

# Quantifying the Value of a Business Rule Management System

InRule Technology  
651 W. Washington  
Suite 201  
Chicago, IL 60661  
312.648.1800  
inrule.com

---

## Table of Contents

<b>Quantifying the Value of a Business Rule Management System</b>	<b>Page</b>
Foreward by James Taylor	2
Introduction to Quantifying Value for a BRMS	6
IT Costs of a BRMS	7
Application Development	7
Traditional Application Development	10
Rule Enabled Development	11
Application Development Costs	13
Application Maintenance	14
Bridging the Gap Between IT and the Business	16
Business Costs and Benefits	17
Leveraging Investments in Key Microsoft Technologies	19
Exploiting the Power of .NET	20
Case Study: Aon	21
Summary	22
 RuleVantage: A Rule-Powered Tool for Calculating the Value of a BRMS	 23
 About InRule Technology	 24

# Quantifying The Value of a Business Rule Management System

---

## Foreward

*By James Taylor*

I have been helping companies automate and improve decisions for a number of years now. Working with many companies and most of the leading vendors I have seen projects of every size and complexity. The results companies get continue to impress me. Experience shows that using business rule technology to automate a decision shows a positive return almost immediately and broader use across multiple projects can deliver terrific value. This foreward is almost a summary of these projects, intended to show you the kinds of returns you might be able to get. The rest of the paper is also based on experience – InRule Technology’s experience with its customers. This paper, and the calculator that supports it, should help you see the potential for a BRMS in your business. I hope it does.



James Taylor  
CEO Decision Management Solutions

In uncertain economic times companies are unwilling to make investments that will not show a return and do so quickly. There is little interest in or budget for high risk or long-running projects when business circumstances can change so quickly. Projects that adopt a Business Rule Management System (BRMS) and use it to automate and improve high-volume, operational decisions can and will show a return that is both significant and rapid. This ROI comes from improvements in IT costs, better decision making, improved customer service and more.

In my experience, the use of business rules and of a BRMS to manage high-volume, operational decisions have a proven track record in reducing application development costs and application maintenance. It takes fewer developers and less time to specify how a system or service should behave using business rules and a BRMS thanks to the increased expressive power of business rules and the improved verification and testing offered by BRMS. Maintenance of these rules is easier, often dramatically easier, than the maintenance of the equivalent code. Not only are can the business rules be changed independently and safely; business users can participate directly in the maintenance process for the first time. Domain expertise is applied more directly and less time and money are spent making changes. Many business rule implementations can be and have been justified purely using these application development savings.

## Quantifying The Value of a Business Rule Management System

---

However the ROI from improved decision making often dramatically exceeds the savings in development and maintenance costs when using a BRMS. More accurate decision making and the option for more fine grained decisions generate better business results. For instance I have seen a huge return from using a BRMS to deliver customer-specific pricing in an online environment. Increased precision in decision making of this kind can boost margins significantly. Consistency across channels, across customers and across customer service representatives can also show a return, though it can be hard to put a number on it. Consistency can show a hard return when compliance is an issue as being inconsistently compliant results in fines and other costly problems. The business value of agility is clear – time to market determines the value of a new product introduction and the time to enforce a new regulation or to respond to a competitor is measured in fines paid or dollars lost.

Business value is also created when BRMSs allow “once and done processing”. Such processing results in empowered front-line staff (who are thus more effective) and helps reduce the tendency of prospective customers to shop-around because they realize they can close the deal right away and get it off their list of things to do. Eliminating potentially costly activities when they will have no impact (such as inspections for policies that will be rejected anyway) creates value as does freeing up staff time for something more useful. Use of a BRMS can eliminate “do-overs”, reduce the unnecessary purchase of third-party data and services and reduce the number of fines, errors and fraud in a process.

Many companies have shown that adopting a more customer-centric approach pays great dividends. The ROI of improved customer retention, better cross-sell and up-sell and more effective development of relationships with customers can be significant. To treat customers more effectively requires, first and foremost, better control over customer treatment decisions. Ensuring that these decisions are made quickly, accurately and consistently, regardless of channel, is essential. Again and again companies have found that a BRMS is ideal for building a customer hub and improving the value of each customer.

Besides these hard numbers, use of a BRMS can deliver significant soft benefits. Resource scalability is improved when the majority of day to day transactions can be handled automatically. Companies that have used a BRMS to approve policies have decoupled the number of underwriters from the number of policies being written. This allows for rapid growth without constant hiring of new staff. Encapsulating the know-how of experienced staff in a BRMS also helps ensure that tribal knowledge and hard won experience can be applied to every transaction, every process. Organizations making broader use of business

## Quantifying The Value of a Business Rule Management System

---

rules also create value through their ability to work down their systems backlog. Each system or modification in the backlog, after all, has a business value. Organizations with a project backlog could be said to have what I call a “value backlog”. Use of a BRMS to make development and maintenance more efficient can allow you to make progress working down your backlog, creating a substantial business benefit.

I have seen many BRMS projects where the ROI is clear right from the start. It is obvious to all concerned that the cost of being unable to respond to a crisis in a timely manner or the fines that will be imposed for failing to be compliant with a new regulation will dramatically exceed the cost of solving the problem with a BRMS. These kinds of projects are justified by the known cost of doing nothing or of trying to solve the problem using traditional IT approaches. In the current economic climate the risks of doing nothing, of the status quo will rise significantly. The flexibility and agility delivered by a BRMS may be measured not in dollars, but in company survival.

One final comment about the benefits of a BRMS: When I talk to companies that have successfully implemented a BRMS about the benefits they often talk about strategic control. Use of a BRMS puts the business, often for the first time, in control of their systems. They are able to make changes as and when business conditions require it. This strategic alignment between business needs and information systems is often reported by BRMS users as the most powerful outcome of adopting the technology. It can be hard to put a dollar value on this but the benefits in terms of business growth and of profitability can be enormous. In a time of economic uncertainty and increased competitive pressure, this kind of strategic control will mean the difference between success and failure.

InRule Technology have applied their experience in many different projects to help you see what the ROI would be for adopting business rules in your environment. This white paper provides an experience-based approach for quantifying the value of implementing business rules technology. Some of InRule Technology’s customers – practitioners whose livelihoods depend on successful application development and deployment – have reviewed and contributed to this paper. InRule Technology also provides a rule-based value calculator, RuleVantage. And while no such calculator is perfect InRule’s will give you a powerful sense of what’s possible and how much that is worth to you. This will help you adopt business rules and start to see the benefits others have seen.

I have seen many companies get a great return on their business rules investment. I hope you will join them.

*James Taylor*

# Quantifying The Value of a Business Rule Management System

## Introduction

### \*Sorting out the Acronyms

There are several options for organizations considering a rule technology solution, with varying degrees of depth and breadth.

The core of any business rules technology is the business rule engine, or BRE.

A full business rule solution will include not only the engine, but also authoring capabilities for technical and business users, rule management, and application integration capabilities. Some use the term Business Rule Management System (BRMS) to refer to this full business rule solution.

In this paper, the term BRMS is used to describe the full business rule solution, which includes capabilities for authoring, management, integration, and execution of business rules.

IT leaders are being challenged more than ever to rationalize expenditures of all types. Organizations considering the use of business rules technology for the first time, or those that wish to extend their use of rules technology to additional applications, may be asked to justify the recommendation to management.

Significant benefits – in the form of lower application maintenance costs and increased responsiveness to business condition changes – have been realized by organizations of all sizes and industries as a result of deploying business rules technology in the right applications. Less often quantified but increasingly important: in a competitive market failure to adapt rapidly can be a corporation's downfall. Many companies that were unable to adapt rapidly have experienced significant capital exposure while they waited for systems to be updated<sup>1</sup>.

Organizations that have deployed rule technology-- referred to as a Business Rule Management System (BRMS)\*-- agree that it has delivered significant value to their organizations, particularly in terms of saved programming hours in developing an application. This paper includes proof points from some of them.

While the value of a BRMS certainly includes saved programmer hours in developing an application, it goes beyond the initial development to include ongoing maintenance and IT operational costs. And the overall value of deploying a BRMS goes beyond IT costs to include potential costs and benefits to the overall business, including risk and compliance management, time to market, and revenue impact.

---

<sup>1</sup> [\*It's Not the Big That Eat the Small...It's the Fast That Eat the Slow: How to Use Speed as a Competitive Tool in Business\*](#), by Jason Jennings and Laurence Haughton

## IT Costs of a BRMS

### Application Development

While often challenging, estimating the initial application development cost is the most frequent scenario for quantifying the value of a BRMS, as calculating this cost is required for funding approval. The primary cost for application development is typically that of personnel or outsourcing. The most basic way of determining costs are to estimate the person-hours required by role, then multiply the hours by the hourly cost.

### Application Development Roles

Most projects require the following roles for both initial application development and ongoing application maintenance.

#### *Architect*

An enterprise architect is responsible for strategic software decisions to align IT and business goals across multiple applications and business divisions. These decisions may direct development methodology as well as the selection and implementation of hardware and software. An application architect has responsibility for the design methodology of the code for a single software application as well as key technology choices, including software and sometimes hardware.

While an architect's time will be most intensive at the beginning of a project as decisions are made around design methodology and technology selections, the architect will provide guidance throughout the lifecycle of the project. Architects are also often involved at a project's completion for audits.

#### *Programmer*

The backbone of any application development project, programmers write software programs according to the specifications determined by more senior programmers and/or analysts within the overall methodology and design defined by the architect.

In order to deliver the application's required functionality, programmers are responsible not only for writing transactional systems, but also for developing user interfaces, setting up data schemas, and selecting and integrating third-party tools.

For most programmers, the process of interpreting requirements and rules is a challenge that directly impacts the delivery of applications. If the interpretation of the requirements and/or rules from the business is

incomplete or inaccurate, the design or coding could be wrong, introducing expensive and time-intensive changes. Because programmers, subject matter experts, and business analysts use different tools and different terminology, collaborating on business requirements and rules is difficult and often does result in the misinterpretations that result in cost overruns and delays.

## ***Subject Matter Expert***

Outside of IT, there are subject matter experts (SMEs) who define business requirements for the application. In an insurance company, this may be an actuary or underwriter; in a financial institution the SME may be a risk manager; in a retailer, a product or program manager; in healthcare, a policy program manager.

Subject matter experts are the people who understand the business and therefore can determine what the business logic for an application should be. These business requirements may be expressed in a number of ways, from actuarial tables to application requirements to formulas in spreadsheets. In most organizations, the SME will hand over these requirements to a business analyst, who then has responsibility to translate and negotiate between the SME and IT.

## ***Business Analysts***

Business Analysts play a key bridging role in the development of applications. In many organizations, the business analysts sit in IT. They understand the business and its current systems. Applying their understanding of the business and its goals, they analyze the current system and provide recommendations for improving and potentially automating processes. Business Analysts are responsible for bringing together the business requirements from the SME and the architectural and other technical requirements from IT within a set of detailed requirements that will enable IT to deliver an effective application that does what the SME needs it to. The Business Analyst defines the “what” of a system, as the architect and programmer define the “how.”

## ***Tester / Quality Assurance***

Programmers are responsible for unit testing the software they write or maintain. Additional system, performance, integration and/or regression testing may be done as the new application moves from development to production.

## *Project Managers*

The time required for project management is proportional to the complexity and duration of the project. Projects with larger teams, multiple locations and time zones, more interdependencies, and longer duration will drive higher project management costs.

## *Release / Deployment*

Additional costs incurred in moving the new or changed application from development to production, including distributing new software to a new server or hundreds or thousands of desktops.

## *Education / Training*

The design, development, and deployment of a new application require education on business requirements and higher level software architecture design goals and guidelines. Team members may also require training as new technologies and tools are introduced.

## **Operational Costs**

In addition to the cost of personnel, there are also ongoing IT costs related to every application. These include software (including the cost of the BRMS), hardware, administration and networking for all environments, including development, test, staging, QA, production, and disaster recovery.

## **Defining Business Logic**

Business logic expresses a business policy, primarily to drive a decision. For example, business logic defines a financial institution's policy for loan approval, including what information is needed to make the decision, what factors and rules govern the decision, what weight is given to each factor, and what information is returned when the decision is made.

Traditionally, business logic is buried within an application's hard-coded application logic, where it can be difficult to find and change. A BRMS externalizes business logic and enables non-technical staff to write and maintain it.

## Traditional Application Development

### Traditional Application Development:

#### Hard Coding Business Logic

When business logic has been hard-coded into applications, changes to that business logic are like any other application changes: they must be done by programmers with a related high cost.

In addition to the straightforward cost of a programmer's time to change a rule, the cost and complexity of hard-coding rules is often exacerbated in two ways:

A single rule implemented redundantly across systems, requiring a scavenger hunt to find and consistently change and test all instances of the same rule.

A single rule "schmeered" across multiple layers, making that rule difficult to identify. For example different aspects of the same rule regarding the validation of data such as Employee ID may be used in the User Interface, the Service Interface (when saving the data) and the batch program processing the data. The UI rule may validate only that a number is required if the user has checked the "employee" box; the service interface may validate the format of data entered; and the batch process may cross-check the number entered with an employee database.

A Service Oriented Architecture (SOA) approach has been used successfully by many firms to break apart monolithic applications into more manageable pieces that can be more easily and quickly updated and that can be shared across applications. However, SOA does not relieve the programmer of the responsibility of maintaining business logic and the challenges of implementing a system to meet user requirements still remain. Changes in the market—such as changes in taxes, new regulations, or the introduction of new products—mean changes to business logic. Applications must be changed to reflect a new tax the day that it becomes effective, but not before. Additional procedures, reporting, or auditing may be required to comply with new government regulations or rules of corporate governance. And new products must be continually introduced—with new features, pricing, and financing—to remain competitive in today's global marketplace.

For any and all of these changes, the SME must work with the business analysts to document and detail the change and when it must become effective. The business analyst must ensure that the change request is effectively communicated to the application development team. And the programmer needs to interrupt his or her current work to make the change. The cost of making a change to business logic is not only the cost of changing the code, but

## Rule-Enabled Development

includes the entire change management process. Any change to business logic must be managed through the application lifecycle.

For example, an increase in the retail tax rate requires a change to the application code to implement the new rate for the geographies where it applies. This simple code change requires not only the time of the programmer, but also the time of the SME and business analyst to implement the change; the QA staff to test the change; and the project manager to ensure the change is put into production on time. The cost of the change is spread across the entire application lifecycle.

### **A New Model for Developing and Maintaining Business Logic: Rule Enabled Development**

When business logic is hard-coded into application logic, programmers are the only personnel who can write and maintain this logic. They will devote more hours both initially and over time on application maintenance, and business logic is not easily shared across multiple applications. A BRMS goes beyond the promise of SOA<sup>2</sup> by enabling non-technical personnel to author and maintain the logic, rules, and calculations that power decision points in applications, in rules that are reusable across applications yet maintained in one place.

#### **Empowering Business Users**

For some organizations, this means that SMEs such as actuaries and risk managers can author and maintain business logic themselves. Other organizations prefer to let the SMEs drive the changes through the business analysts, who implement the changes in the BRMS. And still other organizations prefer to keep all logic in the hands of IT. In about 50% of InRule Technology clients, SMEs or business analysts do all the rule authoring, freeing the developer from writing business logic. A BRMS can also provide different levels of authority for different authors so that only the right person edits the right rule, depending on their role/responsibility, geography, company division, etc.

A BRMS enables both technical and non-technical users to unit test their business logic, without a separate test environment, thereby reducing testing costs and speeding time to market.

Whatever the model, using a BRMS delivers application changes more quickly and cost effectively than hard-coding. When a BRMS is used, the amount of

---

<sup>2</sup> A BRMS can also deliver on the promise of SOA by deploying rules as callable services.

## Quantifying The Value of a Business Rule Management System

---

time a programmer spends developing and maintaining rules will be reduced, due to both the efficiency of writing/maintaining rules with a BRMS and the ability of non-technical personnel to write and maintain rules, offloading work from the programmer.

The work of the business analyst is also eased, as the number of requirement iterations is reduced and the business analyst (or SME) is able to make changes directly. And while the need to communicate and negotiate changes with the SMEs and development is not eliminated, it is made much easier. A BRMS streamlines this process from iterative, error-prone interpretation to more effective communication, negotiation and collaboration through a common language and tool.

A BRMS benefits not only the project bottom line by reducing the number of programmer hours required for the project, but also benefits the IT organization, because programmers are better able to focus on more complex, higher value projects. Additionally, because changes can be made directly by the personnel requesting the changes, application development backlogs are reduced and time to market improves.

Organizations deploying a BRMS for the first time face not only the challenges of deploying a new technology, but also the challenges of changing the way people work. Depending on the skills and mindset of the individuals involved and the processes of the organization, varying amounts of investment are required to make this change. For organizations making a drastic change in how they work, the initial cost savings of a BRMS may be minimal, but the longer term benefit—even over just two or three years—will be dramatic.

A BRMS may reduce project management costs by reducing the overall project delivery time. And because using a BRMS reduces the number of hand-offs between SMEs, business analysts, and programmers, complexity is reduced, potentially reducing the project management cost as well.

# Quantifying The Value of a Business Rule Management System

## Application Development Costs

By externalizing business logic and managing it separately from the application code and logic, a BRMS does for business rules what DBMSs did for data— making them true assets of an organization.

### Initial And Ongoing Application Development Costs

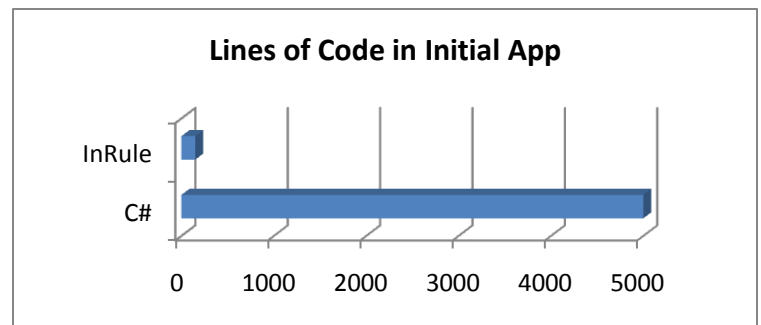
While the same roles and skills are required for both initial application development and ongoing maintenance, there are special considerations for estimating the impact of a BRMS on application maintenance. As many organizations find that the greatest value a BRMS can bring is in reducing application development costs, many others find that a BRMS's value increases over time because of its impact on maintenance costs. It is therefore important to calculate the value over one year and over three years to get a holistic picture of the possible savings in maintenance.

A business rule engine fundamentally changes the nature of business rules from bits and bytes accessible and understandable only by programmers to an organizational asset accessible to the business community by pulling it out of hard code and externalizing it.

An InRule Technology client developed a prototype and sought to compare deploying the same rules-powered application using C# and InRule. Their value quantification was much simpler than the model described in this paper, but it is no less compelling, as it demonstrated a BRMS's value both in initial application development as well as in application maintenance.

It took the following to develop the initial application:

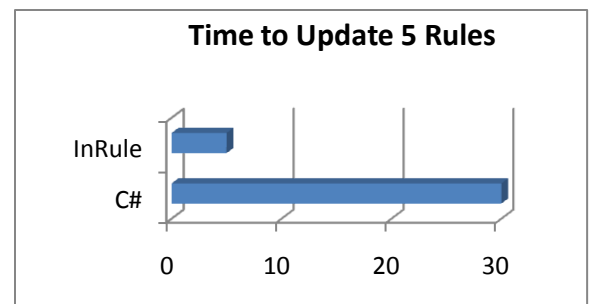
- 5,000 lines of code with C#
- 150 lines of code with InRule



And this to add 5 new business rules:

- 30 minutes with C#
- 5 minutes with InRule

In this example, while the client found immediate value in a BRMS for initial application development, the overwhelming benefit was in reducing the long term application maintenance costs.



## Application Maintenance

An insurance company discovered that in some cases changing one rule required them to touch 50 files! Normal rule changes needed to be made in under 30 days; urgent rule changes need to be completed on demand. This was not possible with hard-coded logic.

A top 5 US bank using InRule for risk rating scorecards created an InRule powered application that reduced their maintenance backlog 68%, and significantly enhanced their time to market. They were able to save time on both the initial application and on maintaining the rules.

## Application Maintenance Considerations

Even InRule Technology, provider of business rule technology, doesn't believe that *all* business logic belongs in a BRMS. Business logic that changes frequently and/or that logic should be authored or maintained by non-technical users is the best candidate for externalization to a BRMS. One of the easiest ways to identify logic that belongs in a BRMS-powered application is to review the Application Change Request Log. Consider analyzing your project requests for changes to rules, calculations, validations, and system notifications. Think about how many files need to be touched in order to update a single rule. You should consider both the time to market requirements of the business as well as the savings you can realize if these changes are managed by a business analyst or SME using a BRMS.

Just as hard-coding all business logic is the wrong answer for most applications, using a BRMS for all business logic may also be the wrong answer. To gain the most value from a BRMS, look for the instances of greatest variability when determining whether or when to use a BRMS rather than hard-code. For many clients, the Application Change Request Log is the key source of data for analyzing variability. And reducing the Application Change Request Log is a key benefit of deploying a BRMS.

## Calculating the Cost of Rule Changes

We've seen organizations estimate the cost of rule changes using two approaches:

- Estimating the average cost per rule change and the number of changes per year, generally based on a prototype
- Estimating the monthly or annual cost of application changes based on current and projected spending

## Estimating the Number of Rule Changes

Prototypes of BRMS-powered applications generally include a small number of representative rules. Sometimes the cost and effectiveness of a BRMS-powered application is compared with hard-coding. The cost differences between the prototype versions are then used to extrapolate ongoing maintenance costs. For example, a prototype may include 10 representative rules changed once. If the production application will contain 200 rules with 20 rules that will change monthly (240 rule changes per year), the monthly cost would be estimated at roughly 24 times the cost of changes made to the prototype.

# Quantifying The Value of a Business Rule Management System

---

“It is like night and day if you look at how far we have come editing rules. We had one application where if you wanted to change a label or a comma or a misspelled word, it took **50 pages of business requirements and 3 months of programming**.

Today we can actually change something like this including formulas in **literally 5 minutes** and publish to market.

The ability to add rules, even significantly complex rules that are modified on a continuous basis, those that have monthly releases, this is one of the state of the art features of InRule.”

*Risk Management Executive,  
Top US Bank*

## Estimating Monthly or Annual Application Rule Changes

Organizations that have deployed rule changes sometimes compare the previous cost of maintaining an application with the new cost of maintaining the BRMS-powered application, rather than using the detailed method described earlier.

## Other Application Maintenance Costs

In addition to looking at the cost of the rule changes, you need to include all the functions that surround application development, including the time of architects, programmers, project managers, SMEs, business analysts, and QA/testers.

In many cases, the programmer is not involved at all in application maintenance that consists solely of rule changes. For example, using a BRMS, a 0.05% increase in the retail tax rate no longer requires a programmer to make a change to the application code. The business analyst or SME makes the change directly, with no custom coding or involvement from the programmer.

Rules can be changed, tested, and rolled out discretely, with appropriate controls on the publishing of rules from development to production, but without the need of the full application management process that code changes would require. A BRMS also provides auditability of changes as well as the ability to easily rollback changes if necessary.

A BRMS with a strong authoring environment also enables the business analyst or SME to spend less time negotiating or documenting business requirements, making their time more productive as well. Because a BRMS should also include a stand-alone testing tool, the business analyst/SME can do the unit testing, leaving only regression testing for the QA team. And because a BRMS enables a SME or business analyst to make changes directly, there are fewer “moving parts” for a project manager to manage.

## Bridging the Gap Between IT and the Business

### From Over the Wall to Bridging the Gap

In traditional application development, there is sometimes the problem of requirements being “thrown over the wall” to development. Many factors create chasms between the business and IT, from vague or non-specific requirements to varying priorities and limited resources. The chance for error increases as the distance between business requirements and programming grows.

In a traditional software development model, the best case scenario would look something like this:

- The SME writes up a changed business requirement and provides it to the business analysts
- The business analysts analyzes the new requirement against the existing system and makes a recommendation to the developer in the form of a requirements document or change request
- The developer interprets it, codes it and tests it
- The developer goes back to the SME for approval and (in the best case scenario) the change is approved.
- If not, we go back to the previous or even first step.

A BRMS helps introduce a new way of thinking and working by encouraging collaboration between IT and the business through a common language and tool. In this model:

- The SME, business analyst, and programmer work together to agree on the logic requirements
- The programmer integrates the rule engine into the application and creates an authoring environment (and may optionally author some rules)
- The SME or business analysts modifies associated rules and verifies that results are correct

Rather than throwing requirements over the wall, Subject Matter Experts, Business Analysts and Programmers team together to define requirements and write and maintain rules, resulting in a streamlined process and fewer errors.<sup>3</sup>

---

<sup>3</sup> For a more detailed discussion, see *Getting to the Handshake: A ROAD Map for IT/Business Collaboration* by Loren Goodman, available on [www.inrule.com](http://www.inrule.com).

## Business Costs and Benefits

### Opportunity Costs and Benefits

To this point the discussion has focused on costs related very directly to IT and the application itself. However, the CIO and Line of Business Manager should also consider the very real opportunity costs and benefits related to the application.

### Compliance and Risk Management

Many organizations turn to a BRMS because it provides the tools necessary for regulatory compliance and corporate governance, particularly for auditing and reporting purposes. With a BRMS, you can ensure that your applications are updated on time to comply with new regulations; you can maintain appropriate business rules for different geographies; and you can easily demonstrate to governing agencies what rules were used to drive which decisions.

Organizations have also used a BRMS to calculate the risk of activities such as lending.

The cost of not using a BRMS can be measured in several ways. Without a BRMS, what would be the cost of:

- Audits to determine whether regulations were applied correctly and consistently to business decisions across multiple geographies
- Manual reports to document how software applications reflect regulations, including geographic and date/time considerations
- Fines when your organization does not comply with regulations because you cannot prove compliance
- Legal fees to fight fines or other penalties for compliance violations
- Other professional fees

### Time to Value

The driving force for many adoptions of a BRMS is the requirement to get application changes implemented quickly and products to market faster. Global competition makes new product introduction more important and global regulations make new product introductions more complex. A BRMS can reduce the time to value from years to months or weeks.

There are two ways to evaluate the impact of a BRMS on time to market: the cost of delays, and the incremental revenue from introducing a new product or updated application.

### Cost of Market Delays

When comparing project plans for an application using a BRMS vs. without, consider not only total person-days, but also elapsed days—what is the

difference in terms of time to market. For many organizations, there is a hard cost in delaying new products.

A BRMS can help companies introduce new products much more quickly, reducing the cost of market delays, as compared with traditional hard-coded methods.

## Incremental Revenue from New Products

Happily, for most organizations time to market concerns are about *opportunity*: being able to introduce new products into new markets to gain additional revenue. Every incremental day in market is incremental revenue and a BRMS can help increase the number of days in market.

## Customer Service and Satisfaction

For some organizations such as those in the services industry, customer satisfaction is a key performance indicator and can often be assigned a specific value or cost. An application that provides key decision-making information may impact customer satisfaction based on the quality of the information and how that impacts customer behavior. In customer purchasing applications, providing better information may drive additional revenue. In customer service applications, providing better information may drive reduced customer service costs and improve loyalty. Organizations that can place a specific value on customer satisfaction may include this value in their calculations.

## Effectiveness

In addition to the many benefits already discussed, a BRMS helps improve the effectiveness of an organization. Some organizations may have the methodology in place to measure these gains in effectiveness, but all organizations should take note of at least two key ways in which a BRMS can improve effectiveness:

- Making better decisions and continually improving decision-making
- Re-use of rules across applications, delivering the value of SOA today
- Consistency and Visibility of business rules
- Better usage of personnel
  - Free critical IT staff from mundane application changes and enable them to focus on strategic projects
  - Enable SMEs and business analysts to express their business requirements more accurately in less time
  - Increase collaboration between IT and Business to reduce errors and improve application quality

## Leveraging Investments in Key Microsoft Technologies

“InRule is now the leading pure .NET choice”

From *A Framework for Business Rules Platform Selection*, presented by Forrester analysts at Business Rules Forum, November 2009. InRule was listed as one of the three strongest BRMS options.

Organizations using .NET as the primary development platform have an opportunity to leverage their investments in Microsoft technologies and skills. This section and the next will help cost-conscious organizations understand how to continue leveraging the value of those existing investments.

### *Visual Studio*

Developers should have the option to write and test rule-enabled code as well as author, test, and maintain business rules directly within Visual Studio. Allowing developers to work within the familiar Visual Studio environment to set up authoring domains and to author, edit and manage rules shortens the learning curve and streamlines rule-enabled application development.

### *Word*

Subject matter experts at many organizations already use Microsoft Word to document business requirements and rules. Using a rule authoring tool integrated with Word allows users to update business rules directly from within a familiar environment. This streamlines the transition from documenting requirements to authoring logic and eliminates the need for separate requirements documents.

### *Active Directory*

Importing Active Directory Users and Groups into the BRMS catalog simplifies the creation of user roles and groups. Once Users and Groups are imported, the administrator can assign roles and permissions to them.

### *BizTalk Server*

A .NET BRMS should provide interoperability with a BizTalk Server Orchestration, providing a way to call the rule engine directly or through the use of static .NET Methods that return an XML document with rule results.

### *Windows Workflow Foundation*

An out-of-the-box WF activity allows calls to the rule engine from a workflow by dropping the activity onto the flow and setting its properties.

### *SharePoint*

Many organizations use SharePoint to manage documents and document-centric business processes, but find that extensive customization is required to use SharePoint for complex, decision-intensive processes.

Using a BRMS with a Business Process Management (BPM) tool built on top of SharePoint significantly extends the power of SharePoint for advanced business process management and business logic automation – *without programming*.

## Exploiting the Power of .NET

In addition to support for popular Microsoft tools such as Visual Studio and Word, developers should consider capabilities that support other .NET features

### .NET Considerations

While supporting .NET simply means that application will run, *exploiting* .NET means the application takes advantage of many .NET-specific capabilities that can only be leveraged when the application is designed for a .NET environment and built by .NET experts.

A BRMS that is designed, built, and optimized for .NET delivers many capabilities that generic BRMS can't (for example BRMS that were built on J2EE and ported to .NET.)

**.NET Classes, XSDs, and Databases** – By harvesting schemas from existing .NET classes, XML Schemas and Database tables rather than re-creating them, authoring takes less time and is more accurate.

**.NET Generics** - Advanced support for generic classes and methods, which combine reusability, type safety and efficiency in a way that their non-generic counterparts cannot.

**.NET Object Graphs** – A BRMS should be able to execute against .NET object graphs at run-time, acknowledging relationships among objects in the graph.

**.NET Collections** – Support for native .NET collections is important. For example, a policy may contain a collection of borrowers, each of which contains collections of assets and liabilities.

**.NET Component Assemblies** – The SDK should include a set of .NET assemblies containing functionality that developers can easily embed within their own applications. The SDK should require no additional tools to install, and enable .NET developers to drop controls (such as decision tables and a language editor) into .NET applications.

**.NET DataSets.** Organizations that have adopted .NET as their development platform may have standardized on .NET DataSets, so a BRMS should operate against these natively. Without this native support, additional code to manage data must be written and maintained— resulting in significant overhead to development resources and potentially slower performance.

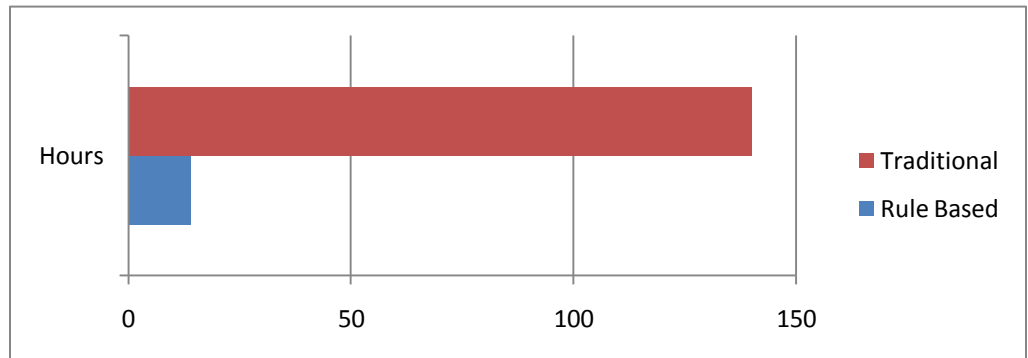
**.NET Events.** A BRMS should expose events to consuming code using native .NET Events. For example, a rule author may indicate that the rule engine should raise a .NET event when certain conditions are met and pass data to the application executing rules. This allows the developer to perform an application-specific action when those conditions are met during rule execution.

## Case Example: Aon

### Aon Empowered Business Analysts to Make Changes... In Less Time Than Experienced Programmers

Aon provides innovative solutions in insurance and risk management, human capital consulting, and insurance underwriting. Each industry has unique requirements. Aon's Winbroker is a .NET application that handles accounting, policy management and issuance, reporting, underwriting, rating, and claims for the insurance programs they underwrite. Rules authored in InRule are used extensively in WinBroker for complex calculations for rating business logic for underwriting and policy form selection

“For one of our insurance coverages, it took an experienced programmer 120-160 hours to develop the logic [without InRule] for the rating program. An actuary, using InRule, was able to build and test the same logic in 14 hours. In addition, they used the tool to express their requirements instead of having to write up a huge Word document.” **Underwriter, Aon Corporation**



# Quantifying The Value of a Business Rule Management System

## Summary

Implementing a BRMS has delivered significant value for more than 170 InRule customers worldwide, including reducing initial development time, reducing critical time to market, and enabling compliance.

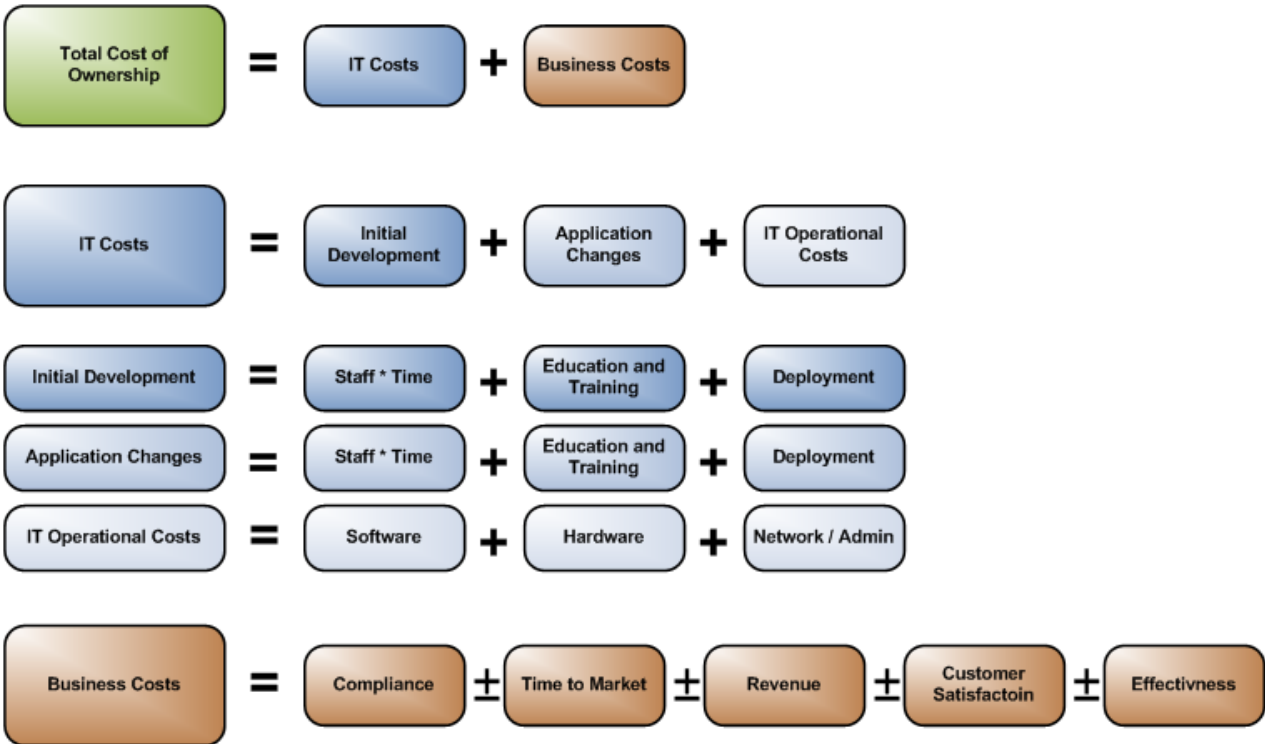
## Delivering Quantifiable Value with Rule-Enabled Development

Organizations of all sizes and industries have realized quantifiable value by implementing a Business Rule Management System. Rule-enabled applications have delivered significant benefits in the form of lower application development and maintenance costs by enabling business analysts and subject matter experts to author and business logic without custom coding or expensive programmer time.

A benefit less often quantified but increasingly important is an organization's ability to adapt rapidly to changing business conditions. While many organizations have calculated significant value in considering only the IT costs, more are also considering benefits to the overall business—including risk and compliance management, time to market, and revenue impact. Calculating the overall value of deploying a BRMS goes beyond IT costs to include potentially significant value to the business.

The graphic below provides a summary of the factors to consider when quantifying the value of rule-enabled development versus traditional application development.

### Calculating Total Cost of Ownership



# Quantifying The Value of a Business Rule Management System

## RuleVantage: A Rule-Powered Tool for Calculating the Value of a BRMS

Using the experience-based information presented in this white paper, InRule Technology has developed **RuleVantage™**, a rule-enabled application to help quickly estimate the costs and savings an organization can realize by deploying a BRMS. Use RuleVantage to:

- Quantify the advantages of deploying a business rules engine
- Gain a vantage point from which you and your colleagues, clients, or partners can clearly compare the costs and savings associated with rule-enabled development and traditional application development
- RuleVantage Features:
  - Rule-enabled application, powered by InRule
  - Based on experience of InRule Technology, our customers and partners
  - Enables consideration of value within IT or across IT and the business
  - Uses data specific to your application and organization.
  - Configurable so that you can add supplemental factors

Visit the Resources page on [www.inrule.com](http://www.inrule.com) to download RuleVantage.

The screenshot displays the RuleVantage application window. At the top left is the InRule Technology logo and the product name 'RuleVantage'. A navigation sidebar on the left includes categories like 'Initial Application Development', 'Application/Rule Maintenance', 'IT Operation Costs', 'Business Costs', and 'Supplemental Questions'. The main area features a 'Summary' table comparing 'Traditional' and 'Rules Enabled' costs over 'Year 1' and 'Years 2/3'. Below this is a detailed table for 'Initial Application Development' comparing 'Traditional' and 'Rules Enabled' across various skills (Architect, Analyst, Programmer, Tester, Project Manager, Training, Deployment) with columns for Annual Salary, Days Effort, Cost, and Savings. A 'Calculate' button is at the bottom right, and a note states: 'Rules-enabled application development and maintenance has lower total costs'.

	Traditional	Rules Enabled
Year 1	\$1,013,779.38	\$292,164.00
Years 2/3	\$1,083,971.69	\$472,587.08
Subtotal	\$2,097,751.08	\$764,751.08
BRE License	n/a	0.00
BRE Maintenance	n/a	\$0.00
Savings		\$1,333,000.00

Skill	Annual Salary	Traditional		Rules Enabled		Savings
		Days Effort	Cost	Days Effort	Cost	
Architect	\$130,000.00	65	\$32,500.00	44	\$22,000.00	\$10,500.00
Analyst	\$70,000.00	65	\$17,500.00	260	\$70,000.00	(\$52,500.00)
Programmer	\$120,000.00	780	\$360,000.00	468	\$216,000.00	\$144,000.00
Tester	\$70,000.00	65	\$17,500.00	65	\$17,500.00	\$0.00
Project Man...	\$90,000.00	130	\$45,000.00	130	\$45,000.00	\$0.00
(Training)	\$100,000.00	0	\$0.00	1	\$384.62	(\$384.62)
(Deployment)	\$100,000.00	25	\$9,615.38	12	\$4,615.38	\$5,000.00
*						

Total	\$482,115.38	Total	\$375,500.00	Savings (loss)	\$106,615.38
-------	--------------	-------	--------------	----------------	--------------

## About InRule Technology

---

### About InRule Technology



InRule Technology delivers InRule, the premier .NET solution for authoring, managing and executing business rules. InRule helps customers better manage and automate business decision logic without expensive custom programming, delivering **Business Logic Automation** and enabling organizations to respond quickly to rapidly changing business requirements.

InRule supports a spectrum of business rule authors – from subject matter experts to business analysts to developers – who can more easily write sophisticated business rules and complex calculations. Improved **Business Decision Management** results from an integrated approach to authoring, storing, managing and executing rules.

With powerful rule management capabilities, a rich SDK, and the ability to execute rules in process or as a service, InRule enables a flexible **Enterprise Logic Infrastructure**, allowing business logic to be managed as an enterprise asset and used consistently across multiple business processes, applications, and platforms.

A Microsoft Gold Certified Partner and member of the Microsoft Business Process Alliance, InRule Technology is committed to helping customers leverage the power of .NET.

For more information, including white papers and a free trial download, visit [www.inrule.com](http://www.inrule.com).

InRule Technology  
651 W. Washington  
Suite 201  
Chicago, IL 60661  
312.648.1800  
[inrule.com](http://inrule.com)