The Return on Investment (ROI) of Data Modeling

MARCH 2010

Tom Haughey

PRESIDENT, INFOMODEL LLC

Table of Contents

Executive Summary

Challenge

Businesses make decisions based on data. Without a clear understanding of the meaning and rules around this data, business decisions can be negatively affected. Data modeling is a time-proven method for understanding data, its interrelationships and its rules. Data management professionals have long understood the value of data modeling. However, business professionals often don't understand the value of data modeling. This is true even though history has shown that omitting the data model produces an inferior database design that prolongs system development, causes increased maintenance and reduces data quality. So how can you demonstrate to business management the return on investment (ROI) of data modeling?

Opportunity

Data modeling helps improve the quality of data, ensure the definition of high quality data requirements, and reduce development time, maintenance time, and redundancy -- among other benefits. The most conspicuous benefit for demonstrating the value of data modeling is the reduction in the cost of maintenance or development it will enable. Most organizations spend 70-80% of their software budget on maintenance.

Benefits

The value of data modeling can be demonstrated as an overall savings in maintenance or development costs. Viewed broadly across an organization's entire budget, this can be truly significant. The value of data modeling can also be seen at a more detailed level by the savings it will provide for development tasks on a specific project. Additionally, its value can be determined by identifying specific benefits that data modeling provides and then quantifying those project by project. Finally, data models can be reused in whole or in part on multiple projects which can result in significant savings to any organization.

SECTION 1

Perceptions of Data Modeling

Information drives the business and its business decisions. Data modeling is critical to understand the information needed to make those decisions. Yet, many business people don't understand the value it provides. Some perceive it as just documentation, as a bottleneck to rapid development, or even as too expensive to do. The data model is not just documentation because it can and will be forward engineered into a physical database. Not only is data modeling not a bottleneck to development, it can actually accelerate development and can significantly reduce maintenance. If data modeling is too expensive to do, then what is the alternative? If you do not use formal data modeling, then the data structures will be done informally and will rely on the intuition of software designers. Experience has shown that data structures developed without data modeling take longer to develop and often required extensive modification after they are implemented.

General Value Proposition of Data Modeling

Here is a very typical experience in the use of data modeling. We have three comparable and concurrent projects in the same organization: Project 1 uses data modeling from the start; Project 2 introduces data modeling late in development but before initial development of the system is completed; Project 3 never uses data modeling. Here are the results. Project 1 was implemented seamlessly and experienced no errors due to data maintenance for two years. Project 2 had to make many changes to the database design before implementation but was implemented successfully with a small amount of data maintenance subsequent to implementation, including data changes; Project 3 was continuously modifying the data structure during test and experienced a significant amount of data change after implementation, including the correction of several errors. In general, history has shown that omitting the data model results in an inferior definition of information requirements, prolongs the development process and increases subsequent maintenance.

SECTION 2

The Benefits of Data Modeling

The first task in determining the ROI of data modeling is to understand the benefits that data modeling can provide. To the data management community, the benefits of data modeling are many, well-known and not hard to recognize. The challenge is in identifying them and quantifying them to management

Let us identify the major benefits of data modeling with a case example of each. These examples are taken from real-world experiences with names omitted to "protect the innocent".

Summary of Data Model Benefits

- Improved data quality
- Better business requirements definition
- Greater reuse of assets
- Reduced data movement
- Reduced maintenance
- Accelerated development

Benefit 1: Improved Data Quality

Data modeling improves the quality of information by ensuring clear and consistent business definitions (or "metadata"). Metadata is the definition of a system asset, such as a database or table. Since one of the major components of a complete data model is metadata, the metadata in a data model will enable the data asset to be properly understood and utilized. This is important for use of the model by developers, by business people and for maintenance in future years.

Data modeling also improves the quality of stored data because it supports definition of data validation rules that ensure valid values are stored in data elements. For example, the data model can ensure that a valid Social Security Number is included for each customer, that correct states codes are used, that customers always have at least one address, and that address data is assigned to a customer.

Case Example 1: Data Quality

A marketing department in a hedge fund launches a major marketing campaign for their high-asset clients. "High-asset" individuals are defined as those who have more than $2MM in liquid assets. In reviewing the customer database, the marketing department based their criteria on the field called "Net Worth", assuming that this represented the liquid assets of the customer. Unfortunately, there was no clear definition of the meaning of the field, and no data model had been published. In actuality, liquid assets were stored in a field called "Balance2", which did not follow a good naming standard or have a good definition. The "Net Worth" field was an estimated value that is based on the sales reps estimation of total income, real estate assets, etc.—it had no relation to the customer's liquid assets at all. Because basic business definitions were not clearly defined in a data model, this very expensive marketing campaign was wasted on the wrong market segment. Fancy brochures, public presentations, visits to prospect sites and lunches at Tavern on the Green in New York City were geared at the wrong customers, eliminating the profitability of the campaign.

Benefit 2: Improved Business Requirements Definition

The process of data modeling, and the information gathering it requires, will uncover the main data requirements of the project. Without an understanding of the business requirements and business rules, the systems are of little value. Data modeling is all about understanding the business and its rules. Data modeling itself is governed by a set of data

modeling rules and principles. Enforcement of these data modeling rules and principles in and of themselves will help ensure the integrity of the data model deliverables. The rules of data modeling will help ensure correct capture of business rules. In addition, data modeling should be done using cross-checks that are self-validating and that will ensure that a model works when delivered. A good system is one that does what it is supposed to do.

Case Example 2: Improved Requirements Definition

In a financial organization, data modeling can easily reveal the subtle business rules and requirements between data elements such as customer and account. For example, it might seem obvious to a business sponsor that a customer can have an account. What the data model design process helps to discover is the subtleties in this relationship that affect how the supporting systems are designed and built. For example, a customer can have multiple accounts. Conversely, an account can also support multiple customers. How does this affect the business? All might go well until a sales agent tries to add another account for their customer, only to find that the software interface won't allow them. The sales agent is frustrated, the customer is angry, and lost business may result. Development costs are increased as IT needs to go back and redesign the system, this time with the correct requirements—in the long run spending more time and money than if the system had been built correctly the first time.

Benefit 3: Greater Reuse of Assets

Data modeling values data as a corporate asset and seeks to share these assets where possible. Data architecture in an organization can help to achieve this by providing an enterprise-wide perspective on data and deployed databases. Reuse of data assets saves the company money and accelerates the payback on the system by enabling a better quality (and earlier) implementation.

Reuse in a data model occurs in several ways. As models are created, they are stored in a repository and can be shared with other projects. Existing models can be shared with other projects and used to jump start them. Databases are intended to be shared. Systems are integrated if they use the same data. If they pass data from one to the other, they are interfaced.

Case Example 3: Reuse of Assets

In a consumer products company, a project is being reviewed for authorization to proceed to implementation. The project is proposing to build a new database to support its needs. The data architect discovers during the data model review that 75% of the model being reviewed is already implemented in an existing database and does not need to be re-built, a significant savings to the organization and improving time to market. This will improve the return on investment (ROI) for the project and reduce the payback period.

Benefit 4: Reduced Data Movement

Data movement is the transfer of data from one location to another. Data movement occurs by necessity in project such as data warehousing, business intelligence and master data management. It also occurs due to data redundancy, where the same data is maintained in different servers, schemas or tables. Data movement is the flip side of data redundancy. Data modeling looks to reduce data redundancy and in so doing, to reduce data movement. This is an often overlooked benefit to data modeling. The process of data modeling itself, with its view of data as a corporate asset, will help reduce an organization's reliance on redundant data, including redundant databases. A replica of a database may improve performance and have other benefits but each copy must be populated, maintained and synchronized with other copies. This synchronization requires a great deal of data movement in an organization.

Case Example 4: Reduced Data Movement

Some organizations have reported as much as 10-15% (or more) of their production jobs do nothing but move data. There is a high cost for this. Data model reviews, done by architects who understand the context of databases throughout an organization, can help discover opportunities to share data and reduce reliance on multiple copies of the same data. This will also reduce and simplify the number of extracts, loads, data transfers and data synchronization runs. A large portion of an organization's software budget is spent on maintenance, and on maintenance of interfaces.

Benefit 5: Reduced Maintenance

Reduction in maintenance is **the big ticket item** for showing the ROI for data modeling. It is generally agreed that 70-80% of an organization's software budget is spent on maintenance. A large chunk of that is spent maintaining interfaces between systems, which we just discussed.

Another major source of maintenance is the correction of errors. These errors can be reduced with the help of data models that define requirements up-front. For example, a major software vendor once revealed that to correct one error in the behavior of one instruction in one of their software products cost $100K. Why so costly? One instruction! Because the error has to be discovered and a correction designed. The correction has to be developed, then tested in unit test, system test and regression test, and finally documented, migrated to a release, announced and released. This is merely a single coding error. Imagine the cost of delivering software that does not do what it is supposed to do!

Case Example 5: Reduced Maintenance

A large international publisher and mail order company received a major change from the Postal Service, which required significant adaptive maintenance. They had very limited time to comply. Two of their systems were affected, one with a data model, the other without. The system with the data model was successfully modified overnight. The other system required weeks of work to implement the change, barely finishing it in time.

SECTION 3

Calculating the Benefit of Data Modeling

There are a number of methods for calculating the benefit of data modeling, and you should choose the method that resonates best with your business audience. We will cover five methods for calculating return on investment here: (1) Basic cost/benefit analysis; (2) a savings during the phases of development for one or more projects; (3) a savings shown via data model metrics; (4) percentage of savings on maintenance; (5) percent savings on development.

General Recommendations

The more thorough the breakdown of costs and potential savings, the better the analysis will be. The extra analysis makes the cost/benefit analysis more realistic and believable. It also gives the project a better understanding of the project tasks. A key factor in expressing the cost/benefit analysis is to understand what is important to the business at that time. For example, if maintenance is important (when isn't it?), then use method 4 below. If controlling development costs is the issue, use method 2 or 5. If specific benefits can be justified against specific projects, then use method 1. If multiple projects will use data modeling and they are new at it, then method 3 is useful.

Definition of Return on Investment

A simple Return on Investment expresses the desirability of an investment in terms of a percentage of benefit on the original investment outlay.

> Return on Investment = Net Benefit / Net Investment Cost * 100

In the ROI of data modeling, this would translate to:

> Return on Investment = Net Savings Due to Data Modeling / Net Investment Cost in Data Modeling * 100

For example, assume a savings due to data modeling of $500,000, and a data modeling cost of $250,000. The return on investment is 200%.

ROI does not consider the time value of money or the economic life of the project. See the appendix for other financial calculations that can be applied to the savings produced by each method.

Method 1: The Cost-Benefit Approach by Project

This method involves identifying the benefit amount associated with specific data modeling benefits on multiple projects. The first step is to identify the projects. The next step is to identify the benefits attributable to data modeling on each project. The final step is to quantify the benefit. It is common to quantify the benefit as a dollar amount saved. It could also be expressed as a reduction in FTEs (Full Time Employees). From this, based on the standard rate for an FTE of the appropriate level, the cost savings can be calculated.

Quantifying the Costs of Data Modeling

The first major task is to understand the potential benefits of data modeling, which we have seen above. The second major task is to identify the specific costs of data modeling. Below is a general list of cost categories. Some costs, such as hardware, software, training and support costs, may actually be sunk costs because the data modeling tools will run on existing servers across multiple projects. The exception is labor costs, which can be estimated and recorded on a project-by-project basis. This is part of the normal project planning process. The third major task is to quantify the benefits, which we will see in the Cost/Benefit method below.

Hardware. This includes the hardware costs to support the tools for data modeling, for storing multiple models that will be shared, and for storing supporting documents such as standards. The costs of specialized printers or plotters are used to document data models should be included.

Software. This includes the software product costs for data modeling tools.

Training. This includes educating the people who are needed to produce, participate in and review data models.

Support. This includes the installation support, help desk support and maintenance support for the other data modeling resources.

Labor. This includes the human costs for people doing data modeling on a project.

Labor

The following table, Table 1: Hourly Labor Rates by Role, will assist with collection of the labor involved in a project for all methods.

Hourly Labor Rates by Role	
Role	**Rate**
Business Analyst	
System Analyst	
Data Administrator	
Database Administrator	
Architect	
Software Engineer	
Test Engineer	
UI Designer	

Table 1: Hourly Labor Rates by Role

Quantifying the Benefits of Data Modeling

There are at least 3 ways to quantify the benefits of data modeling.

1. **Break the project into tasks and quantify the tasks.** To do this first requires that the benefits are related to one or more projects. The projects are then broken into tasks using normal work breakdown. This is normally done in project planning anyway. Each task is then assessed in terms of its cost in time *with* and *without* data modeling. The roles discussed above are then assigned to each task. Each role has a cost. The cost is time x rate. For example, in the Requirements phase, the task of defining data requirements could be defined as follows: create the conceptual data model, create logical data model, conduct use case walkthroughs, conduct data requirements reviews, create first-cut physical design. In the Technical Design phase for database design, the tasks could be collect volumetrics, apply data design optimizations, prototype the database design, iterate design, create final-cut physical. These can be quantified with and without data modeling.

2. Break the benefit into tangible and intangible aspects. Most benefits have both tangible and intangible aspects. In Table 2, Reduction in the Number of Interfaces is a tangible benefit. Interfaces require data movement. Each data movement production run

costs money, say $1000. If we can reduce the number of these runs by 100 runs per month, then we have saved $100,000.

If the benefit is less tangible, then ask the executive: "How much is achievement of this benefit worth to you?" Take the example of marketing in a large consulting firm. The effectiveness of marketing is judged by the number of leads generated. How much is it worth if we can generate 50% more leads than now? 100%? Their problem is that today their postal codes are 40% incorrect, SIC (Standard Industry Codes) are 30% incorrect, and 20% of the prospect data is duplicate. Because of all this, 50% of the data is out-of-date. Some of these problems are structural and some not. Data modeling can help with the structural problems by providing validation rules for the attributes and detection of duplicates. Once the structural rules are corrected, the data can be shipped of to an industry data organization (such as D&B) to update the data values to their current status. A marketing VP could easily quantify such benefits even though they appear intangible at first.

As other examples, how much is improved metadata worth to you? You have a major marketing database. 40% of your market segmentation runs have to be run and re-run to achieve meaningful segmentation because the metadata is poor. Each re-run has a cost. Or you have a mortgage database and you need to understand your risks. You need to know the exposure to jumbo upstate mortgages. With poor metadata, the jobs would have to be run and re-run till you get meaningful results. How confident can you be in the results if there is no clear definition of what "jumbo" and "upstate" mean? Reduction in the number of runs is tangible. Control over (and reduction in) risk is less tangible but much greater.

3. **Apply data modeling metrics**. This will be exemplified in Method 3 below.

Here is an example of quantifying the benefits of data modeling in terms of savings in FTE's.

	Data Modeling Benefit				
Sample Project	**Improved Requirements Definition**	**Data Quality**	**Improved Database Design**	**Improved Metadata**	**Reduced InterfacesDue to Data Sharing**
CRM Project A	.1 FTE for 12 months	.5 FTE for 4 months	.5 FTE for 4 months		.1 FTE for 12 months
BI Project B		.5 FTE for 4 months	.5 FTE for 4 months	.5 FTE for 12 months	.1 FTE for 12 months
Customer Project C				.5 FTE for 12 months	.1 FTE for 12 months
Enhance-ment Project D	.1 FTE for 12 months	.5 FTE for 4 months	.5 FTE for 4 months		.1 FTE for 12 months
				Totals	2.6 over 12 months

Table 2: Quantifying Data Modeling Benefits

Method 2: Percentage of Project Savings during Development Phases

This method will determine ROI as a savings against each major phase of development for a particular project or projects. The point is to determine what data modeling will save throughout the development lifecycle for the project(s) in question. This approach can work with planned projects or completed projects. For planned projects it shows what data modeling is expected to save. For completed projects, it identifies what data modeling is estimated to have saved. The phases of development can be quantified as shown in Table 3.

It is reasonable to expect with data modeling a savings of from 1 to 10% in development costs. Savings of 10% of more are not unreasonable during the *database design phase* or *during development*. Based on a solid logical data model, database design can be completed in anywhere between hours and days, a considerable savings over methods that do not use a data model. The example in Table 3 (which is based on a real case example) illustrates that. Data modeling during Requirements, during which an automated data modeling tool was used, was accomplished with a savings of 5%. Database design, which is done during Technical Design, used the tool to forward engineer the data model to a database design and achieved a much greater improvement—the savings here were 70%!. Optimization was directly applied to the forward engineered model, during which the tool enforced design standards. The tool was used to generate the DDL and stored procedures. Application Development and Test used other automated tools against the new database design and also achieved an improvement of 40%. The overall savings was 23.5%.

Development Task	Without Data Modeling	With Data Modeling	Savings Amount	Proposed Savings %
Requirements	$55,000	$52,250	$2,750	5.00%
Technical Design	$100,000	$30,000	$70,000	70.00%
Application Development & Test	$90,000	$54,000	$36,000	40.00%
Maintenance (5 years)	$375,000	$337,500	$37,500	10.00%
Total	$620,000	$473,750	$146,250	23.5%

Table 3 - Quantifying Development Savings

Method 3: Applying Data Modeling Metrics

The profitability of data modeling can also be determined by applying data modeling metrics. Examples of data modeling metrics are:

- Reduction in Error Correction due to data modeling: the reduction in the time and dollars spent making error corrections after databases are implemented.
- Reduction in Database Design time due to data modeling: the reduction in the time and dollars to perform the development task of database design.
- Improved Data Quality (DQ) due to data modeling. There are many published DQ metrics, such as reduction in Referential Integrity violations.
- Increase in data asset reuse: the time and dollars saved due to the reuse of existing data model objects.

The following example will illustrate this method by utilizing this last point, the savings due to increase in data asset reuse. This method utilizes the fact that reuse of an asset will save money. It then calculates the advantage at different levels of reuse. It addresses reuse of data model objects across multiple subsequent projects. A data model object is any

component of a data model such as entities, attributes, relationships and the entire model itself. The fundamental idea is that once any data model is created, it can be used to jumpstart other projects, thereby delivering a return on the original investment. This process can be enhanced if a set of tools, providing a data modeling infrastructure, are used to store, manage, retrieve and reuse these base models.

This first model serves as a base model. A base model is any previously created model for a given project. This calculation can take advantage of an enterprise model but does not require one. It only requires that some model has been created that can be reused.

The steps to determine ROI using this method are:

1. Identify the subsequent projects that can utilize the base model.
2. Identify the team members per each subsequent project.
3. Identify the base rate per team member.
4. Identify the time required on the subsequent projects for each team member.
5. Calculate the total cost per project.
6. Identify candidate rate of savings.
7. Calculate the savings.

The following chart in Table 4 will help lay this out.

TEAM MEMBERS	**LABOR RATE/HR**	**HRS PROJ 1**	**HRS PROJ 2**	**HRS PROJ 3**
Business Liaison Person	$40.00	125	100	150
Business Analyst	$40.00	125	100	150
Data Analyst	$40.00	300	250	350

PROJECT NAME	**LABOR COST**
Project 1	$22,000
Project 2	$18,000
Project 3	$26,000
TOTAL LABOR COSTS	$66,000

REUSE %	**SAVINGS AMOUNT**
0.00%	$0
25.00%	$16,500
50.00%	$33,000

Table 4 – Calculating the Savings from Reuse

Terms

Team Member: a participant in the project such as Business Person, Data Modeler.

Rate: the hourly cost per team member.

Project Hours: the hours spent in data modeling activities.

Labor Cost: calculated by multiplying project hours by rate.

Total Labor Cost: the sum of labor cost for all projects considered.

Savings: calculated by multiplying Labor Cost by Reuse %.

Method 4: ROI as a Percent Savings of Maintenance Costs

Data modeling helps reduce the cost of maintenance. At the heart of this is one simple principle: the earlier a problem originates and the later it is discovered, the more expensive it is to fix.

The Percentage Measure

One simple way to cost-justify data modeling is to measure the benefit of data model as a percentage of maintenance. This maintenance number should be available to the IT executive, such as a CIO. The monies spent on development and maintenance are available numbers within an organization such as in the budget or AFE (Authorization for Expenditure). The first task is to get management to agree data modeling will save them maintenance costs, using the anecdotal descriptions from the previous section.

The next task is to negotiate how much data modeling can save as a certain percentage of maintenance costs. The percentage has to be negotiated with management. The temptation would be to press for 5 or 10%. Great if you can get the executive to agree. However, method 4 works even if the percentage is as *apparently* little as 1 or 2 % of maintenance. The executive might say, "OK, I agree that data modeling will save us in maintenance but not more than 1% of maintenance costs. 1% is as high as I will go." The next question then is "What are your maintenance costs annually?" "I have a $500MM development budget. 80% of it is maintenance." This means that $400MM is spent on maintenance. The math is simple. Thereby in this case, data modeling can save $4MM. Admittedly, this is a gross number but it is not pure magic.

In Table 5, the non-italicized entries (Proposed Savings %, Project Name, Modeling Cost and Maintenance Cost) are given by the user. The italicized entry (Maintenance Savings) is calculated, as is ROI, using the formulas above.

Note that in this example a conservative savings percentage of 1% was used.

Other Important Terms

Here are some other terms and considerations we will use in this method and method 5.

Savings % is the estimated percentage of reduction due to the use of data modeling.

Modeling Cost is the cost of labor to perform data modeling activities.

Maintenance Cost is the estimated cost to maintain the system over its life, taken from the AFE or Budget.

Maintenance Savings is the dollar reduction in maintenance resulting from the use of data modeling, such as a reduction in the amount of corrective maintenance. This savings should be directly attributable data modeling. It is computed by multiplying the Maintenance Cost by the proposed Savings %.

SAVINGS PERCENT %		1 %		
PROJECT ID	**MAINTENANCE COST**	***MAINTENANCE SAVINGS***	**MODELING COST**	***ROI***
12345	$25,000,000	*$250,000*	$70,000	*357%*
56789	$36,000,000	*$360,000*	$150,000	*240%*
98765	$45,000,000	*$450,000*	$300,000	*150%*
54321	$180,000,000	*$1,800,000*	$500,000	*360%*
13579	$45,000,000	*$450,000*	$350,000	*128%*
TOTAL	$331,000,000	*$3,310,000*	$1,370,000	*242%*

Table 5 - Calculating the ROI from Maintenance Savings

Method 5: ROI as a Percent of Development Costs

Essentially, this is the same method as Method 4 except that it uses development costs as the basis, not maintenance costs. Maintenance cost is not included in this spreadsheet. The definition of other terms and the formulas for calculation are the same as in Method 4.

Other Important Terms

Here are some other terms and considerations we will use in this method. Other terms are identical to method 4.

Development Cost is the estimated cost to develop the system, taken from the AFE or Budget.

Development Savings is the dollar reduction in development resulting from the use of data modeling. This savings should be directly attributable data modeling. It is computed by multiplying the Development Cost by the proposed Savings %.

Table 6 shows how we can use these costs and savings to determine the ROI as a percentage of development costs.

DEVELOPMENT SAVINGS %		1 %		
PROJECT ID	**DEVELOPMENT COST**	***DEVELOPMENT SAVINGS***	**MODELING COST**	***ROI***
12345	$8,000,000	*$80,000*	$40,000	*200%*
56789	$12,000,000	*$120,000*	$45,000	*267%*
98765	$15,000,000	*$150,000*	$30,000	*500%*
54321	$60,000,000	*$600,000*	$120,000	*500%*
13579	$15,000,000	*$150,000*	$34,000	*441%*
TOTAL	$110,000,000	*$1,100,000*	$269,000	*409*

Table 6 - Calculating ROI Based on Development Savings

Appendix: Additional Methods for Determining Profitability of an Investment

Two additional measures are used to express the financial desirability of a project, namely, Payback Period and Net Present Value. Here is a summary of these methods of calculation.

Assumptions for These Methods

Always document the assumptions going into your analysis, such as:

1) **System Lifecycle** in years, for example, 7 years.

2) Rate for incurring **Data Modeling Costs**; probably best to assume a constant rate.

3) Rate for achieving **Savings in maintenance or development**; probably best to assume a constant rate.

4) **Discount rate**. This is the value of money, similar to an interest rate. Rate tables are publicly available. Actual rates will vary. We suggest a rate of 5-10%.

5) Automated data modeling tools are used to hold the data model.

Payback Period is simply an estimate of how long it will take for an investment to pay for itself. This method resonates well with management. However, it does not consider the time value of money. A simple formula for this is:

Payback Years = Net Investment Cost / Net Benefit per Period

Assume data modeling cost of $100,000 and net benefit of data modeling per year of $50,000. In terms of the payback of data modeling, this would be:

Payback Years: Data Modeling Cost/ Data Modeling Net Benefit

$100,000 / $50,000 = 2 years

Net Present Value (NPV) seeks to overcome the disadvantage of non-time value methods. The NPV is an amount at present which is equivalent to a project's cash flow for a particular interest rate. It is the sum of the yearly benefit discounted by the cost of capital. A simple formula for this is:

NPV = Sum of (Year 1 Benefit – Cost) / (1 + Discount Rate) + (Year 2 Benefit – Cost) / $(1 + \text{Discount Rate})^2$ + (Year 3 Benefit – Cost) / $(1 + \text{Discount Rate})^3$ . . . (Year n Benefit – Cost) / $(1 + \text{Discount Rate})^n$ - Initial Investment Cost

In our example, assume a yearly benefit of $50,000, an initial investment of $100,000 and a discount rate of 10% for 3 years. The discount rates used are from a discount rate table. The discount factors are: year 1 = 1.10, year 2 = 1.21, year 3 = 1.33). In the ROI of data modeling, this would be:

NPV = $50,000/1.10 + $50,000/1.21 + $50,000/1.33 - $100,000 or

NPV = $45,454.54 + $41,322.31 + $37,593.98 - $100,000 = $24,370.83

The **NPV** is $24,370.83.

SECTION 4

Conclusions

Data modeling is an important and valuable discipline in building systems. Its benefits can appear intangible. The challenge is first to find the pertinent benefits and then to quantity them. Various methods are available to do this. Data modeling can offer a significant reduction in major system costs such as maintenance and development. These dollar savings can be truly dramatic. It has other benefits that enable the business to understand its data better and thereby make better business decisions, such as improved quality of data and reduced redundancy.

SECTION 5

References

Gary Flye, "The Monetary Value of Data Modeling", Presented At Wilshire Conferences, San Antonio Texas, 2005.. This is an excellent presentation on this topic.

Marcie Barkin-Goodwin, "'Improve Your Bottom Line - Increase Your Modeling Return On Investment', Presented at DAMA Ohio Chapter Meeting 2005, http://www.damacoc.org/presentations/2005_01_13_Marcie_Barkin-Goodwin.pdf. This useful presentation focuses on the value of reuse of data models.

The author would like to thank David Lowinger and Michael East for their helpful ideas.

SECTION 6

About the Author

Tom Haughey is considered one of the four originators of Information Engineering in America. He has been an independent consultant and educator in data management since 1983. For over two decades he has been delivering successful data management solutions in the area of information architecture, business intelligence, master data management, database, data modeling, data warehousing and OLTP (On-line transaction processing). He has worked in many industries such as insurance, consumer products, finance, government and pharmaceutical. He was formerly Chief Technology Officer for the Pepsi Bottling Group and Director of Enterprise Data Warehousing for Pepsico. He was also Vice President of Technology for Computer Systems Advisers, who marketed the CASE tools called POSE and SILVERRUN. He worked for IBM for 17 years as a Senior Project Manager. He is author of many articles on Data Management and Data Warehousing, and is contributor to Information Management's Ask The Experts Column.

Visit www.erwin.com for more information