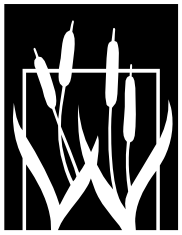

Whitemarsh
Information Systems Corporation

Return on Investment (ROI)

Data Centered Development and Management

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 301-249-1142
Email: Whitemarsh@wiscorp.com
Web: www.wiscorp.com

Table of Contents

Whitemarsh ROI Savings Summary

Traditionally developed business information systems developed through process-driven techniques cost significantly more than those created through data-driven techniques. The conservative estimate is that process-driven business information systems take 4.6 times longer to develop because the quantity of work products is 4.6 times larger.

Increased costs are not the sole problem. Business Information System bloat greatly increases maintenance time and cost, reduces efficiency, and because of the 4.6 greater work products factor, the ability to easily and effectively integrate, interoperate, and have consistent semantics is compromised.

The Whitemarsh data-driven techniques not only greatly reduce Business Information System bloat and maintenance, but also increase integration, interoperability and consistency. In addition to all these benefits, the Whitemarsh approach has an ROI of 28:1. This is a 3x silver bullet.

Supporting Links	
Link Area	**Link**
Enterprise Data Management Areas	http://www.wiscorp.com/roi_datacentereddevelopmentandmanagemententerprisedatamanagement.html
The Data Administration News Letter Articles	http://www.wiscorp.com/roi_datacentereddevelopmentandmanagementtdan.html
Short Papers	http://www.wiscorp.com/roi_datacentereddevelopmentandmanagementshort_papers.html
Clients	http://www.wiscorp.com/roi_datacentereddevelopmentandmanagementclients.html

1.0 Issue

Traditionally developed business information systems developed through process-driven techniques cost much more than those created through data-driven techniques.

The fall-back strategy for enterprises is to “buy packages.” That however just transforms the problem from determining a solution precisely fitted to the enterprise to one of buying somebody else's already determined solution.

Another alternative is to adopt the “Agile” approach to development. While the “Agile” approach has excellent characteristics regarding teamwork and rapid work product development,

it focuses on the excellent development of "individual trees" without regard to the overall architecture of the "forest."

A study which compared the development approach to two relatively large scale business information systems in which the two different business information systems were developed through process driven techniques resulted in 4.6 times more work products to be developed into business information system components than were determined to exist under data-driven techniques.

In this study, the teams and projects were held constant. The only variable was the development approaches: process-driven and data-driven. Both approaches were main-stream and widely used. Both were explicitly defined to ensure faithful execution. At the end, the work products from these two different approaches for the two different projects were counted and their ratios computed. Data-centered won, hands down.

2.0 Solution Approach

The data-centric solution approach includes the development of the following key work products:

- Mission Model
- Database Domain Model
- Database Object Model
- Data Element Model
- Function Model
- Organization Model
- Business Information System Generation
- Cycles of Requirements Iterations
- Requirements Model

Mission Model. The mission model consists of identifying and then describing the essential missions of the enterprise. The result can be 20 to 40 pages, and these become the overall architecture within which all projects must exist. The mission statements become short, hierarchically organized paragraphs that express the enterprise's mission in an ideal form. A Holy Grail "target" of long range accomplishment. The critical test of whether a topic is included is whether the enterprise can effectively exist without the topic. If yes, then the topic not needed. If not, then the topic is essential.

Database Domain Model. Database domains are noun-intensive paragraphs of sentences formed from the leaves of the Mission Model. If a database domain paragraph contains sentences that

describe complex objects then they must be subdivided into subordinate database domain paragraphs.

Once completed across all the Mission Model leaves, the database domain paragraph sentences are transformed into Subject-Entity to Relationship to Object-Entity statements. Here every noun becomes an entity no matter how simple or complex. Once the Entity-Relationship statements are complete, they are combined into larger scale diagrams, each corresponding to the database domain paragraphs.

At that point, the nouns are all triaged into complex nouns that become Database Object Classes, business Data Elements that represent single valued business facts, and property classes that become policy-homogeneous collections of single-valued data elements.

Database Object Model. Ultimately constructed is a higher level data diagram of just Database Object Classes and their relationship to other Database Object Classes. Property classes become candidate tables that ultimately are allocated to Database Object Classes. These data elements within each property class become a business data element.

Data Element Model. Data Elements become a non-redundant set of business facts that become semantic templates for database table columns. These come from the triage exercise and also the data elements from within the property classes.

Function Model. The Mission model is again used as the basis for a verb-based model that describe the essential business processes that must occur across the enterprise. The goal of the function model is to validate the data models, that is, Database Domain, Database Objects, and Data Elements. If there are business essential functions without corresponding data models then the data models are insufficient. Similarly, if there are data models without corresponding essential function models then the function model is incomplete.

Organization Model. The organization model flows from identifying the various enterprise business units involved in the specification of the function model.

Business Information System Generation. The real proof of subsets of the data model and subsets of the function model is a generated business information system. The functional subset becomes the major to minor menu items.

The subset data model is created by engineering columns for each property class that is transformed into a database table. The scope of the database table merely has to be primary key, foreign keys, and a few important columns that clearly set out the purpose of the table within the scope of its host database object.

Cycles of Requirements Iterations. The data model is defined to the business information system generator and from that, the prototype business information system is generated. After a reasonably small quantity of changes to the generated business information system it is demonstrated to the stake holders. Their comments typically result in both data and process changes. The data models are transformed and the business information system is regenerated. The stake holder demonstration is performed again.

This requirements iteration process is repeated four or so times. Once stable, work begins on the next functional subset, an proceeds until all the functional subsets are represented by business information system prototypes. The final set of business information system requirements become the specifications that are stored in the Metabase.

Requirements Model. The purpose of the Requirements Model is to act as a common collector of all the requirements that surface during the development of the models above. In addition to being a collector, the Requirements Model supports the interrelationship of all the other models based on their common requirement specifications. Finally, the Requirements Model enables the direct viewing of the object that is the subject of the requirement.

Data Centric Development Summary. The overall objective of this data-centric development is to arrive at an enterprise data model in third normal form where each table consists of only the primary key that addresses granularity and precision, and enough database table columns to validate scope and purpose. This overall process is quick, effective, and becomes the foundational blueprint for the entire enterprise.

3.0 Solution Engineering

The engineered solution includes highly engineered data models that holds the work product specifications of the enterprise. The included data models are:

- Mission Model
- Function Model
- Organization Model
- Database Objects Model
- Database Model
- Business Information System Model
- Requirements Model

The **Mission, Organization and Function** models are each represented by a few Metabase System database tables that can represent both hierarchical structures and also networks. The relationship among these three models is many-to-many. This avoids redundancy. The Mission,

Organization and Function models are able to be related to various other Metabase System components affected by the unique combinations of functions performed by organizations in the accomplishments of enterprise missions.

The **Database Object Model** consists of a collection of database tables for database object, database object table, database object table process, database object state, and database object information system.

The **Database Model** consists of a number of Metabase tables that contain the specifications for Schema, Table, Column, Primary Key and Primary Key Column, Foreign Key and Foreign Key Column, and SQL Data Type. The three distinct but interrelated layers of data models in addition to a data element model are: Concepts Model, Database Logical Model, and Database Physical Model. There is also the SQL View Model and XML Model.

The **Business Information System Model** consists of a number of Metabase System tables that store the specifications for the business information system, tables for its characteristics, subsystems, and tables for mapping to Views and ultimately to database table columns.

The **Requirements Model** consists of a number of Metabase System tables that store the specification of a requirement, the possibility of hierarchical and network relationships among requirements, and finally the relationship between a requirement from within a requirement's structure and 14 different Metabase System components including for example, mission-organization-functions, business events, business information systems, data elements, and resource life cycle nodes.

4.0 ROI

The cost estimate employed in the Information Systems Planning is based on a data-driven approach. That is evidenced on the fact that the quantity of business information system development function points is derived directly from the quantity of database tables: 80 function points of business information system development per database table.

Under process-driven techniques, no such metric is available because a process-driven approach has no strategy for determining a fixed work product upon which the critical business information system development steps of requirements, design, implementation, testing, and documentation begin.

Based on these two studies, and based on the multitude of studies accomplished by the United States' Government Accountability Office, the time and resources required for process-driven

approach business information system development must include work 4.6 times the number of work products than through the data driven approach.

4.1 Traditional Calculation

Given a 200 tables for a typical business information system, the quantity of Function Points at 80 function points per tables for business information system functionality is 16,000. Without a business information system generator and a cost of $200 per function point implementation, the cost for the first iteration of the business information system would be $3.2 million.

Without a business information system generator that enables completely realistic prototyping, the resulting business information system is likely to change by 25% for four cycles before quieting down. Thus the cost is another 100% for a total of $6.4 million.

If the methodology is process driven however, then the cost increases by a multiplier of 4.6. That brings the overall cost to just over $29.5 million.

4.2 Changed Approach Calculation

With the use of a business information system generator, and with the use of the data-driven approach, the costs drop dramatically. The function point cost for implementation drops to just about $50 per function point. So the initial system costs drops from $3.2 million to $800K.

There are fewer cycles of business information system evolution because the first production implementation is version 5 of 6. So, at three cycles of at most 10% changes, the additional costs are an additional 30%. That brings the cost to $1.04 million ($800K + $240K).

Finally because of the data driven approach there is no 4.6 multiplier. The final system cost is thus remains $1.04 million.

4.3 ROI Summary

For a single business information system the traditional life cycle cost is about $29.5 million. When the approach is changed to be accomplished through a business information system generator and also through the data driven approach, the cost drops to $1.04 million. That represents a ROI of 28. That is 3x a silver bullet.

Organizations typically have 100 such systems and because of the data driven approach that produces a centralized enterprise data model that becomes the common data-center of different functional subsets of these business information systems, the cost is not just a straight line savings of 100 * $1.04 million versus 100 * $29.5 million. There is an extra savings because of the ability to eliminate redundancy, conflict, and the ability to re-use large functional areas of

business information system specifications because of the business information systems generator and because of the enterprise-wide Metabase System's integrated, interoperable, and non-redundant collections of work products.

Experience with building the Metabase System shows that there is a 20% overlap across all the Metabase System modules. That would reduce the $104 million cost for 100 such systems down to about $83 million. The traditional costs would be a straight line increase because of the inability to have integration and non-redundancy. Thus the traditional cost would be about $2.95 billion.

The cost savings by the change to data driven and to business information system generation is so great as to possibly be not believable. Hence the vast majority of the 100 business information systems developed under process-driven approaches are just strung out over many man years, abandoned, or their true costs hidden.