ENTERPRISE BUSINESS INTELLIGENCE ROI

ANALYSIS OF THE FEATURES, BENEFITS, AND COSTS OF "BEST-OF-BREED" INTEGRATION VERSUS A SINGLE ARCHITECTURE

THE BEST-RUN BUSINESSES RUN SAP

HIRED BRAINS, INC.

CONTENTS

ABOUT THE AUTHOR

Neil Raden is the founder of Hired Brains, Inc., **www.hiredbrains**.com. Hired Brains provides consulting, systems integration, and implementation services for clients across North America and Europe. Hired Brains also provides consulting, market research, and advisory services to the business intelligence (BI), data warehousing, and decision support industry. Based in Santa Barbara, Calif., Neil is an active consultant and widely published author and speaker on data warehousing, BI, and information technology strategy. He welcomes your comments at **nraden@hiredbrains.com**.

EXECUTIVE SUMMARY

Return on investment (ROI) analysis of business intelligence[1] (BI) can be a useful technique for evaluating alternatives, but only if the process is rigorous and evaluates all of the true costs over the life cycle of the technology. Incomplete and inconsistent models can lead to false positives, such as favorable-looking scenarios that are in fact just the opposite. Because BI, including data warehousing, represents a family of tools and techniques, it is common in the industry to build these applications incrementally and to integrate a large number of technologies. The common term for this approach is "best of breed," though in many cases, that description may be too optimistic. The premise of this paper is that lifetime costs of BI are inordinately high in best-of-breed implementations, and that ROI analysis often misses this reality as a result of industry biases, tired methodologies that are too hard to displace, and (component) vendor persuasion. A single architecture, from a single vendor, is a very real alternative, especially given the marked propensity for information technology managers to think in enterprise terms more than ever before.

The components of today's BI environments evolved from many different disciplines. Front-end tools include reporting, OLAP,[2] ad hoc query, statistical modeling, and data mining. All of them served different audiences and addressed different needs within organizations. Data management components, including relational, dimensional,[3] multidimensional OLAP,[4] and proprietary structures, emerged in the same way and continue to coexist in the delivery of analytical information. There are two published metadata[5] standards, neither of which has been widely adopted. The tools for extraction, transformation, and loading (ETL) of data from source to target systems run the gamut from COBOL generators to metadata-driven realtime data interpreters and innovations in EAI[6] and EII[7] expand the reach of ETL into real-time integration and the ability to "federate" queries in real time by reaching beyond the data warehouse to fetch information from other systems, message queues, and even external sources and transform it into the BI schema dynamically.

Each of these areas emerged and evolved within its product ecosystem, not necessarily in concert with any of the other areas. The process of selecting tools and integrating them to provide enterprise business intelligence, and sustaining it, has gone from challenging to intricate. The functionality of all of the tools has expanded, causing overlap and incompatibility, and the expansion is continuing. In short, assembling a successful and sustainable BI environment has gotten more, not less, difficult.

In addition, the promised gains from the application of BI in organizations are disappointing and existing methodologies do not address that part of the problem, focusing instead on the data management issues. As the lines between operational and analytical processing become blurred, a trend that is already evident, enterprise-ready BI will become imperative. Organizations with poorly designed or assembled BI environments, lacking integration with the full portfolio of systems in an organization, will see their costs soar as they attempt to retrofit the compatibility, functionality, scale, and interoperation that were overlooked in the initial design. No one can afford to underestimate the breadth of this problem.

Assembling a BI environment from "best-of-breed" components is a deceptively simple and often lower-priced alternative (initially) to purchasing a single integrated architecture. Most "data warehouse practitioners" use this approach, perpetuating a clear bias for it in the marketplace. The purpose of this paper is to examine the issues in these decisions using a return-on-investment approach.

1) In the broader meaning of BI, all of the aspects of analytical and reporting processes are included. The term BI *tool*, however, typically means the front-end software that is manipulated by business people for query, analysis, reporting, and, recently, alerting through the application of analytics.
2) OLAP: Online Analytical Processing, a phrase coined by E.F. Codd (with the assistance of the marketing department of Arbor Software, now Hyperion Solutions) in 1992 to distinguish interactive query, navigation, and analysis of dimensionally modeled data from OLTP, Online Transaction Processing.
3) Dimensionally modeled within a relational database using a "Star Schema" or some other variant to normalized, transactional designs.
4) Multidimensional OLAP: MOLAP, physical structures optimized for OLAP data. They may be permanent databases, permanent "cubes" or slices of the model, transient structures, or hybrids of all three.
5) Metadata is the descriptive information about the subject data, such as when it was mapped, alternative definitions, and a nearly infinite variety of additional information. The OMG (Object Management Group) published the CWM (common warehouse metamodel); the Metadata Coalition published the OIM (open information model) at about the same time in 2000. Neither has become an accepted standard.
6) EAI: enterprise application integration, software tools that provide "adaptors" for packaged software applications and toolkits to allow for the real-time integration between operational systems; the expansion of capabilities of EAI vendors puts them in competition with ETL vendors.
7) EII: enterprise information integration, software to provide access to disparate data sources, including data warehouses, for analytical purposes.

INTRODUCTION

Business intelligence (BI) is entering its second decade as a widely deployed process in organizations. When it was a new and rapidly expanding technology, BI spawned many fresh and intriguing ideas and vendors. For the most part, BI tools were sold to functional areas within organizations, especially finance and marketing, but were not part of an overall information technology (IT) portfolio. In the earlier stages of adoption, these tools were applied to departmental purposes and were tailored to one type of application or another, but always in isolation. As a result, BI practices developed that favored local (departmental/functional) decision making about tools and were designed to satisfy reporting and analytical needs to the exclusion of reuse, collaboration, and standardization across the enterprise.

Data warehousing, on the other hand, entered the collective consciousness as an enterprise concept (even though in practice data warehouses rarely were that comprehensive). Though linked, data warehousing and BI tools and methodologies spun into separate universes with their own vendors, best practices, and architectures. In time, tools for extraction, transformation, and loading (ETL) of data, as well as specialized tools for managing data quality and metadata, emerged as well. The simple departmental BI solution quickly was engulfed in a much larger enterprise-scale effort involving complex tools and designs that surpassed the departments' abilities to build and manage it. In just the past few years, management has begun to demand that the businesses run in a much more integrated way, with information flowing between systems, people, and processes in a seamless manner. Real enterprise BI environments have expanded significantly to include portals, the delivery of information and alerts through e-mail, wireless devices, the emergence of knowledge management, enterprise application integration (EAI) and enterprise information integration (EII) tools, and even a blurring of the line between operational and analytical processing.

The resultant landscape is one of many tools with overlapping functionality, but often lacking any concept of a unified whole, an enterprise BI architecture. Instead, choices about tools and approaches are predominately made at a component level. Which relational database to use? Which ETL tool to employ? Which BI tool or tools to give to "users" for reporting and analysis? All of these pieces are stitched together into a whole that is often less than the sum of the parts. The name for this approach, used optimistically, is "best of breed."

In "best of breed" architecture, tools and vendors are selected based on their fit to the particular situation and cobbled together to form an operating unit. The decisions made about each component are rarely made from a systematic, enterprise-wide perspective, though. Instead, each component is judged on its immediate merits without regard for the longer-term operation of the whole initiative.

> **The presumption underlying this approach is that if the "best" choice is made for each piece, the resultant system will be optimal. In many cases, decisions about compatibility between components are considered, but only between some.**

Does this version of our "standard" database support the latest clustering technology of our favored hardware vendor (without considering if it supports the "must have" features of the ETL tool)? Does this BI tool create "cubes," separate physical structures that support OLAP queries, when the database is scoped and scaled for large-volume direct SQL queries? These are typical oversights, but a far more dangerous one is failing to consider that agile organizations are beginning to see BI as an essential enterprise asset. Many of the "operational" processes, such as those supported by ERP, CRM, or SCM,[8] for example, cannot perform without being informed by analysis provided by BI. There isn't time for a small team of analysts to pull data from a data warehouse and draw conclusions when a customer service problem arises or a district manager has to assemble a pricing approach for 500 locations each morning. Unless the BI environment is engineered to step up to enterprise caliber, cooperating with all of the other assets in the portfolio, it will never generate an adequate return on investment (ROI). In fact, without that capability, the return on investment of the other assets in the portfolio will be degraded.

Typically when organizations make best-of-breed decisions it is because they are trying to reduce the initial acquisitions costs. One major fault with ROI analysis is that it is often misapplied to justify these piecemeal approaches without considering the lifetime consequences (another variant of ROI analysis is TCO, total cost of ownership, which is designed to take these consequences into account).

> **When a BI system is expected to cooperate with other operational systems in an organization, the relentless effort of keeping it compliant with the constant changes can be overwhelming if the integration isn't tight.**

What's more, because best-of-breed implementations are one single system in name only, each individual component must be kept up to date. Though it may appear to be a more economical choice at the outset, best-of-breed approaches that endure for a number of years can rack up significantly greater ongoing costs than a single architecture. It is typical today to see the integration of four or five or even six or more technologies to deliver data warehousing and BI including:

- Extraction, transformation, and load (ETL) software
- Data profiling, scrubbing, de-duping tools
- Metadata repository
- Database design/CASE
- Relational database management systems
- Reporting software
- OLAP
- Packaged analytics (BPM, financial, line-of-business verticals)
- Data mining
- Portal software

It is also typical to see organizations deploy different products to provide similar, overlapping functionality, particularly the BI tools for reporting and OLAP, and also the relational databases. And finally, as the functional breadth of the products becomes ever wider, the overlap becomes even more pronounced. This is especially true with the relational databases, which have all started to encroach on the ETL and BI vendors. And the BI vendors themselves have peeked out of their foxholes and expanded their offerings as well.

EMERGENT PROPERTIES IN BEST OF BREED

At some point, you simply have to ask yourself, "Is there a better way?" It is a well-known fact in engineering that local optimization of variables can not only lead to a poor overall solution, it can lead to disaster. Consider the design of a pickup truck. Suppose there are only two variables to optimize, capacity of the fuel tank and shape of the bed. First the bed is designed, but it leaves no room for a fuel tank. Next, the fuel tank designers try to select the best location for a fuel tank given the shape of the truck as it is. As it turns out, the best decision is to locate two fuel tanks, one on each side of the truck, just below the doors. Now that both problems are apparently solved, the overall design unit, the truck, has everyone's favorite bed and everyone's favorite fuel-tank capacity. The only problem is that the truck now exhibits an unfortunate characteristic – in a sideways collision, the truck explodes on impact.

---

8) Commonly used acronyms for integrated software to support major operational processes; ERP – enterprise resource planning; CRM – customer relationship management; SCM – supply chain management

The term for this phenomenon is *emergent property*. An emergent property of something is an attribute that "emerges" from the whole, a higher-level characteristic that summarizes lower-level things. In our example of truck design, the solution to the bed versus gas tank challenge is neither to pick the best bed design nor the ideal tank location; rather the goal is to optimize the overall truck design where the bed and gas tanks are optimized as part of the overall whole. In understanding best-of-breed versus single-architecture decisions, emergent properties play a crucial role and must be considered in evaluating and building an enterprise business intelligence strategy.

> **This is the drawback with best-of-breed designs – unless the designers are meticulous, and the design always considers the entire finished product, local optimizations (decisions) often lead to dysfunctional composites.**

In data warehousing, meticulous design and attention to the final, integrated product are not very prevalent. Part of the reason for this is that the designers and developers of data warehouses are just learning, they have not been through a project before. Another reason is that the data warehousing industry is stuck with a handful of methodologies that are entirely data centric and focus almost exclusively on the database and ETL aspects of enterprise BI and provide no guidance or urgency for the need to create a unified architecture for all of the parts. Expertise is fractured and most data warehouse practitioners are specialists in one area or another, and there is a lack of qualified architects who are capable of directing a truly unified effort. As a result, few data warehouse/BI projects are driven from a set of principles and understandings that lead to a single-architecture implementation.

SINGLE ARCHITECTURE

Alternatively, SAP offers a single architecture for data warehousing and BI as part of its overall offering that is engineered to cooperate with the entire enterprise suite of applications. That SAP® Business Information Warehouse (SAP BW) is completely integrated with the entire mySAP™ Business Suite family of business solutions is well known. What is more important, however, is how the BI aspect of SAP BW is completely integrated into the wider SAP NetWeaver™ platform's architecture, which can enable BI to grow beyond an ETL-DW-OLAP architecture into a truly cooperative enterprise BI environment. Functioning at that level of integration can ensure overall value and cost reduction. Any true ROI calculation for BI (including data warehousing) must take into account how the ultimate beneficiaries of the effort, the knowledge workers and, even, the unattended business processes in an organization are served over the lifetime of the project. This is an area that bears closer scrutiny. Data warehousing began as a data-centric business, and it is no mystery why the data and database aspects of it seem to occupy center stage. The work that people do with information has rarely been afforded the respect it is due by IT, but how BI gets used in an organization is key to realizing bona fide, sustainable ROI. BI tools may provide many different flavors of analytical processing, but the true measure of their worth in an organization is how well they facilitate "closing the loop" through collaboration, connection with workflow, and overall integration into the business processes they inform. How well "best-of-breed" approaches can compete with a single architecture is a significant part of the calculation.

The subject of emergent properties was discussed in the example of the pickup truck. From a design perspective, it is desirable for emergent properties to be positive, not negative. For example, both chlorine and sodium are quite poisonous to humans if eaten or even breathed, but when they combine to form a sodium chloride molecule, they are completely harmless.[9] In fact, the combined molecule is essential – salt. How exactly all of the different components in a best-of-breed implementation behave, the emergent properties of the assembled whole, is too complex to know in advance. Generally, it takes a great deal of effort to perform testing, both at the beginning and as things evolve, to understand system behavior. In a comprehensive suite architecture like SAP, that responsibility for emergent properties identification and management is handled before the package is opened, so to speak. The software vendor that supplied that set of tools and functions has already created, tested, and perfected the software, alleviating most if not all of the effort. In addition, the interplay between features that adds value to a system is often not immediately obvious, and only through the use of the architecture over time are the positives separated from the negatives. One could argue that this favors a single architecture over best of breed, in theory, but it's always best to have some facts and metrics to apply.

The subject of this paper is to examine the relative merits of these approaches through the use of ROI analysis.

How exactly one measures benefits and costs is the subject of much discussion. Most data warehouse ROI analysis is focused on the cost of acquisition, integration, and storage of data and some rather loose estimates of benefits. The integration of data warehousing and BI with the rest of the IT infrastructure is rarely addressed because of the pervasive "best-of-breed" bias. This cannot be stressed too much. Many data warehousing/BI initiatives self-evaluate based on a disconnected concept of provisioning data and offering tools to view it. While this was initially a giant leap forward, it is no longer adequate for promoting business agility. The models of costs and benefits typically overlook the most crucial aspect – getting connected with everything else in the organization. For that reason, many publicized data warehouse ROI studies, and especially most published BI ROI methodologies, are suspect. In this paper, considerations for evaluating ROI of the entire data warehouse and BI effort, including the implications of the broader organizational IT architecture, are proposed.

9) In fact, an even more extreme example of emergent properties is the combination of two noxious compounds, sodium hydroxide and hydrochloric acid, which, when combined, form saltwater.

ROI OVERVIEW

ROI is simply the ratio of benefits to costs. The mathematics are simple, but how those benefits and costs are recognized and accounted for is much more complex. The values always occur over time, so the use of discounting is an issue. How to represent the boundary between favorable and unfavorable returns, 0% or 100%, is also an issue. In its simplest form, ROI is just a ratio:

Equation 1: Basic ROI Calculation
ROI = (benefits realized)/(cost to deliver) x 100

For example, suppose an organization is able to reduce out-of-stocks on a permanent basis by understanding demand and real inventory at a unit level and as a result, increase sales by a constant amount, leading to an improved margin of $10,000,000 per year. If the cost to deploy the system was $5 million, the ROI is 200%: ROI = ($10 million)/($5 million) x 100. ROI measurement always accounts for a period of time. In other words, the two-year, three-year, and five-year ROI figures will vary. The reason for this is that both benefits and costs are a series of cash flows. Like many large investments, BI investments tend to pay off over a period of time, not all at once, so it makes sense to look far enough into the future to capture the string of benefits that occur after some or all of the spending (investment) is made. A project may show a negative ROI in the first 12 months, turning positive in the second year, and perhaps showing very good or even dramatic returns in the third year and beyond. The problem, of course, is that the farther we peer into the future, the less credibility we can attach to the projected benefits and even, to a certain extent, the costs.

Because ROI calculations always involve time series, it is common to take into account the "time value of money," an exercise known as DCF for discounted cash flow. One simply calculates the NPV (net present value) of each cash flow, numerator, and denominator, and takes the ratio of those sums to calculate the ROI. For investments like BI, where the projected benefits are often not certain, the precision of DCF is not really needed. Interest rates are at historic lows anyway (though not in every country), the number of years under consideration (typically two to five) is not sufficiently large enough for discounting to have much effect. Besides, the returns on investments like BI should be great enough that the precision afforded by DCF may not be needed for such short time frames.

Equation 2: ROI/DCF Analysis
ROI = NPV(B_1, B_2, B_3, . . .)/NPV(C_1, C_2, C_3, . . .) x 100

In this example, the series $\{B_1, B_2, B_3\}$ represents the value of benefits in year1, year2, and year3, and $\{C_1, C_2, C_3\}$ represents costs for year1, year2 and year3. Extrapolating from the example in Equation 1, suppose that the costs to deploy the system in the first three years were $3 million, $1 million, and $1 million respectively. The gain from increased sales occurred only in the second and third years in the amounts of $4 million and $6 million. If we were not using a NPV calculation, the ROI would still be 200%, $10 million in benefits and $5 million in costs. In this case, the bulk of benefits accrue after year2 and the bulk of costs are in year1. DCF analysis will yield a lower three-year ROI than 200% as the gains are discounted more than the costs.

Using a 10% discount rate, the calculation becomes
ROI = $4,305,034/$ 7,813,674 x 100 = ~182%

Consider a BI investment that is meant to understand the actual behavior of a customer instead of a proxy, such as prescriptions written by a doctor.[10] If a pharmaceutical company were to track prescriptions only to the point where a doctor writes them, they can measure and perhaps influence the behavior of the doctor. But once the patient leaves the doctor's office, the rest of the interactions are lost. Where does the prescription get filled? Does it get filled? Under what scheme? Is it refilled? How often? How many times? Without being very precise, assume that being able to understand this downstream behavior has value to the company. For instance, suppose 2% of the doctors account for 30% of the unfilled prescriptions and that this costs the company $5 million per year. Further, assume that the cost to rectify the situation is an ongoing $100,000 per year and results in a positive gain of half the potential, or $2.5 million per year starting in year2. In this sense, we can say that the BI effort showed a net benefit of $2.4 million per year. If the total investment to build and implement the data warehouse and BI system was $5 million, we can say that the 2-year, 3-year, and 5-year ROI is 49%, 96%, and 182%.

Equation 3: ROI at Points in Time

$$ROI_{Yr2} = (0 + 2.5m)/(5m + 100{,}000) \times 100 = \sim 49\%$$

$$ROI_{Yr3} = (0 + 2.5m + 2.5m)/(5m + 100{,}000 + 100{,}000) \times 100 = \sim 96\%$$

$$ROI_{Yr5} = (0 + 2.5m \times 4)/(5m + 100{,}000 \times 5) \times 100 = \sim 182\%$$

A few things are worth noting in the above calculation. First, for simplicity, it did not take into account the ongoing costs of sustaining the investment. It did net out the cost of acquiring the benefit, but a more thorough ROI calculation should consider the ongoing costs. Secondly, since the increased revenue recurs each year, the ROI continues to increase. This has the effect of showing the investment to be an increasingly better performer each year, which is one of the drawbacks of ROI analysis. In point of fact, the real value derived happened in the first year, when the root cause was discovered and remedial action initiated. One could argue that recurring benefits should accrue to the areas that effect the changes. It is often difficult to evaluate and assign credit for gains in an organization. For these reasons and many more, ROI is a technique that is highly nuanced.

ROI has a tendency to be fairly optimistic about what will happen beyond the first year. Like the example above, the costs of maintaining the investment each year are typically minimized. Best-of-breed BI environments, including data warehouses, have very high maintenance costs relative to the initial investment, and it is not unusual for those annual costs to run more than 50% of the development cost. In a five-year ROI calculation, that can triple or quadruple the TCO. The net result is that, when comparing alternatives that have higher and lower initial outlays, but similarly reversed ongoing costs, this initial optimism favors the lower initial-cost model because of the understated future costs. If the assumption is that the higher initial-priced option will operate with less ongoing cost, that benefit is dampened by the optimistic assumptions of the lower-priced initial option.

Another alternative is to match the cash flows and show when (or if) the benefit stream exceeds costs. This is called breakeven analysis (also referred to as payback point) and an example is shown in the figure below. Whether the calculation is based on ROI or breakeven, the important thing to remember is that the accounting for benefits and costs is the same, they are just variations on the same set of assumptions.

10) Measuring behavior is a tricky business and any metric is a proxy, but some are closer to the behavior than others. Looking out the window to see if the pond is frozen is a proxy, but checking a thermometer is a better one.

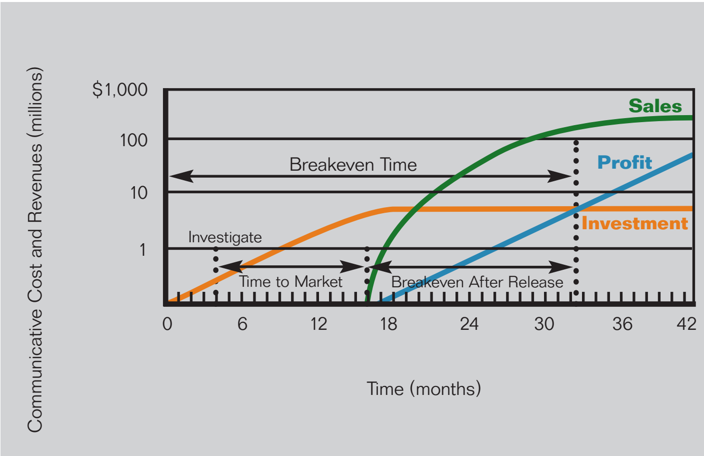

Figure 1: Breakeven Analysis

One problem with ROI analysis is that the calculation depicts the ratio of benefits to cost, which are usually positive numbers. This leads to the strange situation where a project never realizes returns in excess of the investment, yet the ROI is still positive. A project that costs $1 million and realized only $250,000 in returns can be said to have a 25% ROI. To some listeners, that might sound respectable. The solution is to alter the calculation so that a negative return is derived whenever the gains fail to exceed the investment. The formula then becomes:

Equation 4: Alternative ROI Calculation
ROI = (Benefits Realized – Cost to Deliver)/(Cost to Deliver) x 100

And from the example above, ROI after two years is now stated as:

Equation 5: Two-Year Negative ROI
$$ROI_{Yr2} = (0 + 2.5m - 5m - 100{,}000)/(5m + 100{,}000) \times 100 = \sim -51\%$$

Again, what is important is the methodology used to gather the costs and benefits, not the calculations. Another controversial issue in this kind of analysis is claiming the benefit itself. After all, the data warehouse doesn't call on doctors and rectify the unfilled prescription problem, people do. And people require support from other people and infrastructure as well as the direct costs they incur. For those reasons . . .

. . . the benefit taken for the ROI calculation is usually only a fraction of the actual benefit, in recognition that these kinds of improvements rarely happen in isolation. What percentage to use is a matter of policy, but amounts in the range of 20–50% are fairly typical.

The logic behind that decision becomes clearer with a closer examination of the costs and benefits of BI.

ACCOUNTING FOR DIRECT COSTS

There are many costs associated with a BI implementation that are obvious and easy to quantify. In addition, most organizations have fairly standard accounting practices for recognizing these costs, making comparison of IT investments over time balanced. Any item that is charged to the organization through a purchase order, such as hardware purchase or lease, software purchase (or lease, sometimes), and ongoing "maintenance" costs can be sent straight to the denominator of the ROI calculation. To the extent that these resources are used exclusively for the project, their associated costs are identifiable. Sometimes, however, the use of these components is shared with other initiatives. In other cases, the architecture might make demands on the existing infrastructure, such as networks, that are fully accounted for in other calculations. In these latter cases, the ROI is often overstated.

Because BI is increasingly playing a role in ongoing processes, especially customer relationship management (CRM) and supply chain management (SCM), the effort to integrate BI with these processes can no longer be overlooked. These costs can be substantial in a best-of-breed approach (and in a single-architecture approach, depending on how it is configured) and failure to effectively implement the interoperation can cause substantial loss of benefits and even degrade the effectiveness of the operational systems themselves.

We start by examining the direct costs by category: acquisition, infrastructure, implementation and deployment, support and lifetime, and uptime/reliability/performance.

ACQUISITION COSTS

From the very beginning, data warehousing was always isolated from the source systems that provided its data. The reasons were sound, namely, to insure that the intensive processing required to load, index, and query a very large database did not interfere with the performance of the operational systems. As a result, data warehouse projects typically involved the purchase of a separate relational database license and a server, complete with processors, memory, and storage, either integral or separate. Specialized data warehousing tools, such as ETL software and perhaps data cleansing tools were also needed. It was also typical to provide separate environments for development and testing, so multiple licenses and hardware were called for. All of these expenditures occurred in the earliest stages of the project and were clearly identifiable as costs.

Curiously, the cost for BI software was not always accounted for in the initial expenditures. The reasons were varied. In many cases, the developers of the data warehouse were more focused on the data itself than the changing of data to useful information. The assumption was that data would be extracted from the data warehouse and how it was used beyond that was not really a concern. This, of course, was a huge mistake, but one that was repeated many times and persists to this day. In other cases, it was assumed that the task of assembling the data warehouse was so imposing, that "small" decisions about things like BI software could be deferred. For many IT managers, the unspoken assumption was, "Let's see if this whole thing works before we buy more software." In retrospect, this may have been a wise decision, but it did lead to some faulty accounting about the cost of the data warehouse.

Another difficulty with assessing the acquisition costs of BI software is that many organizations tend to start slowly, acquiring a small number of licenses at first and allowing the population to grow over time. Also, the BI vendors continually introduce new products, with new license fees, that add to the ongoing costs of the data warehouse, but rarely appear in the initial investment accounting. *For this reason, the acquisition costs of BI software are often greatly understated in ROI calculations of "best-of-breed" architectures.*

INFRASTRUCTURE COSTS

Infrastructure costs relating to BI are incremental, marginal or imputed. How the costs are accounted for depends on the organization's orientation toward infrastructure costs: pure cost, efficiency, or strategic.

Incremental costs are additive and apply when additional components or capacity in infrastructure are called for when deploying BI. Examples of incremental infrastructure are network enhancements, upgrades to hardware and software that already exist as a result of requirements of the data warehouse/BI effort, or upgraded equipment for the users. These costs can be significant, especially when they push some other component beyond a threshold, requiring a greater investment, such as upgrading the server of a key operational system because of the data extract demands of the data warehouse. Sometimes they are cooperative expenses, where, for example, the BI requirements for storage are sufficient to justify a previously evaluated upgrade to a fast backup/restore system. The types of incremental infrastructure expenses are too numerous to mention because a data warehouse/BI implementation always requires integration with many existing systems and infrastructure components. By its very nature, it is a highly connected enterprise.

Marginal infrastructure expenses happen when surrounding systems are being upgraded and the data warehouse adds some fractional load on the requirements. A few years ago, it was not uncommon for data warehouses to be deployed at the same time organizations deployed e-commerce. The data warehouse architecture may have shared Web servers, increased network bandwidth, new storage and disaster recovery facilities, and so on. For example, without the e-commerce initiative, the data warehouse might have needed to fund a $500,000 storage device, but at the margin, may have only needed $200,000 of additional drives once another initiative funded the device. In this case, the marginal cost was $200,000.

Some organizations find marginal infrastructure cost to be misleading and instead apply an imputed cost, derived by a formula developed by the organization's accountants. In the above example, the e-commerce project footed the bill for the storage frame before the data warehouse added more capacity at the margin. In an imputed scenario, a portion of the initial investment (and ongoing cost) would be allocated to the data warehouse project.

Different organizations view infrastructure costs differently. Some consider them as pure costs, no different from any other expense. For those organizations, a fair ROI calculation needs to be vigilant in finding all of the costs, incremental, marginal, and imputed. Other organizations consider infrastructure a competitive advantage and are very casual about accounting for it on a project basis. Obviously, in these organizations, the costs need to be taken into consideration, but not quite as scrupulously. And finally, some organizations see infrastructure as strategic – it defines the organization. In those situations, organizations are nearly pure information companies; accounting for infrastructure costs in ROI calculations is rarely considered a priority.

How to account for infrastructure costs is a more complex issue than direct, incremental acquisition costs. Since most BI development over the years was departmental in scope, the IT area that provided the infrastructure typically absorbed infrastructure additions. When a BI initiative is enterprise in scope, as is the trend now, the infrastructure costs are usually included in the ROI calculation, most commonly using the marginal or imputed method.

IMPLEMENTATION AND DEPLOYMENT COSTS

Implementation and deployment costs are composed primarily of consulting costs, training costs, and internal labor costs. The last category, internal labor costs, is very difficult to quantify. One line of reasoning is that internal, salaried developers, designers, and managers are employed to do development and should be considered a sunk cost and their loaded labor rates should not figure into the project cost. When the labor costs are taken into consideration, there is difficulty allocating the cost because the duties of employees are rarely focused on a single task over a period of time. Whether or not to include these costs is a matter of policy typically.

Consulting costs are straightforward to quantify, though there are some complexities there too. Consulting is often bundled with the purchase of software and/or hardware and it is difficult to unbundle it for ROI purposes. Services are also priced on a fixed-cost basis and the scope of the project may be greater than just the data warehouse. Lastly, consulting services in the development of a data warehouse may be provided as part of a wider outsourcing contract, whose terms can be quite complex over a period of years.

Training costs are those that are charged by third parties, such as vendors or other training resources, or may also include internal costs such as travel or chargebacks for the use of shared resources like internal training facilities. Again, whether or not to include the imputed costs of employees preparing for and conducting training is a matter of taste. An even larger cost component is the time that users spend in training and whether this needs to be accounted for. In an extreme example, suppose that one BI deployment was so complex and foreign to the users that it took three times longer to train people, and further suppose that the training would ultimately reach 1,000 people. Even if the additional training were only three days instead of one, the additional cost to the organization would be 2,000 person days, or roughly \$1 million to \$1.5 million. Excluding this cost element from the ROI calculation would be a mistake.

> **One innovative client actually evaluated the quality of the training, and only charged the project for what it deemed "wasted time," which in that case was substantial.**

Training is difficult and not always effective, so evaluating its cost with respect to what it delivers is a sound approach.

SUPPORT AND LIFETIME COSTS

In its current state, BI is such a broad category that generalizing about support costs and the expense of maintaining it over time is very difficult. In addition, the concept of enterprise BI is relatively new and most organizations are supporting only one or more departments or applications, not the entire enterprise. For those reasons, actual support costs may be low but the returns are probably equally low. In addition, without an enterprise approach, as mentioned before, it is very likely that there are multiple versions of pieces in the technology stack, which causes costs to be out of proportion, especially when there is replacement of one or more pieces in the stack, which is also very common. In summary, in patchwork BI environments, direct support and lifetime cost projections for ROI purposes tend to be low because BI is not really performing and also because BI support costs are often hidden in internal labor costs (a concept discussed more fully, Shadow IT, in a following section). In some cases, the costs escalate because of churn or just the sheer numbers of different software packages to support, even when the BI performance is still low. Needless to say, the ROI in these environments is extremely poor.

The support costs for the various packages are significant and generally not recognized as such because they are budgeted and funded piecemeal. When evaluated as a whole, the costs of cross-training in multiple products, both the direct costs and the slower ramp-up time as new employees move into positions, can be substantial. The lost opportunity costs, as a result of lower levels of expertise, shortages of trained staff, and extra lead time to integrate tools to perform adequately, all add up to higher carrying costs and dilution of benefits. With a myriad of software tools laced together, patch and upgrade cycles are guaranteed to overlap, causing an excessive amount of installation and testing time and possibly downtime. There is also another subtle drawback that is rarely recognized until the moment it is critical – . . .

. . . each product carries a checklist of features, and when buying decisions are made, the assumption is that those features will actually work. Unfortunately, not every vendor insures that their product can interoperate with every other vendor's. The feature that is needed may not work in your particular configuration.

Alternatively, if we look forward to an expanding deployment of enterprise BI, the ROI elements look very different. For one thing, the technology is simplified and stabilized, reducing duplication of support and maintenance. Training costs should increase dramatically, but the returns should increase to an even greater extent. Now, instead of a handful of isolated analysts, the entire enterprise is informed with critical, useful, and actionable information. The data warehouse/BI environment operates as a central hub of historical, analytical, and organizational reference information, as much a net consumer of information as it is a provider. Support costs in actual amounts will increase, but will decrease substantially in relation to the benefits, a net increase in ROI. Lifetime costs are dependent on the approach chosen, best-of-breed or single architecture. A best-of-breed architecture that actually functions and provides enterprise BI will have similar ongoing interface and maintenance costs to a single architecture because, by definition, it was designed well enough. In current practice though, this is the exception – not the rule – and best-of-breed running costs are significantly higher.

Consider a hypothetical case where a five-year ROI for a best-of-breed proposal is set at close to 200% (remember, this means the benefits are projected to be twice the investment). The simplified model and assumptions are detailed in the Appendix but the results are displayed in Figure 2 (dotted blue line). Because these analyses are typically fairly optimistic about both the spend and the outcome, the actual ROI, depicted in red, is much lower. So much lower, in fact, it never crosses the payback threshold of 100%. Component choices that lead to complications, overlooked requirements, such as the need for a data reengineering tool, a second BI tool is purchased in year3 because the original choice is unable to integrate fully with the chosen portal tool and finally, lost benefits due to delay and integration problems, cause a dramatic reversal in the ROI. As new requirements over a three- to five-year period arise, the delays, incompatibilities, and lack of flexibility all cause the project to miss many of the deliverables. In the third example, a single architecture requiring no integration of separate tools or integration into the existing corporate infrastructure is purchased. The initial outlays are higher, the overall ROI is far superior as the costs in ensuing years are much lower, the project is delivered on time and captures all of the planned benefits.

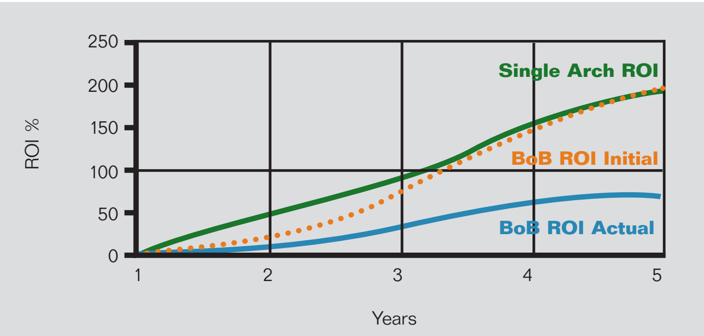

Figure 2: ROI Comparison: Best-of-Breed, Initial and Actual, Versus Single Architecture

Keep in mind that these are just illustrations. Neither the components nor the values are meant to depict an actual project; they are used to broadly illustrate the variations in ROI models and how they are interpreted. Not every best-of-breed implementation runs into the same problems and not every single architecture implementation proceeds without difficulty. In the general case, single architecture implementations are characterized by higher initial outlays and lower maintenance costs as well as lower failure rates. Best-of-breed implementations, even when fully successful, consume greater resources adapting themselves down the line to new requirements, such as the following examples in the past few years that posed major problems for BI implementations:

- **E-business:** The batch nature of data warehouses limited the ability of organizations to quickly apply them to the real-time aspect of doing business electronically
- **Portals:** Pulling the BI function under the portal umbrella required a great deal of retooling and handwork
- **Knowledge management:** BI and KM live in two separate worlds and as stand-alone offerings are incompatible

In conclusion, using ROI to instantly compare best-of-breed and single-architecture approaches can be very misleading. The relatively higher initial spend in single-architecture offerings has to be compared against the lifetime costs of alternatives, including the functioning of the finished whole as part of the organization's portfolio of interoperating technologies.

UPTIME, RELIABILITY, AND PERFORMANCE COSTS

Though not always considered a cost, the premise behind a BI environment is that it is available when it is supposed to be, that its operation is predictable, and that it performs well. These factors ultimately drive the successful capture on the intended benefits assumed by the investment. If the ultimate users of this investment are not satisfied with the service, they will find alternatives and the viability of the initiative will rapidly spiral downward. These three factors each affect ROI directly:

Uptime: Multiple layers of extraction and loading, through various schema and different software packages, create latency and excessive downtime. A well-designed system, with a single metadata schema for import and export, driving the transparent operation of physical structures is less likely to be susceptible to incidents that keep the system out of operation. Measuring the cost of downtime is very subjective, but the loss of uptime can often be related directly to lost benefits, which can be applied to the calculation as either a negative benefit or a an additional cost.

Reliability: Reliability affects uptime, buts its causes are events not process. No complex system is immune to failure, but the frequency and severity of it is measurable and can be compared against benchmarks. Most failures in BI occur at the junction of two processes or the interface between two systems. By definition, a best-of-breed architecture will have many more potential points of failure and diagnosing and resolving failures is much more difficult than in a single architecture if, for no other reason, vendors tend to blame the other vendors in the stack. In a single architecture, there is at least "one neck to choke." Failure most often occurs at the weakest link, which is clearly identifiable and should be factored into ROI numbers. If the server is underpowered, the data model fragile, the development of ETL scripts haphazard, or security model incomplete, problems will surely occur in those spots. Failure is also the result of a complex interaction of variables that isn't predictable or even repeatable. These are generally accepted as unavoidable and it is only the frequency of them that is relevant in ROI.

Performance: Performance tuning is an unending process in best-of-breed environments because the behaviors of the individual components are neither explicitly understood, nor easily interfaced or consistent over time. This is especially true with the ETL tools and the databases. In additional, the interaction between the BI tools (and this can include everything from ad hoc query, reporting, OLAP, data mining, BAM,[11] statistical tools, verticals, and so on.) and the data warehouse is in a constant state of churn due to staggered version releases, changing patterns of use, and even the time of the year or the month. This requires lots of tinkering with database optimization, prebuilt cubes, and data marts and aggregate strategy. In extreme cases, it leads to restricted access to the real detailed data in the data warehouse, a situation that erodes ROI by limiting the usefulness of the whole effort.

In calculating ROI, one should consider the added costs of repeatedly investigating and solving these problems, the added costs of workers finding alternatives as they lose faith in the system, and apply as a cost the benefits lost as a result of the subpar performance of the application.

11) BAM: business activity monitoring, the process of monitoring, either directly or by software "robots," the flow of different kinds of business telemetry in real time, such as inventory, shop floor or even financial. BAM is characterized by integrated, not raw, data and business process definitions, not typical analytical categories.

ACCOUNTING FOR INDIRECT COSTS

Organizations typically have a reliable process for recognizing and accounting for expenses that are actually incurred, but the recognition of indirect contributions to costs is far less precise or uniform. In BI, these indirect costs can occur in a number of ways.

> **A large but insidious cost is the time and expense of non-IT departments performing IT work.**

This common situation is usually referred to as "shadow IT".[12] Another indirect driver of costs is the result of incompatibilities between the BI architecture and other systems or processes in the organization. Compatibility costs are difficult to quantify, but they are serious. A best-of-breed architecture has to be engineered to be compatible with the rest of the enterprise, and it has to be maintained to stay that way. Single architectures tend to come with these capabilities built in. Finally, lost opportunity costs occur as a result of the manner in which the system functions. Typically, these are "failure-to-thrive" situations where the BI deployment just doesn't catch on. The promised benefits fail to materialize. No ROI calculation is balanced unless these indirect costs are addressed. They are particularly challenging to best-of-breed implementations, somewhat less so in the single-architecture deployments.

SHADOW IT

There is an inherent gap between the view taken by IT professionals about how people should interact with systems, and how non-IT people actually go about doing their work. In many cases, IT makes simplifying assumptions about the work in order to reduce the complexity of their job. When the product of this effort meets the intended audience, the gaps become manifest. The response is generally to supplement or even circumvent the

12) Shadow IT is a term that has been in the literature for many years. For a current look at its meaning and implications, see "Shining the Light On Shadow Staff: Booz Allen Hamilton" in CIO Online http://www2.cio.com/consultant/report2085.html

delivered functionality by developing departmental or even personal solutions to cover the unmet needs. When calculating ROI for the project, the expense buckets are filled with only the reduced effort to deliver a partial solution, but the benefits claimed include those gotten at the expense of the non-IT developers, the so-called . . .

> **. . . shadow IT. It's like a one–two punch, because it overstates the ROI of the IT effort, and it overstates the costs of the departments who have implemented their own solutions to cover the gap.**

The cognitive gap between IT people and other workers in an organization is not going to go away anytime soon. It can be moderated with careful management, but it presents a real problem when comparing ROI of best-of-breed solutions, where the risks of the gap are highest, to single-architecture implementations like SAP NetWeaver, where the risks are considerably lower.

One way to understand the extent of shadow IT is to survey people who use BI and ask them how much time they spent managing data and programs (such as managing files and spreadsheets) before the BI deployment and after. If the amount hasn't dropped appreciably, there is a shadow IT problem.

COMPATIBILITY COSTS

The technology adoption curve described by Geoffrey Moore in *Crossing the Chasm* segments technology markets into six standard deviations of a normal distribution and applies names to each group like early adopters, visionaries, late adopters, and so forth. The concept of enterprise BI with its attendant data warehouse architecture is still clearly on the left side of the curve, past the visionary stage but solidly in the early adopter segment. This implies that a significant portion of the data warehouse population (if you follow the normal distribution precisely, it would be more than 90%) has still not taken an enterprise approach. Instead, they are operating under the assumption that a data warehouse is a data utility that is designed to provide data to people, a sort of reference library. To the extent that there is a need expressed for something more specific, a new development cycle begins to either map more data or create data marts or both. The diagrams for the "architectures" are always in landscape mode with all of the arrows moving from left to right, source systems on the left, icons for PCs on the right.

In that sort of environment, there is no concept of BI operating as a hub of information; it is rather like plumbing, where the water (or the opposite) all flows in one direction. Designed with this set of assumptions, it is nearly impossible to hook up BI to the ever-growing inventory of online and real-time processes that other business processes demand unless, just like in your house, common standards are in place, common infrastructure, common plumbing. In that same way that a bidder at an auction cannot make good decisions without a good memory, these newer business processes cannot operate without the kind of analytics provided by a good BI system. In best-of-breed implementations, enabling this connection after the fact is a very costly process, very tricky, and one that is likely to suffer delays due to complexity if not outright failure – just imagine trying to route plumbing once the overall structure has been built and the finishing touches added. What's more, it is a never-ending job and an architecture not designed to participate as a peer with other enterprise systems will be in a perpetual race with the landscape just to keep up, the so-called "Red Queen Effect." [13]

13) "The most curious part of the thing was, that the trees and the other things round them never changed their places at all; however fast they went, they never seemed to pass anything. 'I wonder if all the things move along with us?' thought poor puzzled Alice. And the Queen seemed to guess her thoughts, for she cried, 'Faster! Don't try to talk!'" – Lewis Carroll, *Through the Looking Glass*

Very few data warehouses that have been purpose-built are cognizant of the heterogeneous world in which they exist.

SAP BW is an integral part of the SAP NetWeaver architecture and is an example of a single-architecture data warehouse and BI environment that connects with not only the totality of the SAP software offering, but is designed to easily connect with external processes as well. To the extent that the surrounding IT environment is SAP software, the ease of implementation and maintenance is peerless. Is it possible to build your own environment that works as well if not better in this regard? It is conceivable, but it would be a massive effort. Are there trade-offs in implementing BW instead of best of breed? Of course, any custom-built application has the potential to provide a more exact set of functions than a generic one, if it is done properly, but the effort to design and implement compatibility and connection over the long term is huge, perhaps too huge for most organizations to bear.

It is worth noting that the single architecture, like SAP BW, has to be considered within the context of usefulness, too, not just architecture. Even if it presents an unassailable case for being a better architectural fit, it still has to meet the requirements. A unified architecture is a superior solution provided:

- It all works
- It achieves required objectives
- It is open – pieces are customizable and interchangeable with third-party offerings
- It is cost effective
- It has rich business content
- It has extensive business connectors

The prevailing motivation for best-of-breed decisions is the sense that no one product can meet all of the needs or, even worse, an overlooked requirement can spell disaster later on. Practice has shown, though, that the potential for an overlooked feature pales in comparison to the potential costs of integration.

LOST OPPORTUNITY COSTS

Many of the projected benefits of BI are improvements at the operational level as a result of a more thorough and timely flow of information. A less concrete supposition is that BI will "improve the decision-making process," though there is little evidence that this is a measurable process. There are case studies, of varying levels of credibility, of organizations realizing huge gains by uncovering some massively wasteful process, or by achieving astronomical gains by some clever way of motivating customers or suppliers, but subsequent examination of the financials fails to reveal any onetime or ongoing improvements. In practice, BI is most useful in informing people and, increasingly of late, unattended processes, in ways the operational systems simply cannot. BI presents analysis in an integrated and historical perspective.

There is only one way that can happen effectively, and that is when BI is implemented and deployed on an enterprise basis. Patchwork BI drains ROI by limiting the propagation of the tools in the enterprise. The three cases below illustrate this phenomenon.

Case 1 – Poor Uptake of BI and Limited Use of Tools

Failing to achieve the level of benefits possible is an ROI killer, though one that is not often recognized. If one were to build a bridge, at great expense, that represented an engineering breakthrough but failed to attract traffic because of misplacement, it would be considered a waste, an obvious and visible one. But in data warehousing, when a similarly well-built data warehouse is put into production but the level of BI use never grows, the lost opportunity is rarely obvious or visible. This is because . . .

. . . BI requires changes in the way people and organizations work, . . .

. . . unlike the bridge example, where traffic is still traffic. No one notices that something new hasn't happened. Nevertheless, it's a loss just the same.

Organizations that succeed with BI understand that there is a need to experiment and to modify elements that surround BI, such as other systems, work processes, and supplier and customer relationship handling, to name a few. BI exhibits something like the network effort, its value increasing geometrically with the number of connections. This argues very strongly for an enterprise approach to BI, and that implies that the ability of the entire environment to connect and stay connected to everything else is paramount.

BI has to make sense to people before they will invest time in it. It also requires a change in work processes, reporting relationships, and localized power bases. Data czars[14] draw their influence from the data they "own" and control. BI tends to break up these artificial relationships (and rightly so), but it doesn't happen automatically.

For BI to be successful, it has to be campaigned, it has to be made relevant, and it takes much more effort than most organizations have expended to date.

People, even with the best intentions, have a tendency to resist new things, especially at work. Uptake of BI is dependent on people connecting, one on one, to see the value. When the process is disjointed, not connected to other things they do, and presented in language and metaphor that isn't understood, the uptake is very low, initially and over time. When BI is presented as another aspect of a set of coherent processes, the success rate is dramatically higher. BI is no different than a new product introduction – unless certain volumes are reached, the costs of development and introduction overwhelm the benefits.

Case 2 – Attrition

Another drain on enterprise BI ROI is the diminution of activity and participants over time. Usage profiles, even of very successful cases, tend to increase rapidly at the beginning, fall off sharply but only partially, and then stabilize in a steady upward trend. There are many explanations for this phenomenon, but the most likely one is that some people stick with it because they see the value, others don't or won't after trying it, but if it works, it propagates over time.

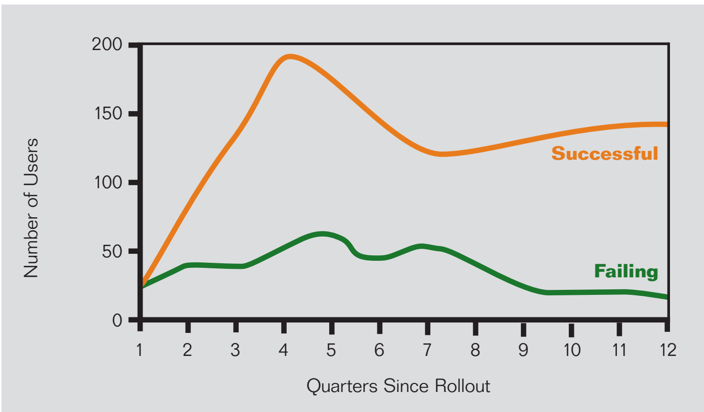

Figure 3: BI Adoption Curves

14) Data czar is a term used to describe individuals in organizations who control the flow of data by operating as the single point of contact for some integrated, transformed, modeled, or otherwise value-added data, often to the detriment of the organization and engineered in such a way as to provide themselves influence, control, esteem, or job security.

When there is attrition beyond the normal curve, many factors are at work. The initial enthusiasm swells the number of participants, but problems with availability, reliability, and performance degrade the impression of the system and sooner or later, people regress. Very often, BI systems become victims of their own success – as use increases, the stress on the system begins to show and brittle architectures weaken.

Integration of six technologies is a tricky business and scaling up the activity level is bound to reveal flaws.

But it is not just scaling that causes the architectures to degrade, success in BI breeds more and more demanding applications, as it should, and as the process matures, thresholds are reached and breached. In a single architecture, there is no guarantee that this won't occur, but it is less likely and far more certain to be addressed – in-house IT staff are stretched very thin compared to a well-heeled software vendor with enterprise-caliber products.

Case 3 – BI Functioning Well, but at Much Higher Carrying Cost Than Industry Norms

Resourceful people can get results from situations that are not in the best condition. BI in organizations, because of its origins as a departmental solution, may be delivered through more than one BI product. In fact, it is very common to find two or three different BI tools in use, all with overlapping functionality. This situation has an impact from the top to the bottom of the data warehouse technology stack because each tool has its own idiosyncrasies and places its own demands on the data warehouse structure and operation. One tool may generate SQL dynamically, and to operate efficiently, requires a physical model that conforms to certain conventions. Another tool may operate strictly in a reporting engine and requires a very different schema and indexing approach to provide acceptable performance. Yet another tool offloads large amounts of data into its own flat files for serial processing. The manner in which all of these schema are populated and maintained adds additional effort to the ETL process.

In a situation like this, each of the constituent clients may be quite happy with the way the BI process is working, but the cost to the organization is substantial. It is unlikely, however, that this approach will be able to keep up forever. As new metadata management approaches mature, patchwork environments will not be able to implement the advanced capabilities that are just around the corner, and there is increasing pressure from organizations to reign in the costs of data warehouses and BI based on publicized best practices.

	Best of Breed	Single Architecture
Acquisition Costs	Initial outlays may be lower, can rise significantly over time	Initial outlays can be higher, more controllable
Infrastructure Costs	Incremental	Marginal
Implementation and Deployment Costs	Considerable, especially integration efforts and patchwork BI over time	Significant, but controllable and predictable
Support and Lifetime Costs	Uneven releases of upgrades multiplies support costs	Upgrades in sync with broader systems, though license fees and maintenance still significant
Uptime, Reliability, and Performance Costs	Complex to diagnose and resolve due to the number of parts	Single point of contact and responsibility
Shadow IT	Completely dependent on the quality of the effort	Consistent UI and integration with operational process will tend to reduce it over time
Compatibility Costs	Potentially very high	Likely very low
Lost Opportunity Costs	Likely due to process	Somewhat less likely, but the same forces are in play

Table 1: Summary of Cost Items

EVALUATING BENEFITS

The connection between information technology investment and business value is rarely straightforward. Like our earlier example of the bridge, counting cars is simple and is a good proxy for measuring the amount of traffic it serves. In organizations, the positive effect of technology is neither straightforward nor well understood because it often involves the interaction of a series of highly complex operators – people. Weill and Broadbent[15] conclude, "Measuring the impact of an information technology investment will be much easier at the bottom of the hierarchy than at the top, where many factors dilute the investment . . . There must be other factors, apart from the size of the investment, influencing the relationship between information technology investment and business value. Two such factors are the balance between different parts of the information technology portfolio and the ability of the firm to convert these investments into business value."

The concept of dilution raised by Weill and Broadbent is rarely discussed in ROI calculations. The assumption is that if this technology implementation supports the realization of certain business values, those gains flow all the way to the bottom line. It's implicit in the ROI calculation itself. For example, suppose that a BI effort makes it possible to segment and understand customer behavior at a level not possible before, fostering much more precise pricing decisions that are projected to provide a multimillion dollar benefit to the bottom line. Even as those decisions are made, other processes and people have to fall into step for it to happen. Suppose the sales force either can't get the information fast enough or resists the whole approach for some reason. What if sales management isn't convinced the process will work and obscures and obfuscates to the point where very few of the pricing decisions are implemented?

Another form of dilution is when the pricing decisions are put in place and work, the benefits are certain and countable, but the new surplus gets drained by creeping inefficiency two steps up the chain.

What Weill and Broadbent allude to is that the investments may be laden with potential, but unless the organization knows how to and is willing to leverage the investments with all of the other elements of the technology investment portfolio, the gains may be short lived, like so many subatomic particles that wink into existence and disappear just as quickly. *This argues very strongly against delivering BI as a stand-alone effort.* The original concept of data warehousing, to provide a reservoir of data from many incompatible systems, is not holding up. The collection of systems in today's organization is more tightly integrated and balanced than it was 20 years ago and a data warehouse can no longer stand alone to support a handful of report writers. BI is part of the enterprise portfolio and it has to be managed that way. When looking for benefits from BI today, it is impossible to separate BI from everything else. The days when a company could build a data warehouse for $5 million and claim a 30-day payback when they discovered $50 million in billing errors are over, if they ever occurred in the first place.

The benefits that are possible through the careful implementation of enterprise BI are not only too numerous to mention, many of them are yet to be discovered.

Informing business decisions at the right moment has an almost incalculable positive effect on an organization, but it requires more than just information.

---

15) Peter Weill and Marianne Broadbent, *Leveraging the New Infrastructure*, (Boston, Harvard Business School Press, 1998), 50-51

The delivery and presentation has to be suited for the intended use. Many knowledge workers lose their enthusiasm for BI when they realize that in order to weave the results into the work they perform, they still have to resort to manual tools like spreadsheets. The disjointed, end-of-the-line nature of BI is inadequate for the role it needs to perform in the enterprise. Some broad categories where substantial benefits can be found in almost any organization are:

1. **Timeliness:** Timing involves more than responsiveness, it requires getting information to the right process at the right time. BI may respond to requests from other systems, may be contacted by event brokers based on circumstances occurring in real time, it may be the resource for presenting a single view of enterprise data not available in any other system. When BI can be responsive enough to participate in these collaborative processes, its value becomes a multiple of that from just classical decision support. The value that will accrue to organizations from the joint interaction of its enterprise systems will be shared by the BI investment. This argues for an architecture that is aligned with the enterprise.

2. **Expressiveness:** The models in today's data warehouses are limited to relational data models or dimensional data models. As useful as they are, they are not expressive enough to provide stakeholders with real modeling capabilities. There is still a gap between these data-centric conceptual models and the types of models that people need to apply these resources to their work directly. The construction of metrics, formulas, dependencies, and all other types of relationships in business has to be accessible to nontechnical people and the development, maintenance, and sustenance of their incremental and collaborative work has to be controlled and protected by the BI environment. Achieving business value in this area will be of major importance going forward. Pointing a BI tool at a data warehouse is not adequate.

3. **Completeness:** BI has been a piece of the puzzle, and an unconnected one, for historical reasons and for technological ones too. When BI fails to thrive in an organization, it is often because stakeholders despair of its inability to solve real problems from end to end. Understanding customer behavior is not of much value unless there is a way to project the correct response, implement it, and monitor the effects of the change. The integration of BI with operational systems has the promise of allowing individuals and groups to collectively solve problems and manage the ongoing implementation of the solution. The business value potential of this dwarfs what data warehouses have been able to deliver in the past.

Business benefits from most technology implementations are possessed of an inescapable shiftiness – the closer you examine them, the less certain and . . .

. . . measurable they become. ROI calculations in this regard are not very helpful where the benefits and costs are not much different. ROI in BI should not be a matter of 105% or 110%, it should be much greater than that. Even large variances in how the benefits are accounted for become unimportant when the gap widens. Implementing a new operational system involves controllable costs and very obvious, accountable savings (typically), but BI is about large improvements, important ones, and fundamental ones. One thing is clear, however, from over 20 years of data warehouse history – the implementations that generate the best ROIs are the ones that are used, where there is wide and deep propagation of BI usage. The more people and processes in an organization use BI, the more benefits will accrue, but scaling has to be accompanied by careful planning and solid architecture. If you want the most benefits from your efforts, you will need to design from an enterprise perspective.

CONCLUSION

ROI metrics can be manipulated to support a business case that is not, in reality, supportable. The true test of the adequacy and completeness of an ROI model is how thoroughly it pushes the envelope of the status quo. When the normal mode of operations in an organization is to cost out the acquisition costs of a development project and downplay the lifetime costs (TCO), an ROI calculation that takes this approach will always support a "best-of-breed" approach. Only when an ROI model challenges existing assumptions and takes a fresh look at the long-term implications can it arrive at anything other than the preordained solution. The calculation itself, whether it's DCF, payback period, or any other variant, is less important than the methodology of gathering the costs and benefits authentically. Experience with thousands of customers has shown that the single-architecture approach is much more likely to provide a favorable ROI over a reasonable time period then a best-of-breed approach assembled by all but a few of the most skilled and talented practitioners.

The era of massive cost cutting is nearing an end. Organizations are once again looking at enterprise-level technology solutions as the vehicles to propel them forward. Existing data warehouse and BI deployments that are underperforming will either be scrapped or reworked. The smooth integration of analytical information with internal and external operational processes will be at the top of most wish lists.

Most BI implementations are completely unprepared for this. The complexity of stitching together four or five or six technologies is expensive and risky enough. Having this structure cooperate with operational systems and providing the kind of reliability and response time needed to support online business processes is an order of magnitude more challenging. Is best of breed not a viable choice going forward? Not necessarily, but current data warehouse and BI best practices need to be drastically revised for it to be practical.

The managing layer on top of all this complexity can no longer be a manual process tended to by a few skilled technologists who know all the parts and connections.

Rather, it has to be a cooperative infrastructure that is bound to a lower-level physical implementation. To call this layer metadata is technically correct, but the term has been so diluted over the past decade, that it doesn't convey the depth and breadth of functionality that is needed.

Over the past 10 years, organizations have achieved universal connectivity and access to information, have renewed their aging systems as a result of Y2K through either remediation or replacement, and weathered brutal cost cutting as a result of a weak economy. At this point in time, the focus has turned to enterprise systems and architectures and the careful allocation of funding for any pure development work. For handcrafted best of breed to work going forward, there must be unifying metadata and controlling devices to tie all the layers together through semantic layers instead of direct physical connections. This would allow best-of-breed components to actually operate like components in the formal sense. At this point, there is no product available off the shelf that can operate as a unified modeling tool. In a best-of-breed world, nothing short of this will work over the long term, and the maintenance and support costs will drain initial ROI.

In an integrated, configurable environment such as SAP NetWeaver, with the SAP Master Data Management technology component and SAP BW, and an active metadata management facility, providing the level of services needed to support enterprise BI is far more likely. Organizations in 2004 are striving to be agile, collaborative, and efficient, which requires operating in a real-time hybrid operational/analytical mode. With all the complexity that entails, it is clear that a single architecture will yield a better ROI, is more likely to come online faster, and is less likely to fail, undeniably.

The benefits achieved to date from data warehouses and BI have been small compared to what is likely to happen in the next few years. The demands are increasing, the will is reemerging and the lessons learned are clear: BI must deliver to the entire enterprise and the only true benefits are those that leverage the entire information technology portfolio. BI started as end-user reporting and analysis, an activity completely off the radarscope of IT. Going forward, BI is an indispensable tool that is vital to the success, perhaps even the survival, of organizations in this increasingly complex and connected world. Getting it right the first time is more important than ever.

APPENDIX

The following examples of cash flows for ROI calculations are meant to be illustrative, not an actual case study. The numbers, though arbitrary, are reasonable, based on Hired Brains' experience with over 150 data warehouse and BI implementations. The options in a best-of-breed approach are nearly infinite and these numbers depict just a single observation.

TYPICAL BEST-OF-BREED "OPTIMISTIC" PROPOSAL

Year	1	2	3	4	5
Server Hardware	500,000	100,000	–	–	–
RDB Licenses	400,000	–	–	–	–
Storage/Backup/Recovery	300,000	–	–	–	–
ETL Software	150,000	–	–	–	–
Data Quality Software	–	175,000	–	–	–
BI Software	150,000	–	–	–	–
Consulting/Labor	850,000	200,000	200,000	200,000	200,000
Hardware Total	800,000	100,000	–	–	–
Software Total	700,000	175,000	–	–	–
Hardware CUM	800,000	900,000	900,000	900,000	900,000
SW CUM	700,000	875,000	875,000	875,000	875,000
HW Maint	144,000	162,000	162,000	162,000	162,000
SW Maint	126,000	157,500	157,500	157,500	157,500
Total	2,620,000	794,500	519,500	519,500	519,500
Total CUM	2,620,000	3,414,500	3,934,000	4,453,500	4,973,000
Disc. Factor	100%	91%	83%	75%	68%
PV Incremental Cost	2,620,000	722,273	429,339	390,308	354,825
NPV CUM Costs	2,620,000	3,342,273	3,771,612	4,161,920	4,516,745
Benefits	–	600,000	3,000,000	5,000,000	5,000,000
Benefits CUM	–	600,000	3,600,000	8,600,000	13,600,000
PV Incremental Benefits	–	545,455	2,975,207	6,461,307	9,288,983
ROI Initial	–	17.57	91.51	193.11	273.48
ROI/DCF Initial	–	16.32	78.88	155.25	205.66

Interest rate 10%

TYPICAL ADDED COSTS IN BEST-OF-BREED IMPLEMENTATION

Year	1	2	3	4	5
Addl ETL Effort	–	250,000	250,000	250,000	250,000
Integration to Portal	–	200,000	150,000	50,000	50,000
Addl Hardware	–	200,000	200,00	–	–
Second BI Software	–	–	375,000	–	–
Data Cleansing SW	175,000	–	–	–	–
Addl ETL SW	–	–	200,000	–	–
Total SW	175,000	–	575,000	–	–
Total SW CUM	175,000	175,000	750,000	750,000	750,000
Addl SW Maint	31,500	31,500	135,000	135,000	135,000
Addl Consulting/Labor	350,000	250,000	600,000	250,000	250,000
Benefits Lost	–	600,000	1,500,000	2,000,000	1,000,000
Total Added Cost	906,500	1,531,500	3,985,000	2,685,000	1,685,000
PV Incremental Added Cost	906,500	1,392,273	3,293,388	2,017,280	1,150,878
Total CUM Added Cost	906,500	2,438,000	6,423,000	9,108,000	10,793,000
NPV CUM Added Costs	906,500	2,298,773	5,592,161	7,609,441	8,760,319
ROI Actual	–	10.25	34.76	63.41	86.26
ROI/DCF Actual	–	9.67	31.77	54.89	69.96

Interest rate 10%

TYPICAL SINGLE ARCHITECTURE

Year	1	2	3	4	5
Server Hardware	750,000	–	–	–	–
Storage/Backup/Recovery	300,000	–	–	–	–
Package License	1,200,000	–	–	–	–
Consulting/Labor	850,000	100,000	100,000	100,000	100,000
Hardware Total	1,050,000	–	–	–	–
Software Total	1,200,000	–	–	–	–
Hardware CUM	1,050,000	1,050,000	1,050,000	1,050,000	1,050,000
SW CUM	1,200,000	1,200,000	1,200,000	1,200,000	1,200,000
HW Maint	189,000	189,000	189,000	189,000	189,000
SW Maint	216,000	216,000	216,000	216,000	216,000
Total	3,505,000	505,000	505,000	505,000	505,000
Total CUM	3,505,000	4,010,000	4,515,000	5,020,000	5,525,000
Disc. Factor	100%	91%	83%	75%	68%
PV Incremental Cost	3,505,000	459,091	417,355	379,414	344,922
NPV CUM Costs	3,505,000	3,964,091	4,381,446	4,760,860	5,105,782
Benefits	–	2,000,000	3,000,000	5,000,000	5,000,000
Benefits CUM	–	2,000,000	5,000,000	10,000,000	15,000,000
PV Incremental Benefits	–	1,818,182	4,132,231	7,513,148	10,245,202
ROI Initial	–	49.88	110.74	199.20	271.49
ROI/DCF Initial	–	45.87	94.31	157.81	200.66

Interest rate 10%

www.sap.com/contactsap

50 068 083 (04/03)