ER/Studio Data Architect Whitepaper

# THE ROI OF DATA MODELING

BY TOM HAUGHEY, INFOMODEL, LLC

Learn how to calculate the Return On Investment (ROI) of data modeling.

# TABLE OF CONTENTS

# EXECUTIVE SUMMARY

For over three decades now, data modeling has been the leading discipline for understanding business data requirements and representing them in a precise, understandable structure. Today, more than ever, businesses rely on data for their decision making, sometimes even vast amounts of data. To those in data management, data modeling has repeatedly proven its business value and needs no further justification. They have seen the tangible value of the model and the equally tangible danger of omitting it. To others, because these benefits are not so clear, data modeling requires systematic economic justification.

This can be done by showing the economic value of real data modeling benefits such as improved requirements definition, reduced maintenance, accelerated development, improved data quality and reuse of existing data assets. Client experiences are available that show the benefit of data modeling in each of these areas. These economic benefits can be expressed in different units of measure, such as dollars saved, human resource costs saved, or a percentage saving on different development expenditures. These benefits can also be collected and aggregated at different levels of detail, such as by project or by development phase. Maintenance remains the largest expense in most development budgets, generally accounting for 50-80% of the budget. Reduced maintenance is, thereby, the big ticket item in savings due to data modeling.

To maximize these benefits, data modeling must be done in a productive way. It must be iterative, incremental and collaborative. The day of monolithic projects is over. Modeling must progress through different levels, such as from the conceptual level of planning, to the logical level of business detail, to the physical level of the implemented database.

Challenges exist, and new ones surface regularly. New technologies and methods, such as agile development, column-oriented databases, NOSQL and big data, put data modeling under fire. To survive and sustain its momentum, data modeling is adapting, redefining its role in these trends, but will continue to play a key role in each of these innovations.

# SECTION ONE INTRODUCTION

Data Modeling is the activity of defining the information needs of an organization by classifying the objects of interest and their interrelationships. A simple return on investment (ROI) formula expresses the desirability of an investment in terms of a percentage of benefit on the original investment outlay.

This paper addresses the measurement of ROI of using data modeling within an organization. Reaping the full benefits of data modeling is achievable only by using data modeling properly. The remainder of this section explains proper use of data modeling.

## BASIC DEFINITIONS

The term “data model” implies three components:

1 **The model diagram** with its entities, attributes, relationships and keys shown via its boxes and lines, etc.

2 **Definitions** of all data model objects (metadata), especially definitions of: entities, attributes, identifiers and relationship roles.

3 **Supplementary business rules.** These are constraints that cannot be represented in the structure of the model. For example, a data model can show that a Customer can have a PO but a PO must belong to a customer. However, it is not possible in a data model that a Purchase Order must be awarded to the vendor with the lowest proposed price; this is a non-structural rule.

Three main characteristics of data modeling are that it is:

1 **Incremental**, because the data model is built in pieces. Data models are divided into subject areas or partitions to allow them to be created in pieces.

2 **Iterative**, because the data model is not done in one sequential pass. It is revisited and sometimes rethought multiple times.

3 **Collaborative.** Because the data model is the product of multiple viewpoints, input to the data model is a cooperative effort involving business people and other subject matter experts, data professionals, and database administrators. The data model is created by one person (or a few very congruent minds) but the participation of many people is required to contribute to it. A data model must be harmonious with other data models within the organization.

It is the contention of this paper that data modeling is inherently incremental, iterative and collaborative. In fact, throughout the 1980’s and ‘90s, few disciplines have emphasized these requirements more. With the evolution of development methodologies in the 2000’s and 10’s, this is even more evident.

# LEVELS OF DATA MODELING

Data modeling should be done in three levels.

1 **Conceptual.** Contains major business entities, major attributes and business relationships. This model is used as a planning tool at a business level or initial project data model.

2 **Logical.** Contains all the data entities, all attributes and all data relationships. A complete model is accurate and detailed, and identifies all the required data. A good model offers a number of options for the physical implementation.

3 **Physical.** Contains table names, columns, indexes, and constraints as well as partitioning. This is the implementation model, targeting a specific physical DBMS and DBMS type, such as DB2, Oracle, MongoDB, Hadoop, SQL Server, etc.

Contrary to what some agilists say, data modeling is special because data is special and has deeper ramifications than code. Data is a corporate and reusable asset, not just an application or program assets, and it affects every application (and application component) that uses it.

# SCOPING DATA MODELING PROJECTS

To measure ROI, and in fact to have a successful project, data modeling must be meaningfully scoped. This means that the project, not just the data modeling activities within it, has to be meaningfully scoped. Scope should generally be expressed in function or process terms. As an example, it is not sufficient to say that this project deals with “customer data”. That is too ambiguous for a scope. A project that deals with “customer credit” is more tangible.

Five simple guidelines help govern project scope:

1 A project should be divided into **increments**.

2 These increments should be **short and manageable** in breadth and depth.

3 Each project should provide for **early delivery** of a prototype for testing and verification.

4 Although the final data model should be the responsibility of a single data modeler (or a few congruent minds), a **cross-functional team** that provides collaboration among team members for identification of data elements is essential.

5 The work should be **time-boxed.**

Data modeling does not have to take forever. Proper scoping will allow a project to avoid what is commonly called Big Design Up Front (BDUF). In this approach, nothing is constructed until the entire design is finalized. One major problem with this is that inevitably requirements will change. Due to the time required to deliver a very large project, extreme change is likely making obsolete in the later stages things that were defined in the early stages. Dividing work into increments does not eliminate change; it just makes it more palatable. If the same project is divided into increments, the same changes in business requirements will occur over the same period, but they can be applied to more manageable increments, and the business has had the payback from the increments already delivered.

## ENTERPRISE STANDARDS

If an enterprise data model is available, it should be used as a guide, and relevant portions of it can be used to jumpstart a project. An organization should have data modeling standards, naming standards, and metadata standards, and these should be used.

## THREE TYPES OF SYSTEMS

Data modeling is relevant to all types of systems.

1 **New development,** where a system is built from scratch. Here data modeling is used in the conventional top-down/bottom-up/middle-out approach.

2 **Maintenance,** where changes and enhancements are applied to an existing system. Here reverse engineering is first applied to an existing database, the resulting model is enhanced, and then the data model is expanded as in new development.

3 **Rapid development,** where the project is done in small increments that iterate through analysis, design and construction of the increments. Here data modeling occurs in each increment.

In each of these types of systems, data modeling is used, although it plays a slightly different role.

## PLACEMENT OF DATA MODELING (LOGICAL & PHYSICAL)

In business intelligence, two complementary models can be used: the more normalized model of the central data warehouse, and the more dimensionalized model of reporting structures. Both rely on data modeling.

In operational systems, there is traditionally a clear correspondence between the data model and its implementation. NOSQL poses some differences, even in transactional NOSQL systems. The conceptual and logical models still represent the data and its constraints. These can and should be done using ER modeling even in NOSQL because even NOSQL data structures must obey business rules. The physical model is where the real difference is. It is based on query analysis and represents the unique NOSQL implementation. For example, if the NOSQL data store is hierarchical, then it is in the physical model that this appears. If using NOSQL, there will be a greater transition between the logical and physical models.

Column oriented databases can achieve great improvements in analytic query performance. Column orientation occurs at the storage layer, not the logical layer. Consequently, ER modeling, strongly driven by query analysis, is still pertinent.

Big data presents another case. By its very nature, big data is not amenable to data modeling in advance. Yet even big data cannot spontaneously combust into order. The patterns have first to be discovered before they can be modeled. Data models can play several roles. They can be used to serve as the abstract layer used to manage the data stored in physical devices. Today we have large volumes of data with different formats stored in all kinds of devices. The big data model can provide a logical layer to manage this data, and can offer a basic data architecture for using applications, which will reduce development costs and take advantage of data reuse. Some big data may be so transitory in nature that modeling it does not make sense because it will be used and then discarded or replaced.

# **SECTION TWO** DATA MODELING VALUE PROPOSITIONS

## THREE WAYS TO CALCULATE ROI

There are three classical ways to calculate ROI:

**1 Revenue enhancement,** simply expressed, means that an organization will generate more money as a result of doing the given project. Such systems directly affect the bottom line.

**2 Cost savings** refers to benefits of a project that can be added back into the budget and spent. For example, an organization negotiates a 10 percent decrease in the price of a consulting contract, resulting in a savings of $100,000 that year. This is an example of cost savings or reduction.

**3 Cost avoidance** also causes future spending to fall, but by slowing the rate of cost increases. For example, an organization resists a vendor's attempt to raise prices by 5 percent, allowing the agency to avoid spending an additional $200,000 that year. Without this cost avoidance, future spending would have increased more.

## THE BENEFITS OF DATA MODELING

Data modeling classically provides many benefits; among them are the following:

**Better data requirements definition.** Data modeling provides clear focus on the requirement beyond what use case modeling can surface. It can effectively focus on the rules that govern the relationship across elements.

**Faster development.** It is faster to do a project with data modeling than to do the same project without it because data modeling is a known discipline, has inherent rules, is supported by a knowledge base, separates the business from technology, and naturally generates enriching questions.

**Reduced maintenance in terms of error correction.** Data modeling has a simple principle, functional dependency, which states that data is placed where it belongs. Data modeling provides rules to determine functional dependency. This reduces the incidence of misplaced data, which is harder to correct, and eliminates needlessly redundant data.

**Better data quality.** This requires data validation and edit rules, plus focus on business rules.

**More reuse of data assets.** Will take advantage of known assets and enable better consistency between databases and applications.

**Easier change management.** Because the data model provides a single source for data knowledge, change is easier to apply.

**Better systems integration.** This is a major requirement today due to the prevalence of mergers and acquisitions.

These benefits translate directly into value propositions for data modeling. A value proposition is an innovation, service, or feature intended to make a company or product attractive to a market.

## CASE STUDY

The fundamental value proposition of data modeling is that systems can be created better, faster and with fewer errors when using data modeling. Real world experiences have confirmed this. Here is a very representative test in the use of data modeling. An organization compares three equivalent internal projects: Project 1 uses data modeling from the outset; Project 2 introduces data modeling late in design but before deployment of the system; Project 3 never uses data modeling. Project 1 was implemented on time and had no data error correction within the first two years. Project 2 found it necessary to make many changes in database design before implementation but was implemented successfully, though it experienced a small amount of data error correction subsequent to implementation, including changes to the data; Project 3 frequently modified the data structures during test and had to make many changes after implementation, including the correction of several errors in business rules. These results are neither isolated nor simply anecdotal. Experiences have shown that omitting the data model results in an inferior definition of information requirements, prolongs the development process and increases subsequent maintenance.

## DATA MODELING BENEFITS AS VALUE PROPOSITIONS

### VALUE PROPOSITION #1: BETTER DEFINITION OF THE DATA REQUIREMENTS

The overarching benefit of data modeling is the better definition of business data requirements. A good system is one that does what it is supposed to do. The process of data modeling, and the information gathering it entails, will capture the data requirements of the project in an effective way. This means that the requirements will be correct and complete. Data modeling can generate more questions, provide more integrity guarantees and discover more business rules than any form of process modeling, use case modeling or workflow modeling. It is immeasurably more efficient than prose definition of data needs.

Without an understanding of the business requirements and business rules, a system will fail to meets its requirements. Data modeling is all about understanding the business and its rules.

Data modeling is governed by rules and principles which make it an efficient vehicle for stating data requirements. A data model is more expressive and less verbose than capturing requirements any other way. Models generate questions that would otherwise escape. This is a natural byproduct of the model. The visual nature of the data model facilitates communication. Business people, and other subject matter experts, can easily be taught to interpret data models so they can validate them independently. One of the great benefits in having business subject matter experts review the model is cross-pollination. Business users not only contribute their knowledge but also learn about other parts of the organization.

Data models are guided by modeling rules and principles. Enforcement of these in and of themselves will help ensure the integrity of the data model. The rules of data modeling will help ensure correct capture of business rules. In addition, data models should be examined using crosschecks that are self-validating and that will ensure that a model will work when implemented.

Data modeling efficiently creates precise requirements because it is an engineering-like model. Data modeling captures good requirements early in the development process and corrects any changes equally early. A good data model will reduce development cost because fewer unknown or unanticipated requirements will be discovered during the application construction process.

## CASE STUDY

To make this tangible, consider the case of a major software vendor who stressed the importance of good requirements. They postulated years ago that their cost to no-op an instruction in their software products was $100,000. A no-op is a null instruction in assembly language that causes execution to pass to the next instruction. If it costs $100K to no-op an instruction, imagine how much it costs to deliver a product that doesn't do what it is supposed to do, that is, one that doesn't meet requirements. Consequently, a major goal of data modeling is to ensure that correct data requirements are defined.

### VALUE PROPOSITION #2: REDUCED MAINTENANCE

Data modeling significantly reduces maintenance costs holistically and discretely. It will reduce maintenance requirements at large throughout an organization and on individual projects.

It is no secret that the largest piece of most development budgets is maintenance. Historically, but conservatively, maintenance accounts for 50-80% of development budgets. Reduction in maintenance will have significant impact on project costs.

At the heart of this is one simple principle: the earlier an error is discovered, the less expensive it is to fix. A corollary to this is that the earlier an error originates and the later it is fixed, the more expensive it is to fix. An error in the data requirements that is discovered during data modeling is inexpensive to fix. If however that error is not discovered until way into the coding process, it will be exponentially more expensive to fix.

Data modeling will allow an organization to catch errors early, reduce the incidence of errors, and make it easier to perform maintenance. This includes error correction and adaptive maintenance. Standard data model policy checks, such as model reviews, walkthroughs and scenario analysis, help with this. Data modeling can make it easier to apply changes and implement enhancements. Non-redundancy in models helps with this. This means that data modeling will increase error prevention and reduce the cost of error correction. This includes the cost to enhance a system.

## CASE STUDY

A large international food supplier received a major change from a government agency, which required significant adaptive maintenance. They had a very short time to comply. Two systems were affected, one with a data model, the other without. The system without the data model was a file-based system. The system with the data model was successfully modified overnight. The other system required weeks of work to implement the change, and the change was finished just in time.

### VALUE PROPOSITION #3: ACCELERATED DEVELOPMENT

With data modeling, developers can focus on development, not discovery of requirements, and can develop with fewer errors during the development cycle.

DBMS schema designs can be generated and maintained entirely from the data models (normalized and dimensional). This is so because modeling tools automatically create DDL scripts, which are sometimes lengthy and complex. This is real code. Data models allow easy modification of existing scripts. Data models help developers visualize and understand the business area's data structures. Online, synchronized model management provides shareable models that enable collaboration and sharing.

Data modeling will allow systems to be delivered as early as possible. Early delivery of a system will mean earlier payback on development. An earlier payback means increase revenue from systems.

Data modeling needs to be an integral part of any agile development project including any of the agile methods used, such as Scrum, XP, etc.

## CASE STUDY

Consider the case of a large data warehouse in a financial organization. The platform was a large scale MPP (massively parallel processing) system with over 50 nodes. The entire database design was maintained in a data modeling tool. The entire database was generated directly from the tool. As business or system changes occurred, they were applied to the logical data model, and forward engineered to the physical model. Then the scripts to apply the changes were generated and applied directly from the tool, hugely accelerating and simplifying their implementation.

### VALUE PROPOSITION #4: IMPROVED DATA QUALITY

Data modeling improves the quality of data in several ways. It supports enforcement of domains of values by ensuring that only valid values are stored in fields; by defining and enforcing editing rules; by insuring relationship cardinalities and integrity; and by requiring meaningful metadata. Domains of values identify valid values that can be used in fields. Data modeling allows for definition of mandatory, optional and postponed attributes. A postponed attribute is one that cannot be entered at data creation time but must be introduced later for the entity to be complete. Data validation rules ensure that values correspond to legitimate data formats and editing rules. Data modeling improves data integrity by providing for relationship cardinalities and allowing relationship constraints. It enforces referential integrity, which improves overall data integrity. A proper data model expects good metadata. For example, the data model can ensure that valid state codes, zip codes, phone numbers and business party names are included for each customer, that customers always have at least one address, and that address data is assigned to a customer.

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.

## CASE STUDY

A large consulting firm brought in a new VP of marketing whose primary measure was to generate leads. Upon examining the database, he found that he could not generate leads. There was no data model and the data itself was a train wreck. The problem was the dreadful quality of the data. For example, postal codes for customers and prospects were 40% incorrect, SIC (Standard Industry Codes) were 30% incorrect, and 20% of the customers were entirely redundant. Zip codes and countries were embedded in addresses. Engagement managers put their own names in for customer contact information, and on and on. On top of this, 50% of the data was obsolete. Some of these problems were structural and some not. Data modeling corrected the structural problems by providing validation and integrity rules for the attributes and by detecting duplicates. Referential integrity was added. Data entry validation was enforced. Once the structural rules were corrected, the data was shipped to Dun & Bradstreet to update the data values. Finally, now the VP was able to generate leads.

### VALUE PROPOSITION #5: LEVERAGING EXISTING DATA ASSETS

It is important that an organization leverage existing data assets. This means the reuse of data models, including fully populated databases. Reusing existing databases where a pertinent database exists should take priority over creating application-specific redundant databases. For example, say a consumer products organization is building a database to cover equipment maintenance services. In addition to creating tables to cover services and to record the services performed, the application plan calls for creating a new database for locations, equipment types, and service personnel, among others. However, say, this data has already been created and is in use by the system that was used to install the equipment in the first place. Creating a database that is redundant in whole or in part will create significant extra cost. It will involve extra support to create the database, extra interfaces to move data back and forth between these databases, and synchronization procedures to avoid update anomalies. Instead, where possible, the new application should use the data structures and data values for these from the existing database.

A further implication of this value proposition is that data modeling supports better data and systems. This is no small task. In today's markets, mergers and acquisitions are a major strategy for business growth and profitability. For example, say two large financial organizations decide to merge. Each has its own databases, but unfortunately they do not have good data models and there are major differences between their common databases. It can easily take these organizations years to integrate, even just to consolidate, their databases for customers, accounts, transactions, and products, to name a few, if they have to do it manually.

## CASE STUDY

In one financial organization surveyed it was discovered that 15% of their production jobs did nothing but move data. A reduction in that number can represent a significant savings. It also reduces the risk of inconsistency in the data. If a data synchronization procedure is not performed, or is performed incompletely, data across those databases will be inconsistent. The company implemented a data architecture group that participated in project reviews and all projects had to present a data model in the review. They were able to prevent the further proliferation of redundant database and plan the progressive elimination of existing ones.

# SECTION THREE CALCULATING ROI

## DEFINITION OF RETURN ON INVESTMENT

A simple Return on Investment (ROI) formula 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 is expressed as:

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

For example, if the savings due to data modeling is $250,000, and the data modeling cost is $125,000, then the return on investment is 200%.

ROI alone does not consider the time value of money or the economic life of the project.
Three methods of ROI determination will be discussed in this paper:

1. Cost/benefit analysis
2. Percent savings of development
3. Percent savings of maintenance

These represent some valid approaches to calculating data modeling ROI. It is up to you to choose the approach that best addresses the concerns of your business audience. Some complex situations may require multiple methods.

## METHOD 1 COST/BENEFIT ANALYSIS

This basic approach gathers the various costs and benefits and compares them. The process is simply to identify the quantifiable costs, identify the benefits realized, and then to quantify each benefit as a cost saving.

The more complete and detailed the breakdown of costs and savings is, the better the analysis of cost/benefit can be. A more detailed analysis makes the cost/benefit analysis more accurate. It also provides for project management a better understanding of the project tasks.

Identifying the costs and the benefits is not always an easy task. As a general rule, it is advisable to collect as many data points as possible to make the analysis more realistic and thereby believable. It is up to you to decide how to express these in terms of two factors: the unit of measure and the granularity. Typical units of measure are dollars, hours, percent, FTEs (full-time employees). These can be collected by projects, project development phases or project tasks, which represent different granularities. For example, costs and benefits can be measured as dollars amounts by project, or FTEs by project phase, or any other combination.

## QUANTIFYING THE COSTS OF DATA MODELING

Table 1 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. Labor costs can be estimated and recorded on a project-by-project basis. This is part of the normal project planning process.

## TYPES OF COSTS

| HARDWARE | SOFTWARE | TRAINING | SUPPORT | LABOR |
|---|---|---|---|---|
| This includes the 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. | This includes the software product costs for data modeling tools, including license purchases and maintenance fees. | This includes educating the people who are needed to produce, participate in and review data models and metadata. | This includes installation support, help desk support and maintenance support for the other data modeling resources. | This includes the human costs for people doing data modeling on a project. |

Table 2, Hourly Labor Rates by Role, will assist with collection of the labor involved in a project for all methods. This can be customized for specific team members and hourly rates.

## HOURLY LABOR RATES BY ROLE

| Role | Hourly Rate |
|---|---|
| Business Analyst | $ |
| System Analyst | $ |
| Data Analyst | $ |
| Data Modeler | $ |
| Database Administrator | $ |
| Architect | $ |
| Application Developer | $ |
| QA Engineer | $ |

### QUANTIFYING THE BENEFITS OF DATA MODELING

There are at least 3 ways to quantify the benefits of data modeling in this approach. Each of these ways can be used effectively at project planning time.

### QUANTIFY THE DEVELOPMENT WORK

This works well if the benefits can be related to one or more projects. To use this approach, the project work is broken into tasks using normal work breakdown. This is fairly standard part of project planning. The cost in time with and without data modeling is then associated with each task. The roles in Table 2 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 the logical data model, conduct use case walkthroughs, conduct data model reviews, create the 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 the design, and create the final-cut physical design. These are quantified with and without data modeling.

### DISTINGUISH TANGIBLE AND INTANGIBLE BENEFITS

Some benefits are easy to identify because they are very tangible. Other benefits are less tangible and require some form of estimate, preferably by the responsible management. The benefits of data modeling identified above will serve as the basis for this. Most projects will have both tangible and intangible benefits.

Here are three examples of tangible benefits:

1. Reduction in redundant databases,
2. Reduction in interfaces,
3. Improvement in a business process.

The rationale is that the use of data modeling will result in a reduction in the incidence of redundant databases, and thereby the reduction in the number of interfaces. Reduction in the number of interfaces is a tangible benefit identified in Table 3. Each interface requires a data movement job. Each data movement job has a cost, a number that computer center operations can provide; for example, a standard cost of $1000. If we reduce the number of interface runs by 100 runs per month, then we have saved $100,000 per month.

## CASE STUDY

Here is an example of a business process improvement. Our company provides vending machines stocked with snacks. Each visit of a service rep to a vending machine has a standard cost. Ideally, a rep should arrive when the machine is 50% empty. In practice, we discover that sometimes the rep arrives too soon, and the machine does not need replenishment thereby wasting the visit. Other times the rep arrives too late, the machine is nearly empty, thereby losing revenue. We discover that with better data we can reduce the number of visits by one per month per machine. We have 50,000 machines that are visited 4 times per month. Each visit costs $100. Our goal is to reduce this by one visit per month per machine. This is a very tangible benefit.

Benefits that are less tangible need to be postulated by responsible management. A possible approach is to ask responsible management the following questions: “What benefit will you get from this system? How much is this benefit worth to you?” Say you are implementing a personnel system. Management says that this system will enable them to reduce disability payments. A natural follow-up question is “by how much”. HR says the system will reduce such payments by at least 1% per year. If the total of disability payments per year is $50MM, then the savings is $500K. Generally, management has a clear understanding of the benefit expectations of systems.

### EXAMPLE: COMPARING COST AND BENEFITS IN FTES

This example involves identifying the costs and benefit amounts associated with specific data modeling benefits across multiple projects. The unit of measure here is FTEs; the granularity is the project. The previously identified benefits of data modeling are used as the basis. The first step is to identify the projects. The next step is to identify the benefits attributable to data modeling on each project. These two provide the X and Y axis. The final step is to quantify the benefit, which is the cell. As we said earlier, the data modeling manager needs to choose the unit of measure and the granularity. It is common to quantify the benefit as a dollar amount saved. It could also be expressed as a reduction in FTEs. From this, based on the standard rate for an FTE of the appropriate level, the cost savings can be calculated.

The following example in Table 3 quantifies the benefits of data modeling in terms of savings in FTEs, which is the unit of measure, and sample projects by data model benefit (or value proposition).

## QUANTIFYING DATA MODELING COSTS AND BENEFITS

| Sample Project | Improved Requirement Definition | Data Quality | Improved Database Design | Improved Metadata | Reduced Interfaces Due to Data Sharing |
|---|---|---|---|---|---|
| Project A<br>**BI** | .2 FTE for 12 months | .5 FTE for 4 months | .5 FTE for 4 months | | .1 FTE for 12 months |
| Project B<br>**Sales** | | .5 FTE for 4 months | .5 FTE for 4 months | .5 FTE for 12 months | .1 FTE for 12 months |
| Project C<br>**Enhancement** | | | | .5 FTE for 12 months | .1 FTE for 12 months |
| Project D<br>**CRM** | .2 FTE for 12 months | .5 FTE for 4 months | .5 FTE for 4 months | | .1 FTE for 12 months |

**TOTALS: 2.8 OVER 12 MONTHS**

**Assumptions:**
The benefits listed come from the value propositions of data modeling, described earlier.
The cost of an FTE is $100,000 per year. Thereby, data modeler cost per year is $100,000.
Total savings is 2.8 FTE or $280,000

# **METHOD 2** PERCENTAGE OF SAVINGS IN DEVELOPMENT

This method uses development tasks that are already defined as part of the SDLC (system development life cycle) or the project management system. The first step is to assess the cost of each task both with and without data modeling. Don't forget intangible costs, which may require you to ask for estimated values. These may be self-reported so it is important to ensure they are reliable. The same task might be performed by multiple roles. Thereby consider each role when addressing the tasks and determine the cost of each role. Then determine the time demand for each task. Finally, calculate the total for each task by multiplying cost for each role by the task time.

Do not be concerned about precision in numbers. For example, recording all monetary amounts with two decimal points will be of no added value especially if the numbers are not accurate in the first place. The same is true for using percentages with multiple decimal points.

## EXAMPLE 1: SAVINGS IN DEVELOPMENT DOLLARS BY DEVELOPMENT TASK

The following example determines savings by development phase. Such an approach is useful for planned and historical projects. As mentioned earlier, this analysis can be done in different units of measure, such as money, time, etc., and at different granularities, such as by project or project task.

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 4.

It is reasonable to expect with data modeling a savings of from 1 to 10% in development costs. Savings of 10% or more are not unreasonable during certain phases, such as the database design phase or during development. Based on a solid logical data model, database design can be completed in anywhere between several hours and several days, a considerable savings over methods that do not use a data model.

The example in Table 4 (which is located on the next page and 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% in the task for Application Development & Test. The overall savings was 23.5%.

# QUANTIFYING DEVELOPMENT SAVINGS

| Development Task | Without Data Modeling | With Data Modeling | Savings Amount | Proposed Savings |
|---|---|---|---|---|
| Requirements | $55,000 | $52,250 | $2,750 | 5.0% |
| Technical Design | $100,000 | $30,000 | $70,000 | 70.0% |
| Application Development & Test | $90,000 | $54,000 | $36,000 | 40.0% |
| Maintenance (5 years) | $375,000 | $337,500 | $37,500 | 10.0% |
| **TOTAL** | **$620,000** | **$473,750** | **$146,250** | **23.5%** |

In doing these calculations on development, include maintenance tasks since in this approach maintenance is considered part of development. The reason for this is that so often projects may make short-term decisions during the development phases that postpone errors into the maintenance phase. Every project must decide on its priorities. The two general trade-offs are:

1 Use a disciplined approach based on sound architectural principles for benefits and cost control that endure for the life of the system.

2 Use a short-term "crunch mode" approach which simplifies meeting a pending deadline but will entail more maintenance later.

True, compromises must sometimes be made.
However, the organization must decide on its overriding principles.

### EXAMPLE 2: SAVINGS IN DEVELOPMENT DOLLARS ACROSS MULTIPLE PROJECTS

Essentially, this is the same method as Example 1 above, except that it uses a gross number for development costs as the basis, as shown in Table 5. Maintenance cost is not included in this spreadsheet. The definition of other terms and the formulas for calculation are the same for Example 1.

### OTHER IMPORTANT TERMS

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

**Development Cost** is the estimated cost to construct the system, taken from the AFE (Authorization for Expenditure) or Budget.

**Development Savings** is the dollar reduction in development resulting from the use of data modeling. This savings should be directly to data modeling. It is computed by multiplying the Development Cost by the proposed Savings %. Table 5 shows how we can use these costs and savings to determine the ROI as a percentage of development costs.

## CALCULATING ROI BASED ON 1% DEVELOPMENT SAVINGS

| 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%** |

## METHOD 3 PERCENT SAVINGS OF MAINTENANCE

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

The challenge, as always, is tangibility. Somehow, the percent of savings must be quantified. An acceptable way is to consider it as a percentage of maintenance. This will require negotiation with executive management. At the low end, 1% may be the best you can get. True value may be as much as 10%.

### 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 money spent on development and maintenance is an available number within an organization such as in the budget or AFE (Authorization for Expenditure) process.

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 3 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, based on self-reported data, but it is not pure magic.

In Table 6, some numbers (Proposed Savings %, Project Name or ID, Modeling Cost and Maintenance Cost) are given by the user. The identified 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.

**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 %.

## CALCULATING THE ROI FROM 1% MAINTENANCE SAVINGS

| 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%** |

# SUMMARY

Data modeling can provide tangible economic benefits. These are best shown by quantifying the traditional benefits of data modeling. Several types of savings can be used, such as savings in development and reduction in maintenance. Classical financial methods, such as return on investment, net present value and payback, resonate well with business management. Analyzing the economic value of data modeling will not weaken industry's commitment to it.

In fact, the value of data modeling can sometimes be surprisingly dramatic. On the contrary, analysis will strengthen data modeling by exposing the true and continuing economic contribution of data modeling to improved delivery of agile, profitable business applications. New technologies, such as agile development, NOSQL, column-oriented databases and big data, create new opportunities for the use of data modeling.

# APPENDIX

## DATA MODELING AND REVENUE ENHANCEMENT

One form of ROI is revenue enhancement. In and of itself, data modeling does not create revenue. Effective use of it in systems does. Data modeling enables better use of data for decision-making. The biggest business opportunity for revenue generation is with analytical data, not operational data. Implementing a new OLTP system will not, in and of itself, generate more revenue. It can reduce costs and introduce other operational efficiencies. However, the use of data modeling on BI and Big Data projects can generate significantly more revenue by focusing on new customer, product and marketing strategies. Experience has shown that an organization can sometimes generate enormous increases in revenue by redefining its products (and product relationships) or by re-examining its customer market segments. To do a data warehouse, BI or Big Data system without the use of the appropriate data modeling approach is imprudent. Having the data in the right structure is indispensable in this process. This structure does not have to be relational and the data model does not have to be rigid. BI in particular would be immeasurably more difficult without data modeling. Data modeling has been universally considered indispensable in projects such as business intelligence and data warehousing.

## 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 lifetime** 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 3-10%.
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. It can be very effective for justifying agile or incremental delivery of systems. 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 + Discount Rate)$^2$ + (Year 3 Benefit - Cost) / (1 + Discount Rate)$^3$ . . . (Year n Benefit - Cost) / (1 + 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.**

### 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.

Kim Medlin, "Prove It! Verifying the Economic Worth of Data Modeling with ROI Analysis," presented at Data Modeling Zone, October 5, 2015.

## ABOUT THE AUTHOR

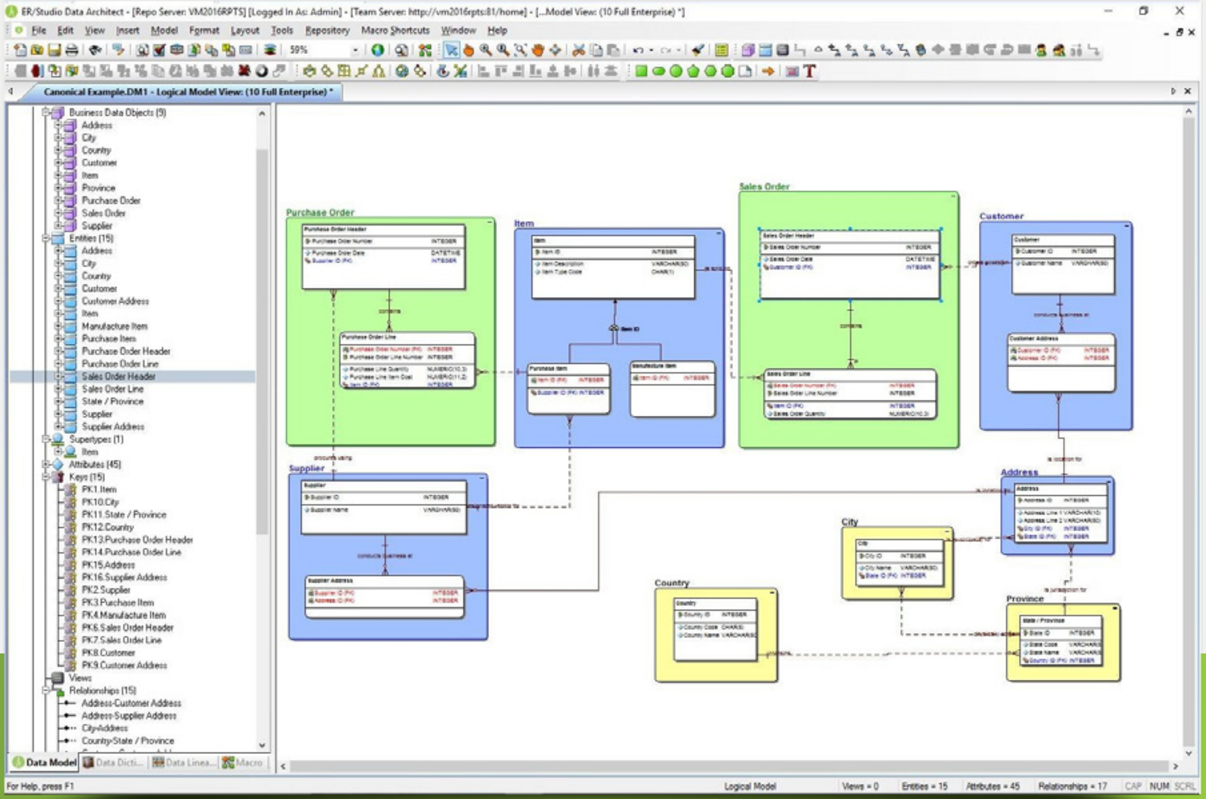
Tom Haughey is considered one of the four originators of Information Engineering in America. He has been an independent consultant and educator in data management since 1983. He is regarded as an exceptional teacher. For over three decades he has been delivering successful data management solutions in the area of data modeling, data warehousing, business intelligence, master data management, information architecture, database, 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. He can be reached at tom.haughey@InfoModelUSA.com.

# ER/STUDIO DATA ARCHITECT

## Model, Analyze and Optimize Enterprise Data

- Create effective models to build a business-driven data architecture
- Document and enhance existing databases to reduce redundancy
- Implement naming standards to improve data consistency and quality
- Effectively share and communicate models across the enterprise
- Map data sources and trace origins to enhance data lineage

**Download Here**



IDERA

IDERA.com