPV of ATP and Private Investment	$52 million
Internal Rate of Return	136%
Producer Surplus	$37 million
Consumer Surplus	$19 million

B.7.4 Qualitative Benefits Attributable to ATP

General Impacts of ATP Funding on the Company

Tom Sawyer said that ATP funding accelerated the R&D that they conducted to develop their graph visualization technology by several years. In addition, ATP funding greatly enhanced the quality of the resulting products. ATP funding enabled Tom Sawyer to focus on R&D and good product performance, rather than on tracking down additional sources of revenue to stay alive (i.e., by moving a less capable product to market quickly). As a result, the product that was released was a much higher-quality product, which is reflected in the price premium that they have been able to charge.

A second benefit of ATP funding was increased credibility. Customers said that receiving the ATP funding was a signal that Tom Sawyer had high-quality technology. The halo effect was a substantial benefit. In addition, several customers identified Tom Sawyer based on the documentation and media attention from the ATP focused program on component-based software. In essence, the ATP award announcement provides marketing value that new, small companies need to sell their products.

General Impacts of ATP Funding on the Market

Tom Sawyer believes that the market for components is driven by getting products to market sooner rather than by out-of-pocket cost savings. By accelerating the R&D process, ATP was able to expand the component-based market.

B.8 XEROX, PALO ALTO RESEARCH CENTER

B.8.1 Company Description

This project was conducted at Xerox's world-renowned Palo Alto Research Center (PARC). PARC was founded in 1970 with a mission to create the "architecture of information," and over the past 30 years has invented graphical user interfaces, Ethernet networking, the computer mouse, object-oriented programming, and many other path-breaking technologies. Xerox is one of the 100 largest corporations in the United States, with 1999 sales of $19.2 billion. The Xerox corporate headquarters is in Stamford, CT,

Company Demographics

Type of Firm	Publicly traded
Founding Date	1906
Company Size	Large
Headquarters Location	Palo Alto Research Center, Palo Alto, CA

B.8.2 Technology Description

Goals of the ATP Project

When Xerox first started its ATP-funded project they realized that the state of technology of component-based software was already advanced. Numerous companies had already produced components that customers could purchase, install, and use within their production process. However, an infrastructure was needed that did a better job of supporting plug-and-play compatibility. It was difficult to determine which components fit together, how they interacted, or whether they would even work together. Components that are unable to fit together or work together have minimal, if any, value; their value occurs when they are tied together within one program and used as a group to achieve a greater purpose. Xerox's goal was to change the nature of component-based software development by incorporating a new model of software development, aspect-oriented programming, into software tools. These tools would separate the semantic details (functionalities) of a component from the implementation details to increase the plug-and-play compatibility of components.

Technical Accomplishments

Xerox developed technology that fully accomplished the goals of the ATP-funded project. Their tool AspectJ embodies this technology. This tool operates within object-oriented programming and Java programming environments. AspectJ allows more components to be plugable into a software program, which greatly reduces the cost and time of installing components and increases the selection of available components that a software developer can use within its production process.

Products

Product	How Product is Sold
AspectJ	Free download from the Internet

Customer Use of the Product

The use of AspectJ is simple and straightforward. After installation, the product blends with the other Java tools, as well as the compiler and other development tools that software developers use. Xerox designed the product to be as simple out of the box as possible. AspectJ can be put to use as soon as it is installed, and software developers who use the product can put together componentized code much more easily and efficiently.

Future Products

AspectJ is currently in beta testing. It will continue to evolve and improve through time. No other products are available based on the same technology, although other researchers at the PARC lab most likely benefited from the ATP award, according to Gregor Kiczales, who manages the project. Furthermore, AspectJ researchers learned about new technological innovations from discussing the ATP-funded project with other researchers at PARC, and may have applied some of those lessons to other products not benefiting from ATP-funded research. However, there is no direct link between the ATP funding and the impact on the non-ATP products. A benefit most likely existed, but it is excluded from this analysis.

B.8.3 Project Performance

This section estimates the economic performance measures for the ATP-funded Xerox PARC project. From structured interview information on product life spans, actual and projected sales data, and total project costs, we derived demand curves for AspectJ during its expected life and estimated consumer and producer surplus and R&D expenditures. Based on these estimates, we estimated the benefit-cost ratio, net present value, and internal rate of return for the ATP-funded project.

Product Life Spans

Xerox first released AspectJ in 1998. Since that time, it has been available for downloading from the Internet and used by thousands of software developers. Xerox will continue to release the product and build on it well into the future; however, to maintain our conservative bias we only estimated benefits through 2002. This is in fact a very conservative approach. Xerox is still beta-testing the product. Once this process is complete, they will release a new product that is expected to have a life span that lasts several years past 2002. These future benefits are excluded from this analysis.

Product	**Life Span**
AspectJ	1998 to 2002

Xerox chose to release the product free over the Internet as part of their pricing strategy. They believe that to turn aspect-oriented programming into a new standard programming methodology, AspectJ must have broad acceptance. Once the methodology has been widely accepted, Xerox could then make a profit from the sale of complementary products, consulting services, and training. This strategy has been pursued by other information technology companies such as Adobe and Netscape.

Competing Technologies

The only competing technology that exists is manually installing components in a program. The costs of doing this are so prohibitively expensive that it is not often done.

R&D Expenditures for the Technology

Since the start of the ATP project, Xerox has spent roughly $500,000 dollars per year on R&D related to the AspectJ technology. Since the end of the ATP project in 1999, Xerox has received additional funding from the Department of Defense that has equaled roughly $500,000 per year in 2000 and 2001. We estimate the total private expenditure to be $5,000,000 over the entire project horizon. ATP contributed $1,670,000 to the project, from 1995 to early 1999.

Estimation of Performance Measures

Because Xerox is planning to offer AspectJ at no cost during the entire span of our analysis, the marginal production costs are very small, and Xerox will earn zero revenues and therefore no producer surplus. However, those firms and individuals who use the product will be earning a consumer surplus, which we estimated from data shared with us by Gregor Kiczales. We estimated the benefit-cost ratio, net present value, and internal rate of return and present them in the table below. The net benefits and NPV from the project are very small as would be expected.

Measures of Performance

Benefit-Cost Ratio	1.2
NPV of ATP and Private Investment	$1.2 million
Internal Rate of Return	13%
Producer Surplus	$0
Consumer Surplus	$6.5 million

B.8.4 Qualitative Benefits Attributable to ATP

General Impacts of ATP Funding on the Company

Without the ATP funding, Xerox would never have been able to develop AspectJ. The idea existed at Xerox and many people agreed it could be a successful product. However, Xerox was unable to muster the resources internally until the ATP funding was put in place. Xerox scaled up their entire production process to match the flow of ATP funds.

There was no external halo effect from the ATP funding. No mention of the funding is included with any of the information that Xerox presents about the product. However, there was a significant internal halo effect within PARC about the project. The ATP funding allowed resources within PARC to be readjusted to allow the development of AspectJ.

General Impacts of ATP Funding on the Market

The Xerox company representative noted that ATP funding of the component-based market has allowed companies to focus on some of the longer-term technologies that are needed to fully develop this market. Companies can be profitable in the short run based on sales of specific components, but the tools and infrastructure that are needed to develop a full-scale market take longer to reach profitability. Because of this, the private sector generally lags in the development of these infrastructure and tools. In the view of Dr. Kiczales, ATP encourages companies to focus on these issues, which has increased the potential for component-based software.