

WHITE PAPER

Companies Achieve Significant ROI Using Fair Isaac Blaze Advisor

Sponsored by: Fair Isaac Corporation

Rikki Kirzner Stephen D. Hendrick
June 2003

SITUATION OVERVIEW: THE NEED FOR BUSINESS RULES

Business rules are those decisions that involve taking a specific action or making a particular decision in real time based on the conditions and business policies and procedures that are in existence at a given time. Any single rule/decision may be used as a means to determine the next action that needs to be taken to drive the business process. Business rules can be anything that is used to run the organization — corporate policies, regulatory authorizations, the logic and procedures found in programming code that comprise corporate or departmental applications, or simply a set of best practices used to solve problems. The ability to create, maintain, update, and follow the business rules governing the operational processes of an organization is critical to a business' success.

Business rules may be as simple as calculating the sales tax for a specific city or putting the terms together for a weeknight promotion at a resort hotel, or they may be as complex as establishing all the possible contingencies in case of catastrophic failure of a real-time, high-transaction financial services application. In any established business, rules and business processes are numerous and cover many situations, calculations, terms, and mitigating factors and conditions. The instability of business rules adds to the complexity of the situation. As companies and their business processes, competitive and regulatory factors, internal policies, and economic conditions change, the rules affected by these changes must be located and updated. Unfortunately, most organizations are unprepared to manage, track, and make frequent changes to the organizational rules.

Trying to define, develop, change, and maintain accurate business rules that are the foundation of how an organization conducts its business can be a costly, time-consuming process, often creating resource bottlenecks throughout an organization. When changes or additions need to be made to existing rules, identifying the person responsible, finding the rules that need to be changed, and delegating the person who needs to make this change are equally difficult and time-consuming tasks. Many factors contribute to these problems, including, but not limited to, the following:

- ☒ The rules that govern how a company runs its business are often written in COBOL or some other archaic language and often reside on legacy hardware or mainframe systems. This means that the business decisions have become embedded as part of the general application framework code.
- ☒ Finding and extracting key rules that need to be updated and changed has been a manual process generally based on using some reverse engineering tactics that don't easily lend themselves to integration with modern, more efficient development environments and newer third- and fourth-generation languages (3GLs and 4GLs).

- ☒ The process of finding and extracting key rules is often difficult because of the spaghetti code nature of how the original applications were written and maintained. Changing the code to reflect new business rules is almost always a difficult, costly, and time-consuming process. Eventually, the application code becomes so entrenched and interdependent that changing it is no longer practical.
- ☒ The process of finding, revising, or updating rules is usually distributed across a variety of departments and professionals ranging from IT staff to database administrators to business architects and programmers.

Companies requiring an effective way to handle the management of those policies and procedures governing the order of execution of business rules in their organization often turn to rules management tools. Business rules, properly implemented, allow legacy systems to continue to support business processes while accounting for changes to business policies, practices, and procedures. Separation of business rules that need to be changed frequently from the remainder of legacy application functionality should revolve around the following procedures:

- ☒ Companies need to identify the rules independent of the underlying structures used to find and update data as well as from the systems used to carry out recommended actions.
- ☒ Companies should be able to choose, manage, and execute the right rules at the right time in the right order without manipulating, affecting, or controlling the application code.

Business rules management tools centralize the definition, discovery, storage, and submission of the vast quantity of rules used in business operations to provide organizations with greater automation, more responsiveness to change, and less expensive distribution and maintenance of their business activities. Because business rules are a part of every business process in every organizational function, it is far more cost effective to manage rules with systems intended specifically for that purpose rather than attempt to manage rules manually. This becomes particularly apparent when the business rules that need to be managed are complex, numerous, interdependent, or frequently changing.

FAIR ISAAC BLAZE ADVISOR DELIVERS REAL VALUE TO DEVELOPERS

In this section, IDC examines a rules management solution that is able to provide real cost savings, time savings, and increased productivity for rules development and modifications through shortened development and deployment cycles. Companies using Fair Isaac Blaze Advisor are deriving significant returns on investment (ROI). Fair Isaac delivers a completely Java-based, rules management software solution designed for companies that need to automate, manage, and personalize business processes in a way that mirrors the unique rules that govern how the company conducts its business. Blaze Advisor enables business users and IT to work together to jointly maintain rules and to reuse rule logic across the enterprise in a unified environment. Blaze Advisor offers integrated, standards-based development, deployment, and management tools that provide an XML multiuser repository; a graphical, integrated development environment (IDE); readable business rules syntax; and a high-performance, scalable, 24 x 7 application server.

Companies using Fair Isaac Blaze Advisor can develop and update business applications more quickly relative to a more traditional 3GL approach to application development. The advantage of faster development allows companies to reduce the total cost of ownership (TCO) for developing, managing, and maintaining applications.

Blaze Advisor increases developer productivity and decreases time to market. It also gives business experts the control and power they need to make smart business decisions and avoid delays in implementing operational rules.

Fair Isaac Blaze Advisor provides unique functionality covering all the criteria for a comprehensive business rules system. Blaze Advisor supports a rigid adherence to the concept that business logic must be independent from the mechanisms used to manipulate data or implement decisions. This separation of business logic from system code allows a given business process the flexibility to use rules to reach operational decisions that can be carried out by a variety of systems and applications. It supports reusability of rules by different applications and business processes as well. For instance, when responding to a large customer requesting product discounts or special pricing considerations, the supplier company needs to make a number of decisions regarding the customer's status, size of the order, purchase history, type of items to be purchased, available inventory, age of items in inventory, seasonal considerations, and so on. These considerations follow a single preestablished decision process, which can be shared or used by completely separate applications, salespeople, or automated phone response systems. There is no need for physical interaction with the customer.

Independent business rules systems offer the flexibility and capability to work identically within multiple implementations. Business rules should not be intricately bound to specific interaction methods or system functions. If they are as in traditional COBOL or other 3GL programs, any attempt to change the decision logic so the application can run on other types of systems or handle similar business processes requires a rewrite of the rules to meet the new requirements.

Blaze Advisor does not include facilities to physically control input and output on a browser or to any specific device. It contains clearly defined integration points for systems that are built to accomplish these mechanical tasks. So a rule can recommend an actionable concept, such as "offer platinum class member customers an automatic 15% discount," which can be used by various systems to make the promotion on the Web, through mail order catalogs, at retail outlet terminals, or other channels without needing to duplicate and alter the rules.

Business rules must be written in a formalized, unambiguous syntax or language capable of expressing the business logic. Blaze Advisor uses a proprietary rule language that combines readability and flexibility that makes the intent and logic of a rule clear to a business user. Business policymakers can participate directly in the development, testing, and modification of the business rules by using Blaze Advisor's rule language. This rule language also helps to reduce the time it takes to implement the business rules and decreases the potential for interpretation errors between business intent and programming implementation. Blaze Advisor includes support for calls to other applications, as well as a rich procedural syntax that handles looping, mathematical operations, and operations on text strings. These functions are executed within the framework of the Blaze Advisor rules processing environment that maintains the separation of rules from the remainder of the system application code.

Blaze Advisor contains automated tools to read these existing data definitions and make them available for rule writing. At the same time, links to the external data sources for those items are established so that when rules refer to them, the values can be read from or updated in the external storage locations. This approach eliminates the need for a dictionary definition step before writing rules. In addition, items can be given more understandable names for use within Blaze Advisor rules without needing to change their existing data definitions. As a result, Blaze Advisor lets developers extend the usefulness of preexisting data sources and legacy system code while allowing more natural business logic definition within the rules system.

Blaze Advisor allows the rule developer to create templates that can be used for faster creation of rules, increased security, and improved consistency in rule definitions. Blaze Advisor is built to be a callable component of a business application, providing decisions based on data about the case in question. That data may be read from database tables, XML documents, MQ Series data streams, Java or COM objects, or passed parameters from the calling application. Blaze Advisor supports decision tables that can be imported from Excel spreadsheets. Values can be changed within the table, and tables can be formatted to allow multiple rows or columns, or single-axis lists. Results in the lookup cells can be complex rule actions as well as simple return values. Blaze Advisor also supports decision trees for increased clarity and decreased development time and effort when specifying business logic.

Using a complete business rules system reduces the complex programming work needed to create and support the execution of business logic that comprises a business application. Furthermore, business rules technology incorporates sophisticated software components designed to accomplish all the tasks that would otherwise need to be written in the application code. Blaze Advisor contains ready-to-use components to manage simultaneous requests for decisions from different applications or for different users; condition-based selection of the right rules to fire, in the right order; auditing software to record what rules are used in the course of making a decision; and support for complex decision chains without needing explicit control from the calling application.

It is important to realize that business decisions must be made even when there is no known or available information. There is often a need to interact with the user of the business application to get data that is used in the decision process. Blaze Advisor includes functionality to make this process efficient and powerful. One unique feature of rules development in Blaze Advisor is the ability of the tool to distinguish between data values that have not yet been asked for, data values that the system attempted to retrieve and found no value for, and data values explicitly set to null (no value).

Blaze Advisor can specify a group of questions that should be asked as a set at the same time. Each question can have an internal property or data item associated with it so that any answer that is provided is placed directly into its correct storage location. A rule can then call for a single action to retrieve the full set of data values, thus reducing coding time and complexity during development as well as simplifying the process of updates and changes should the business analyst or developer want to add more questions later.

Finally, Blaze Advisor contains features to simplify the dynamic updating of information on Web pages for interactive applications. Rules can specify what information should be included on a Web page, including additional questions that should be asked based on the answers to previous questions. Throughout this process by which the Web pages are displayed, the Web pages are handled independently from the business logic so that separation of decision logic from infrastructure code is maintained. Easily defined and maintained links to the external systems provide speed and efficiency during development and execution.

Blaze Advisor operates as a pure Java application and provides full support of the write once run anywhere functionality. Companies do not have to worry about creating different versions of the software to run on different devices or purchasing separate licenses or versions of the software to run on different operating systems or types of hardware. Blaze Advisor also runs on any of the major Java application servers without modification. A rule service (decision process) may be called as a logic component from any enterprise application, whether written in Java, COBOL, or C++. The rule execution stays consistent no matter what system calls for it or what machine runs it. This capability allows the decisions at the heart of legacy systems to

be renovated for clarity, processing efficiency, and ease of maintenance without having to replace the entire system. Once rule services are defined, they can be reused by additional applications and physical systems throughout the enterprise without modification.

There are many technical implementations used to manage large-scale business systems. Systems architects face decisions about issues such as application server integration, stateful versus stateless sessions, synchronous versus asynchronous operation, and Enterprise Java Bean (EJB) and Microsoft Access Database (MDB) container specifications. Blaze Advisor automatically generates the proper integration code to work under any of these scenarios so that implementation time and costs are reduced.

Blaze Advisor uses configurable parallel processing threads so that enterprises can balance machine resource utilization against processing volume demands. These threads can interact with load monitoring and balancing services provided by application servers to dynamically adjust system performance for optimum performance/capacity tuning. Blaze Advisor can not only scale but also handle increased processing needs in a well-defined manner, making use of additional computing resources as they are added. Blaze Advisor can help companies handle and manage large interactive transaction loads servicing nationwide Web sites as well as processing-intensive batch operations in a finite processing window. Blaze Advisor can also efficiently handle increases in the number of rules as they are added over time. To date, there have been no limits found on the number of rules that can be managed and used in a Blaze Advisor rule service.

The unique ruleset and ruleflow architecture of Blaze Advisor enables the processing engine to examine a subset of the entire rule population to find the rules applying to a particular task in an overall rule service. Then the advanced rule filtering technology in the Blaze Advisor rule execution engine efficiently categorizes rules based on similar conditions to add the right rules to the processing queue, bypassing rules that do not affect the decision outcome. Rules can be predefined and stored in the runtime system with effective date stamps that indicate when the rules should be considered by the rules engine and when they should be bypassed. This ability reduces the need for code changes to the application and redeployment of the business logic.

Newer features in Blaze Advisor include the ability to construct chains of rules in a graphical tree format, which lets designers focus on a particular branch of logic leading to a decision. The software has also been integrated with modeling software from Fair Isaac, allowing companies to use mathematical techniques to segment data populations for best action strategies and import those strategies into Blaze Advisor as business rules. Advanced versioning controls allow companies to track changes to rules over time, checking out rules to prevent editing conflicts, comparing different rule versions, and reinstating past versions of rules if it becomes necessary to roll back to previous versions

Business rules constitute the part of an application most prone to change over time. New or altered regulations must be adhered to, new business methods are implemented, competitive pressures force changes in policy, new products and services are introduced, and so on. Changing business situations require modifications to the decisions that control behaviors and actions in business applications. The people best able to gauge the need for new operational behaviors, envision the new decision criteria and responses, and authorize implementation of new business policies are seldom technically trained in programming techniques. Traditional organizational behavior is for business policymakers to gather a set of business changes that should be made, submit them as a formal request to a programming department, sign off on the programming interpretation of the changes, and wait for a scheduling opportunity to have the changes implemented in a new

software release. The delays, costs, and lost business opportunities in this type of cycle are apparent, but there is no alternative in traditionally implemented software systems.

Blaze Advisor offers organizations a way to implement such business changes more efficiently and more quickly, with less chance of errors, less cost, and less operational impact. Because business rules are separated from and independent of the underlying system code that keeps a business application operating, they can be changed without impacting basic system functionality.

Blaze Advisor's operational reliability and performance have led it to be included as the rules technology of choice in major enterprise software products in the three companies profiled in this document.

ROI CASE STUDY METHODOLOGY

IDC employed a structured interviewing methodology to collect ROI qualitative and quantitative data for Fair Isaac customers using Blaze Advisor. Three case studies are presented. The same methodology was employed for all case studies. The information in these case studies represents the experiences of end-user organizations with Blaze Advisor but in no way amounts to an endorsement of Blaze Advisor. Please note that the company producing Blaze Advisor underwent several acquisitions and name changes during the time period spanning some of the case studies in this report. Therefore, the name of the company offering Blaze Advisor may be different from Fair Isaac in some of the following case studies.

ROI COMPONENTS DEFINED

Net quantifiable benefit (NQB) is the benefit to the organization from the implementation of Blaze Advisor. The term *implementation* can include the relative benefits in the development, deployment, operation, and maintenance of Blaze Advisor applications. NQB components can be either positive or negative. NQB can be either direct, in terms of incremental revenue generated, productivity gained, or expense saved, or indirect, from the redeployment of resources to tasks for which the organization would alternatively have had to hire new and like resources to perform. All net quantifiable benefits are expressed in U.S. dollars rounded to the nearest \$1,000.

Net quantifiable cost (NQC) is the difference between the total cost and expenses associated with Blaze Advisor and the similar costs associated with the most likely alternative solution (which can be none). Typical NQC components include software licensing and maintenance costs, product consulting costs, training and education costs, and any added hardware costs associated with implementing Blaze Advisor. All net quantifiable costs are expressed in U.S. dollars rounded to the nearest \$1,000.

ROI CALCULATION

ROI is the ratio of total net quantifiable benefit divided by total net quantifiable cost. This ratio is equal to 1 (or 100%) when net quantifiable benefits equal net quantifiable costs, which represents a break-even condition. A ratio less than 1 reflects a negative improvement and a ratio greater than 1 reflects a positive improvement.

ROI is frequently expressed as a percentage, in which case the ROI result is multiplied by 100. The ROI percentage calculation can thus be represented as:

$$\text{ROI} = (\text{NQB}/\text{NQC}) \times 100\%$$

Other ROI terms and information that may be provided in a case study include:

- The *ROI factor* is the ratio of NQB to NQC (NQB/NQC) expressed as a real number instead of a percentage.
- The *ROI value* is the difference between NQB and NQC (NQB–NQC).

Each case study references the annual net quantifiable benefits and net quantifiable costs as well as the annual and cumulative ROIs over time. IDC finds that in the early stages of some Blaze Advisor implementations there is development cost that may not have any associated quantitative benefit. This situation may produce an initial undefined ROI even though there is a negative net ROI cash balance.

OTHER ROI CONSIDERATIONS

ROI is also a relative measure in that the benefits and costs of Blaze Advisor are generally compared to those of an alternative solution — typically, what the organization had been doing to address this business problem before the implementation of Blaze Advisor. It follows that NQB and NQC will be defined differently for each interviewee situation, and, in some cases, the results will not be as quantifiable as in others.

Increased productivity of a workforce is often a result of implementing new software technology. Where productivity gains are achieved, IDC assumes that the resulting "found time" is put to use in some other productive use within the organization or leads to a reduction in force.

ROI DATA COLLECTION METHODOLOGY

To collect the data, IDC conducted telephone interviews between June 2002 and April 2003. The purpose of the interviews was to understand the business problems the organizations were seeking to solve, the role of Blaze Advisor in solving these business problems, and the ROI associated with the implementation of Blaze Advisor.

Not all benefits or costs are easily quantifiable. However, the approach to ROI taken in this paper seeks to identify and quantify all significant net benefits and costs. If during the course of the interview we encountered significant net benefits or costs that the interviewee did not feel comfortable quantifying, our policy was to terminate the interview and retire the case study.

IDC established a series of predefined questions for use in the ROI interviews. These questions were designed to enable the collection of complete ROI data as well as to provide a consistent framework for analyzing the ROI data. These questions enabled IDC to understand the organizations' general approach to development and deployment as well as evaluate the relative attributes, benefits, and costs associated with Blaze Advisor.

IDC analysts performed all data collection and ROI calculations associated with these case studies, independent of any involvement by Fair Isaac.

ROI TIME FRAMES

The time frame for assessing ROI varied from case to case. The shortest ROI time frame was three years and the longest was eight years. IDC prefers ROI time frames that span three or more years.

The number of historical and forecast years in the time frame and the overall length of the time frame were left to the discretion of each company. While IDC believes that longer time frames result in an ROI with increased accuracy, it is also clear that the proportion of historical data to forecast data also influences accuracy. Therefore, the parameters for ROI analysis (start year, end year, and any special analytical techniques) will be presented in each case.

The time frame for every case study included two or more years of historical data and/or one or more year(s) of projected data. Projected data is based on the trends suggested by the historical data after adjustments for management expectations regarding how benefit and cost components would change over time.

To simplify the analysis, IDC based all projected quantitative data on 2002 constant dollars.

FAIR ISAAC BLAZE ADVISOR ROI CASE STUDIES

Three distinct and unrelated companies using Fair Isaac Blaze Advisor were interviewed for this study. The resulting case studies are presented below.

CASE STUDY 1: A GLOBAL SERVICES COMPANY

COMPANY OVERVIEW AND BUSINESS ENVIRONMENT

This ROI analysis was performed on one particularly large division of a multibillion dollar global services and systems integration company headquartered in North America providing customized solutions for all industries (herein referred to as "global services company"). The specific department participating in the ROI analysis is a member of a "solutions consulting practice" that has more than 50,000 employees who provide industry-specific, technology-based infrastructure solutions primarily targeted at the healthcare and insurance industries. The solutions provided for the customers of this global services company cover the entire life cycle ranging from business process management to consulting and/or sourcing. The specific department participating in this case study tailors every solution to meet an individual customer's specific needs. The global services company may leverage previous work done for other customers or work done by other divisions by reusing some key core components and infrastructure, but each solution is specifically created to solve the individual customer's particular business plans and requirements.

CURRENT IT ENVIRONMENT

The global services company has two business goals — crafting customized solutions to meet customers' specific needs and entering into long-term retainer-based engagements focused on providing additional value on a day-to-day basis. Because the technology provided is custom tailored to the client, it spans all software and hardware platforms and architectures that may be deployed in or desired by the customer corporation. The global services company never assumes there will be a specific hardware or software technology in place before entering into an engagement. Therefore, the global services company is prepared to support all the

different types of hardware and software platforms that its customers might have at their own organizations. Two requirements drove the global services company to look for tools that could help it deliver solutions to its customers in a rapid, more cost-efficient manner: the need to rapidly build the original solution and the need to quickly change the solution once it is built.

COMPELLING REASONS FOR SELECTING BLAZE ADVISOR

The global services company first came across the rules engine product later known as Blaze Advisor when it was being produced and sold by Neuron Data. The global services company was searching for rules management capability in the late 1990s and began using Neuron Data's products in 1999. Neuron Data's tools were chosen because the global services company had the following set of requirements: to deliver a Java application that was flexible and easy to manage; that could be written using business rules; and that could be used to implement and manage business processes. Neuron Data met these requirements at the time. The global services company evaluated Neuron Data products, conducted a test pilot, and felt the product solved the business requirements for managing and writing business rules. Because business rules are the fundamental driver of most projects undertaken by this global services company, it needed tools that would let its developers write business rules quicker and manage them easier than by having to write everything in COBOL, for instance.

Despite Neuron Data's transition to Blaze Software and its eventual acquisition by Fair Isaac, the business rules company maintained an ongoing, trusted relationship with the professional IT staff of the global services company. More important for the global services company, the same people that it relied on for technical support and product advice at Neuron Data are still with Fair Isaac today. This relationship is an added bonus of maintaining the business relationship with Fair Isaac because the global services company knows that these employees will continue to provide the support and advice that they require.

With more than two years of experience using Blaze Advisor, the global services company is pleased with the tool's ability to solve their requirements. It evaluated some other offerings but found that no other company in this space provides the same type of capability.

PROJECT DETAILS

The initial evaluation of the Blaze Advisor project for this particular case study began in 1999. The system creation started in 2000 and went into production at the beginning of 2002. The global services company took this time to bring the system under construction to its present state, where it can now add, delete, and modify the rules and order of rules that currently exist in the system. It has also added automated conversions.

The alternative to Blaze Advisor would have been coding all the required business rules in COBOL and creating its own rules engine. The global services company noted that because of the amount of work involved, if it hadn't liked Blaze Advisor or hadn't been able to use Blaze Advisor the way it needed to, then it would have canceled the project with this particular customer. The time to create the rules engine would have required in excess of one or more man-years of effort. At the beginning of the project, the global services company had to recreate the equivalent of more than 100,000 rules that were running on the mainframe. Once the IT staff figured out the efficiency and redundancy factors, it still needed to code or create more than 70,000 business rules.

During the time the global services company was creating the rules, the customer had to remain competitive and respond quickly to changes in its own market and in government regulations. Therefore, complex rules had to be created quickly (in days rather than weeks or months) so the customer could continue to conduct business while also responding quickly to changing market and regulatory demands. The customer needed the global services company to help them remain competitive or it would have abandoned the contract. The global services company found a way to respond to the customer's needs by using Blaze Advisor. This solution satisfied both parties.

Blaze Advisor allowed the global services company to create a baseline application framework that enables it to plug in the unique rules of each new client's specifications, making future implementations that much more efficient and quicker to develop and deploy. Had the project originally been coded using COBOL, then the global services company would have had to code the application in a top-down process with multiple conditions. Using Blaze Advisor, developers can code a single rule by describing what needs to happen when a rule fires. The Blaze approach requires a different way of thinking that is far more efficient and organized. This approach produces code that is also easier to debug. It frees the developers from constriction of the COBOL language. It also frees up the server because Blaze Advisor manages the code required during runtime execution of the rules.

ROI ANALYSIS

The benefits and costs of using Blaze Advisor are amortized across a three-year period from 2000 to 2002, eliminating the typical fluctuations that can occur in any given year.

NET QUANTIFIABLE BENEFITS

The global services company recognized ROI benefits in several areas as a result of using Fair Isaac Blaze Advisor: Real benefits were derived from the efficiency of building business rules and the resulting cost savings in terms of increased developer productivity, quicker time to market, and the ability to take rules that were created for the first project and reuse them in future projects for this particular customer and for new customers that require implementation and management of business rules.

The global services company expected that the portion of the project involved in the creation of more than 70,000 required rules would consume the majority of manpower and time. The cost for business rule development alone using Blaze Advisor was approximately \$2.7 million.

Due to the efficiency of Blaze Advisor, building the business rules actually took the least amount of man-effort and time to complete. The global services company originally assigned 15 people to the project for the sole purpose of coding the rules. The company anticipated that more people would be added to the project as it progressed. Instead, the global services company quickly realized that the extra resources wouldn't be required because of the efficiency of using Blaze Advisor and, in fact, reduced the number of developers as the project progressed. On average, coding rules with Blaze Advisor requires between five and eight people. Since the original estimates were that it would take 15 to 24 people, the company saved the costs associated with hiring the additional number of developers originally proposed for the project.

Having to build business rules is just the beginning of the problem for most companies. In many businesses, the highest costs, the most difficult aspects of development, and the least automated business processes are those that rely on business rules or decisions that require an appropriate action to be taken. Business rules governing operational processes may be complex, with many interrelated considerations. Business rules may

also be simple, but there may be so many rules spanning a wide variety of different situations that it becomes extremely difficult to create the order and circumstances under which the rules are to be executed. For developers it is particularly difficult to try and code this type of logic using a 3GL, such as COBOL, because the logic has to be written in a sequential top-down order to accommodate the limitations of the coding process. All the possible combinations and circumstances have to be thought out in advance and laid out in a series of conditional statements or the equivalent. To make things even more complicated, rules may change frequently based on modifications to internal or external policies and external competitive or regulatory factors. A rules management engine offers an automated way to meet the complex requirements of organizations trying to cope with the use of business rules in their operations. Blaze Advisor comes with its own rules engine; therefore, the global services company did not have to build this logic and capability from scratch. The availability of being able to deploy the Fair Isaac Blaze Advisor Business Rule Engine rather than having to build it resulted in a savings of nearly \$135,000 in actual development costs. However, IDC did not attempt to quantify the actual additional savings derived from not having to debug, test, and maintain this coding effort.

ADDITIONAL BENEFITS OF USING BLAZE ADVISOR

By using Blaze Advisor, the global services company has achieved the following additional benefits that were not quantified by this ROI analysis but that will result in cost savings for this and future projects. The global services company is now able to:

- Create and manage new business rules
- Change existing business rules quickly
- Quickly reorder its business rules to better comply with the company's business processes
- Establish a future framework that can be leveraged and partially reused for other projects requiring rules creation
- Allow developers to see the flow of the rules and determine where changes need to be made

NET QUANTIFIABLE COSTS

The project consisted of license, maintenance, and training costs. The license costs have both server and client components. Not all the costs for licenses were captured in this ROI analysis, because the first licenses were purchased in 1999, and this study starts capturing ROI data beginning in 2000. The Blaze Advisor license costs are actually composed of both a developer license and a server license. There is also a maintenance cost for both the server and the developer licenses.

The total cost of the Blaze Advisor licenses was more than \$504,000, but this figure was amortized over three years. All of these costs were incremental and associated with bringing Blaze Advisor into the global services company.

Total maintenance costs for both the developer and server licenses were nearly \$142,000.

Training costs were incurred in two areas — one week of formal training by Fair Isaac for 15 people and internal training costs. Total training costs amounted to approximately \$45,000.

Total costs for this global services company project were slightly more than \$691,000 amortized over three years.

RETURN ON INVESTMENT

Due to initial learning curves and early license costs, the ROI value for this project (NQB–NQC) started as 448%. This increase in ROI value was attributable to the fact that not all the costs for licenses were captured in the initial year's ROI analysis, since the first licenses were purchased in 1999 and this ROI study starts capturing ROI data beginning in 2000. As the initial benefits associated with easier rule processing were instituted and the next license costs were calculated, the ROI value dropped to 369%. However, as the ongoing benefits of using business rules became apparent in the reduction of maintenance costs, the ROI value increased to 413% in the third year of this project. The overall ROI for this project is expected to be 409%, with a total ROI value of slightly more than \$2.1 million, which equates to an ROI factor of 4.13. Thus, for every dollar invested in rules creation and management for this project, a net benefit of \$4.09 was realized, resulting in an attractive benefit to the company for switching to business rules.

CASE STUDY 2: CALIFORNIA DEPARTMENT OF MOTOR VEHICLES

COMPANY OVERVIEW AND BUSINESS ENVIRONMENT

This ROI analysis was performed for the California Department of Motor Vehicles (DMV) vehicle registration computation project. California's DMV vehicle registration division is responsible for licensing, tracking, and collecting fees for more than 30 million drivers, more than 20 million vehicles, and more than 5 million vessels in the state of California. There are more than 3 million online transactions per day regarding information on drivers, vehicles, and vessels. The balance of requests for this information comes from law enforcement agencies. In addition to its role as the licensing agency for all vehicles in the state, the DMV serves as a revenue collection agency for city, county, and statewide offices, including responsibility for collecting parking tickets, vehicle licensing fees, and driver license fees.

The main reason the DMV decided to use a rules-based approach was that rules implemented for the DMV to collect various revenues are continuously being changed by the state legislature and must be implemented within a limited amount of time. Making changes to the rules governing fee collection is time consuming, complicated, and must be extensive across the entire state. In the rush to implement the new changes, quality often suffered and changes were not always implemented uniformly across the system. The DMV was looking for a solution that would help it improve quality, increase the speed of implementation, and allow the business analysts on staff to make their own changes rather than wait for the changes to be made by the DMV's IT department.

CURRENT IT ENVIRONMENT

The DMV houses a collection of all types of legacy technology systems, with some dating as far back as the early 1960s, from online system monitors to applications written in assembler to pre-SQL databases such as VSAM. DMV information and the systems hosting this information are available 24 x 7. Monthly notifications for renewal, for instance, are done via COBOL applications running in batch mode. IBM 390 mainframes run many of the applications that are key to DMV operations. Online distributed applications run on RS6000, which uses an obsolete legacy language in 170 field offices to help drivers renew licenses and process their license fees. The DMV's requirement was to have a single set of rules that was well integrated with Java and new technologies as well as with the old legacy language environments.

Any changes to this environment are not only at high risk for error but are extraordinarily complicated to implement. The DMV is using Blaze Advisor to take all the rules that deal with computing fees currently written in old legacy language and COBOL and make them available to both the batch systems and the computers located in all the field offices as well as newer Web-based applications.

COMPELLING REASONS FOR SELECTING BLAZE ADVISOR

The DMV first came across Neuron Data (later known as Blaze Software and now a part of Fair Isaac) when searching for a business rules engine to use for business rules management capability in the late 1990s, and it began using Neuron Data's products in 1999. The DMV chose Neuron Data's tools because it needed to deliver a solution that was better than previous versions, faster to market, highly technical, and of higher quality. The DMV hoped the work involved in deploying a rules engine would only have to be done once and would run on all the diverse systems in the DMV environment.

Neuron Data met the DMV requirements and was the only product that seemed capable of handling the complexities of the technologies making up the DMV computer environment. Blaze Advisor encoded some of the DMV rules and performed a proof of concept, which proved Blaze Advisor could handle the project. The DMV felt the product solved the business requirements for managing and writing business rules and, in fact, could effectively handle these business rules.

PROJECT DETAILS

The project began in 1999 when the first consultant was hired by the DMV to try and produce a workable solution to allow all computers to access the same set of business rules. The consultant spent almost a full year reverse-engineering the existing rules within the DMV COBOL applications. In the end the consultant had not produced anything that could be easily used by DMV. In April 2000, the DMV replaced the consultant with a new project manager who did enough research to realize that only Blaze Advisor was cross platform and only Blaze Advisor could run across the RS6000 and IBM 390 environments. The tool also had to have an interface useable enough for the business analysts to be able to go in and modify the business rules once they were developed. At the onset of the project only Blaze Advisor met these criteria. The proof of concept using Blaze Advisor occurred in January 2001, and the project began in earnest in February 2001.

The development team consisted of five full-time and two part-time employees handling the technical portion of the development, one business manager, four business analysts, and one consultant with rules engine expertise.

The project has not yet been fully deployed. Currently, all the business rules have been identified and documented so that they can be converted into Blaze Advisor as soon as the project commences. In July 2001, the development team had to institute a major design change that had nothing to do with the Blaze Advisor product. The change resulted in a considerable delay. Additionally, the state budget was stalled by the California legislature's decision to cut off funding to the project for almost three months.

ROI ANALYSIS

The benefits and costs of using Blaze Advisor are forecast and amortized across a period from 2000 to 2007, eliminating the typical fluctuations that can occur in any given year. Development of the system commenced in 2000, the system is scheduled to go live in 2003, and the DMV suggested that we base our ROI calculation on five years of production experience (2003–2007).

NET QUANTIFIABLE BENEFITS

The DMV recognized ROI benefits in several areas as a result of using Fair Isaac Blaze Advisor. Real benefits were derived from the efficiency of building a single set of business rules that can run on all its hardware and software platforms. Cost savings occur in terms of increased business analyst productivity; that is, business analysts can modify the rules themselves rather than submit them into a busy IT organization work queue and wait for the IT organization to make the changes. This improvement in workflow will also result in less time required to implement necessary rule changes and improved quality control because rules are changed by the business analysts who are most knowledgeable about the required changes. The changes then can be deployed on all the diverse systems within the DMV offices, thereby eliminating the inconsistencies inherent when trying to create multiple versions of the changes to run on a variety of different systems.

The DMV expected that the amount of time required to make and implement rule changes for each of the major types of systems deployed throughout the DMV utilized the majority of the organization's manpower and time. By using Blaze Advisor to create changes to the rules, the DMV reduced the time to 20 hours for complex rules and fewer than 10 hours for simple rules from the original average of more than 100 hours per rule change. Due to the efficiency of Blaze Advisor, the ability to make frequent changes to business rules significantly reduced the amount of man-effort required to complete each rule change. Seven full-time people are assigned to the project coding and deploying the rules. The DMV anticipates that it will not have to add people to the project as it progresses. In the first year of use, Blaze Advisor is expected to save the DMV more than 13,000 hours to make changes to the rules. These savings are expected to increase over time due to a projected annual increase in rule changes.

Business rules governing revenue and fees may be simple or complex, with many interrelated considerations and decisions that govern the amount of money licensed drivers have to pay. Because of the wide variety of different situations that may be triggered for each fee, it is extremely difficult to create the order and circumstances under which the rules are executed. For developers it is particularly difficult to try and code this type of logic using a 3GL because all the possible combinations and circumstances have to be thought out in advance and laid out in a series of conditional statements or the equivalent. Rules governing fees in California change frequently based on modifications to laws and other regulatory factors. Blaze Advisor's rules engine offers programmers an automated way to meet the complex requirements of the DMV without forcing them to code either in a precise order or according to the convoluted logic governing all the conditions that fire the rule.

The availability of being able to deploy the Fair Isaac Blaze Advisor business rule engine rather than having to build complex business rules using other coding methods produced a savings of nearly \$439,000 in actual development costs. Being able to deploy the Fair Isaac Blaze Advisor business rule engine rather than having to build simple business rules using other coding methods will produce a savings of nearly \$2.96 million in actual development costs over the same time period.

ADDITIONAL BENEFITS OF USING BLAZE ADVISOR

Having to build business rules is just part of the problem for most companies — and particularly for the DMV. The highest costs for development as well as one of the more difficult aspects of deployment involved the developers' difficulties with handling the antiquity of the DMV's legacy systems and outdated software: IBM no longer supports old legacy language; the mix of languages ranges from COBOL to Assembler; SQL data must interface with VSAM data; RS6000s and OS390s must integrate with newer WebSphere servers. Attempts to propagate required changes to rules to another system have suffered from a lack of quality and have been error prone, resulting in costly delays to rule deployment. It has been both expensive and time consuming to find and fix the bugs in this incredibly complex environment. Blaze Advisor's ability to avoid problems stemming from forced integration between obsolete systems is proving to be a real benefit to the DMV. Real cost benefits are derived from making changes to the rules in one place and using Blaze Advisor to take all the rules that deal with computing fees currently written in old legacy language and COBOL and make them available to both the batch systems and the computers located in all the field offices as well as newer Web-based applications.

By using Blaze Advisor, the DMV has achieved the following additional benefits that were not quantified by this ROI analysis but will result in cost savings for this and future projects. The DMV is now able to:

- Create and change business rules quickly using its own business analysts
- Respond more quickly to state, county, and city legislative changes
- Improve quality and significantly reduce the risk associated with rules changes
- Establish a future framework that can be leveraged and partially reused for other projects requiring rules creation
- Allow developers and analysts to see the flow of the rules and determine where changes need to be made

NET QUANTIFIABLE COSTS

Costs for the project consisted of license, maintenance, training, consulting, and education costs. There is also a maintenance cost for both the server and the developer licenses.

Blaze Advisor licenses were purchased in 2001 and 2002. The amortization schedule for each license purchase is three years. Maintenance and upgrade fees are assessed annually. Total license and maintenance costs over the seven-year time horizon for this ROI study were \$774,000.

The cost of using outside professionals to help with system development was \$265,000.

Training and education costs were also incurred during the development phase of the project and amounted to \$70,000.

Total training, education, and consulting costs amounted to approximately \$335,000, but this figure was amortized over three years.

Total costs for this DMV project amount to \$2 million amortized over six years.

RETURN ON INVESTMENT

The ROI value was negative in the first three years of this project largely due to the impact of training and start-up costs that had minimal benefit on ROI value. During the three years of actual use and anticipated production from 2003 to 2005, IDC anticipates that a significantly increasing ROI value will enable the DMV to break even by the end of 2005. This estimate is attributable to the declining cost structure associated with a stabilized development environment and the significant benefits that commence with deployment. As the cost structure plateaus in 2005 for the license and maintenance costs, the annual ROIs anticipated during 2005–2007 are estimated to be in the 900–1,100% range. By the end of five years of rule production use, the overall ROI is expected to be 170%.

This ROI is being achieved with what appears to be a modest number of business rule changes. Had this solution been deployed more quickly, in a more dynamic environment, DMV would have recovered its costs sooner and achieved a much higher ROI in less time.

The overall ROI for this project is estimated to be 170%, which equates to an ROI factor of 1.7. Thus, for every dollar invested in rules creation and management, a net benefit of \$1.70 was realized.

CASE STUDY 3: GLOBAL FINANCIAL SERVICES COMPANY

COMPANY OVERVIEW AND BUSINESS ENVIRONMENT

This ROI analysis was performed for a department within a large global financial services company for its new online dispute resolution application project. The company provides a variety of financial services for its end-user and business customers. It also offers many Web-based services, including allowing customers to check their account balances, reconcile financial statements, conduct brokerage-related activities, and perform other financial transactions. The department is responsible for creating and enabling most of the Internet-based, customer-facing activities for the company.

One major production application already existed to handle customer dispute issues. The main reason the financial services company decided to use a rules-based approach was that it wanted to provide a second service channel that would allow it to both automate the call-handling capability and to improve the efficiency and maintenance of the existing application. That application had been developed internally years earlier and entailed a rules engine that handled a large number of business rules to determine the precise steps the customer call center representatives should follow when trying to resolve customer disputes. In 2001, the company decided to update and improve the application and to provide this capability as an online application that customers could access if they were unable to call the call center during its hours of operation and/or were unable or unwilling to wait for a live person with whom they could speak.

The company was looking for a tool that could leverage the numerous rules already in existence in the older application. It also wanted a tool that would help improve quality, increase the accuracy of implementing the requirements, and ultimately allow the business analysts on staff to make their own changes to any rules that needed modification.

CURRENT IT ENVIRONMENT

The financial services company operates a variety of centralized hardware and software systems. The preferred environment consists of WebSphere servers and Java running on top of Solaris. The company also has a variety of legacy systems that must be accessed frequently, including DB2 and IBM mainframes, because many of the customer records, demographics, and important information reside on legacy systems. The account receivables and other legacy applications reside on mainframes. Additional key applications — many of which are global in nature — that were either developed internally or purchased and customized many years ago are still crucial to the operation of the business.

Because no changes to this environment are being considered at this time, the tool that was chosen had to accommodate the diversity and variety of the existing systems.

COMPELLING REASONS FOR SELECTING BLAZE ADVISOR

The financial services company first came across Blaze Advisor when it issued a large request for proposal (RFP) in search of vendors with tools that could help the company update and migrate its existing dispute resolution application to the Web with minimal disruption and effort. It wanted a tool that would handle the numerous rules that were already in existence without having to significantly rewrite most of the application.

Several vendors' tools were evaluated. Blaze Advisor was selected because the financial services company had a requirement to deliver a solution that was better suited as a customer-facing application. The company also wanted a rules engine and tools that would be more efficient and easier to maintain than the internally developed rules engine. In addition, the financial services company needed a tool that could be used by other people within the organization so that the application and its future maintenance could be transferred to another department.

Fair Isaac met the requirements, and the developers preferred the environment and rules organizational capabilities of Blaze Advisor to the other products being evaluated. Blaze Advisor seemed capable of handling the complexities of the technologies comprising the financial services requirements and operating successfully within the diverse computer environment.

PROJECT DETAILS

The project began in September 2001 when the company issued an RFP. Because the rules were already in place and the company wanted to move to a better solution for handling the rules, there was an urgency to complete the project as soon as possible. Because there was no need to rewrite most of what it had created, the new application was deployed in September 2002. While the internal solution was capable of handling the company's requirements for dispute resolution, it was a fairly basic application. As the strategy turned more toward the Internet, it became clear that the company needed something that was more sophisticated for a customer-facing application than the application that was currently being used by the call center. The internally developed rules engine had been in use for several years. It was complicated to use, proprietary, and did not have the same functionality as Blaze Advisor. The processes to access and add rules were proprietary as well, and so it was difficult to integrate the internal application with other tools or technologies. The company wanted to create a new application to help customer service representatives avoid any disputes that might arise. It wanted to extend its existing application to run on the Internet, and the company already had a clear idea of what methodology it wanted to use and the infrastructure on which the application would run. More important, it wanted to leverage the work that had already been done to create the rules that were in place as well as extend the

application to handle additional rules. As part of the process of converting to a new rules engine, it didn't want to rewrite everything; rather, it just wanted to be able to "tweak" the rules to create a new application.

The types of transactions required for resolving customer disputes are essentially the same whether the dispute resolution is handled via the call center or over the Web. The advantage of having the application available to the customer on the Web is a matter of convenience for the customer. The customer does not have to stay on hold if the call center is busy, and Web access may be more convenient for some customers.

The original development team that created the proprietary application consisted of 15 full-time technical developers, 20–22 extended developers who supported the infrastructure and the development environment and systems, and 13 business analysts. The new application required only two technical developers and three rules authors.

ROI ANALYSIS

The benefits and costs of using Blaze Advisor for this project are forecast and amortized across a four-year period between 2001 and 2005, eliminating the typical fluctuations that can occur in any given year. Development of the system commenced in late 2001; the system went live in mid-2002.

NET QUANTIFIABLE BENEFITS

The financial services company recognized ROI benefits in several areas as a result of using Blaze Advisor. Real benefits were derived from the efficiency of building business rules that can more easily interact with all its hardware and software platforms. The original application grew up over a number of years; therefore, making business process changes and technical changes to the application and interfacing those changes across a variety of other different technical applications were huge challenges for the financial services company. One of the benefits of Blaze Advisor was the ability to isolate the different business processes in place across the organization and segregate the business logic into components that would be both reusable and easier to maintain going forward. The components also will be leveraged in the future when the company begins to offer Web services.

One of the issues this company encountered was the difficulty associated with translating complex business processes and business logic into code. Blaze Advisor's ability to ease that translation and simplify the process was another direct benefit derived from this tool because the language used closely approximates the language used by business analysts. Business partners can maintain the business processes and business logic themselves, which is a huge win for the financial services company because it no longer has to be involved with every change that its business partners want to make. Resulting cost savings occur in terms of business partners being able to modify the rules themselves rather than submitting them into the very busy financial services organization staff and waiting for the IT organization to make the changes.

Using Blaze Advisor has resulted in a shorter time to implement rule changes and better quality control in terms of being able to more accurately interpret the requirements for new rules, implementing exactly the changes that need to be made, and making those changes only once. Using the older in-house tools, technical staff members often made mistakes because they did not fully understand what new rules needed to be created, and time was lost in having the business analysts involved in rewriting the code several times before they got it right.

The financial impact from being able to update complex business logic in Blaze Advisor was significant. Based on a quarterly update cycle, the annualized benefits from using Blaze Advisor relative to traditional coding methods were between

\$260,000 and \$270,000. Over the 3.5-year production time horizon, the use of Blaze Advisor results in a net benefit of slightly more than \$924,000.

ADDITIONAL BENEFITS OF USING BLAZE ADVISOR

By using Blaze Advisor the financial services company has achieved the following additional benefits that were not quantified by this ROI analysis but will result in cost savings for this and future projects. The company is now able to:

- Create and change business rules quickly using its own business analysts thereby significantly reducing the time it would take the IT department to have technical programmers try to make these changes using their homegrown business engine
- Reduce defects in future applications and rules changes
- Implement fewer post-launch enhancements because the rules are initially implemented with greater accuracy because business analysts are able to do this themselves while using better tools
- Do a better job implementing requirements as rules because it eliminated communication problems that previously occurred between the technical developers and business analysts
- Respond more quickly to future required changes and updates to rules and processes because the business analysts are responsible for maintaining the application using a language that is more natural to them
- Establish a future framework that can be leveraged and partially reused for other projects requiring rules creation

NET QUANTIFIABLE COSTS

Costs for the project consisted of license, maintenance, training, consulting and education costs.

Blaze Advisor licenses were purchased in 2001 and 2002. Many developer and server licenses were purchased in each of the two years. The amortization schedule for each license purchase is three years. Maintenance and upgrade fees are assessed annually. Total license cost projected for a year or two beyond the four-year time horizon for this ROI study was \$340,000. Total maintenance cost for the same period is estimated to be \$188,000.

Training and education costs were also incurred during the development phase of the project and amounted to slightly more than \$204,700. These training costs related primarily to nontechnical staff members who needed to learn the process of rule authoring.

Application development time required of professionals was virtually identical to that required to build the application through a more traditional approach and therefore did not represent a net cost.

Total costs for this project are expected to be just under \$733,000.

RETURN ON INVESTMENT

Due to the initial learning curve and upfront license costs, the ROI value for this project (NQB–NQC) started out negative in 2001. A cumulative break-even point will be reached in 2004 as the benefits associated with easier rule maintenance begin to

accumulate. Annual ROI for 2005, the last year of the analysis is 576% due to license costs, which became fully amortized in 2004. The overall ROI for this project is expected to be 134%, with ROI value of slightly more than \$250,000. Although this ROI is not exceedingly high, the ability to achieve a break-even point in just two years of production and the potential for high ROI beyond this time horizon prove to be an attractive combination.

CONCLUSION AND ESSENTIAL GUIDANCE

CONCLUSION

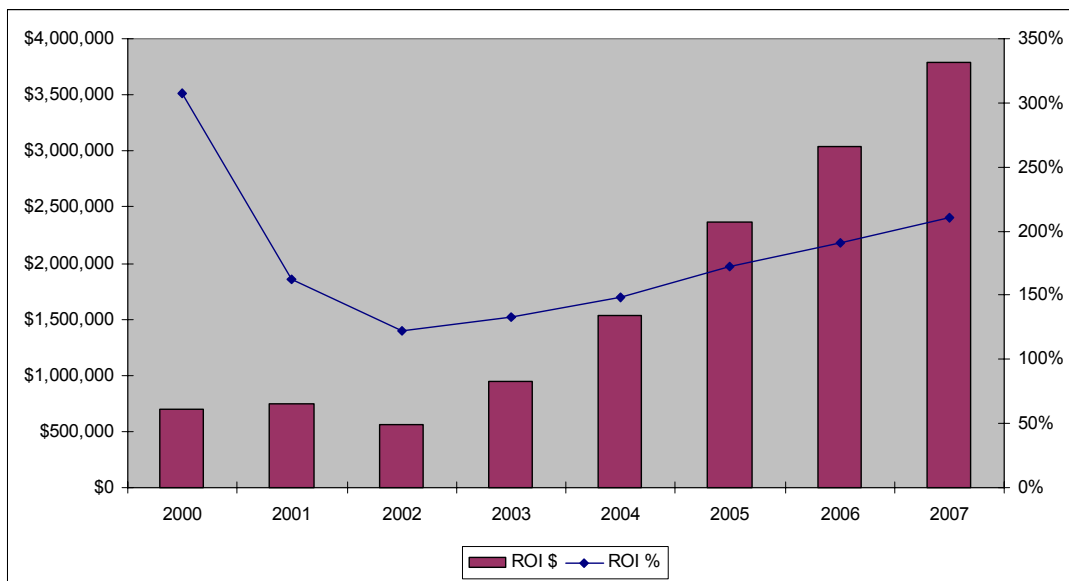
COMPOSITE ROI

Although the time frames for these three cases overlap, the start year for the ROI analyses was 2000 in two cases and 2001 in the other case. Variation in the end year of the analyses was greater: 2002, 2005, and 2007. Despite these variances, IDC has constructed a composite ROI that displays the ROI value (NQB–NQC) and ROI percentage. The assessment of these composite measures is that the ROI value provides a view of the cumulative \$ benefits and cumulative ROI. Figure 1 provides a summary and shows the following:

- ☒ The cumulative composite ROI percentage never dipped below 100%.
- ☒ The overall cumulative composite ROI percentage was 210%.
- ☒ The cumulative composite ROI value was \$3.8 million, suggesting significant savings over the collective time horizon for the analysis.

FIGURE 1

CUMULATIVE ROI VALUE AND ROI PERCENTAGE FINDINGS



Source: IDC, 2003

If the ROI calculations continued beyond the current time frame, the probability is high that the composite ROI would have been even higher.

While it is unusual for a composite ROI percentage to be above 100% (breakeven) early in the ROI time horizon due to initial investment costs, the fact that this did occur (driven by the strong results such as those of the global services company analyzed) does point to some interesting characteristics of rules engines such as Blaze Advisor.

ESSENTIAL GUIDANCE

Companies considering alternative approaches to development methodologies to improve time to market, decrease development and maintenance costs, and increase developer productivity should consider using Fair Isaac Blaze Advisor. This study depicts the significant ROI benefits derived from using this type of development tool. However, the expected and unexpected benefits of using Blaze Advisor that were common to all the participants in this ROI case study analysis are just as important as the ROI results. In all cases, the Blaze Advisor business rules management solution provided real-time savings and increased productivity for conducting business through the use of rules in application development and modifications. The benefits also provided concrete evidence in terms of anticipated shortened development and deployment cycles once the rules management approach was in place.

Blaze Advisor enabled the three companies to help their business users and IT staff jointly maintain rules and reuse rule logic across the enterprise. It also allowed them to develop and update rules more quickly than a more traditional 3GL approach to application development. Equally important was the fact that business managers had the control and power they needed to make smart business decisions quickly and more efficiently. Because business rules could be created and changed quickly using their own business analysts or internal developers, companies were able to reduce the time it would take the IT department to have technical programmers try to make these changes.

Each company also anticipated future benefits in terms of reduced defects when making changes to current and future applications as well as more accuracy in implementing business requirements as rules. Future benefits are also expected in terms of having established a framework that can be leveraged and reused for other projects requiring rules creation — a framework in which developers and business managers could see the flow of rules and more quickly determine where to make changes.

Large organizations should consider the business rules approach before embarking on their next application development project.

COPYRIGHT NOTICE

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2003 IDC. Reproduction without written permission is completely forbidden.