Whitemarsh
Information Systems Corporation

Our Challenge for Business Information System Success

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

1.0 Objective 1

2.0 Topics Covered 1

3.0 The Success, Challenged & Failure Statistics 1

4.0 The Knowledge Worker Framework Context for IT Errors 3

5.0 The Government Accountability Office's (GAO) Allocation of IT Errors 6

6.0 Error Elimination through Key IT Processes and Strategies 9

6.0 Key Work Products and Metabase System Cross Reference 13

8.0 Key Work Products and Data Architecture Reference Model Cross Reference 17

9.0 Our Challenge 20
- 9.1 The Challenge 21
- 9.2 Metabase System Projects 22
  - Enterprise Mission Architecture 22
  - Database Objects 23
  - Enterprise Data Architecture 24
  - Enterprise Data Elements 24
  - Resource Life Cycle Analysis 25
  - Concept Data Models 25
  - Database Models 26
  - Fact Name and Definition Management 26
  - Fact Semantics Management 27
  - Use Cases 27
  - Business Information Systems 27
  - Information Systems Plans 28
  - Reverse Data Engineering 28
  - Forward Engineering 29
  - Requirements 29

10. Conclusions 30

11. References 31

Figures

Figure 1. Interaction among Knowledge Worker Framework columns. 6
Figure 2. Traditional business information system development time-line by phase. 12
Figure 3. Iterative requirements development approach. 12
Figure 4. Cross reference between data architecture reference model components and business information system work products. 18

Tables

Table 1. Standish Project Benchmarks over the Years. 2
Table 2. Knowledge Worker Framework. 5
Table 3. Knowledge Worker Framework with Work Products. 7
Table 4. Allocation of Percents of Tallied GAO IT System Errors. 8
Table 5. Normalized Cost-to-Fix Estimates 9
Table 6. Metabase System modules with respect to Knowledge Worker Framework. 14
Table 7. Metabase System functional modules. 17
Table 8. Cross reference between work products and data architecture reference model component. 19
Table 9. Cross reference among work products. 20

1.0 Objective

The objective of this paper is to present the Whitemarsh approach to dealing with the very unacceptable business information system development statistics that are generated by the Standish Group every year.

This paper examines the possible reason for these dismal statistics and proposes a development approach that has been used by Whitemarsh with great success and by the 30,000 member strong Clarion community for a large number of years.

2.0 Topics Covered

The topics is this paper include:

- The Success, Challenged and Failure Statistics
- The Knowledge Worker Framework Context for IT Errors
- The Government Accountability Office's (GAO) Allocation of IT Errors
- Error Elimination through Key IT Processes and Strategies
- Key Work Products and Metabase System Cross Reference
- Key Work Products and Data Architecture Reference Model Cross Reference

3.0 The Success, Challenged & Failure Statistics

For years we have been reading about the statistics of failed, challenged, and successful IT projects. These statics have been popularized in the Chaos Reports from the Standish Group (http://www.standishgroup.com/).

Table 1 presents these statistics over time. There has been significant discussion about these statistics, and an excellent article about the validity and reliability of the statistics, *The Rise and Fall of the Chaos Report Finding*, is located at: (http://www.cs.vu.nl/~x/chaos/chaos.pdf).

While the article is interesting and does clearly point to issues related to the clarity, precision, and accuracy of the Standish Group's statistics, the article offers no help in figuring out why projects are successful, challenged, or failed. Or, more importantly, how to pro-actively prevent challenges or failures.

The article does, however hint at one of the reasons for the dramatic Standish statistics. This is provided in the "Rise and Fall" article's Figure 1. This figure shows that as the project nears completion, the estimates of the time and cost to achieve completion become more accurate. It's tempting here to say, "Well duh!" Unsaid, however is that what is more likely known is a clearer picture of the true requirements of the effort, and the amount of work, expressed in terms of costs that will be required to conclude the result. If more accurate requirements and amount of work were better know at the very start of the project, it's likely that the Standish statistics would have a much larger percent in the Success column and dramatically smaller percents in the Challenged and Failed columns.

Standish Project Benchmarks over the Years.			
Year	**Success**	**Challenged**	**Failed**
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

Table 1. Standish Project Benchmarks over the Years.

Analogously stated as to whether you, as a person, are either alive or dead, the article advocates a more refined strategy on identifying exactly when you become dead, but nothing on why you are dead or how to prevent your death. The article's solution to this problem seems merely to add more columns in between success and failure, and to move some of the percents amounts among these new columns. While that might be more interesting, it certainly does not help. After all, if you're dead, you're dead. What's really needed is knowing why you are likely to die, and taking steps to prevent your impending death.

What this TDAN article does, in contrast, is focus on the reasons why your project may be successful, challenged, or failed. Even more importantly, this article also sets out a strategy to dramatically move the failed and challenged percent amounts to success.

Key to understanding the Chaos statistics is the Knowledge Worker Framework. That is because through the Knowledge Worker Framework you can understand where the errors occur, which work products must be deficient, and what the down-stream implications are. This framework has been addressed indirectly in the following TDAN articles:

- 1998, Database Objects (http://www.tdan.com/view-articles/4268)
- 1998, Resource Life Cycle Analysis (http://www.tdan.com/view-articles/5045)
- 2000, A New Paradigm for Successful Acquisition of Information Systems (http://www.tdan.com/view-articles/4868)
- 2003, Frameworks, Meta-Data Repository & the Knowledge Worker (http://www.tdan.com/view-articles/5130)
- 2003, Comprehensive Metadata Management (http://www.tdan.com/view-articles/5133/)

- 2007, Information Systems Development (http://www.tdan.com/view-articles/6124)
- 2008, Data Semantics Management (http://www.tdan.com/view-special-features/8670)

In reviewing these articles, it became clear that the following has never been presented:

- The Knowledge Worker Framework itself.
- The allocation of GAO IT errors in terms of percents.
- A summary of the key IT processes that need to be accomplished to address these errors.
- The cross reference between key work products and the Whitemarsh Metabase System that can capture, store, interrelate, report, and evolve these work products.
- A cross reference between key work products as it relates to a comprehensive data architecture reference model.

This article not only presents these topics, it sets out strategies to both prevent IT business information system failure and to also prevent challenges from occurring in the first place.

4.0 The Knowledge Worker Framework Context for IT Errors

The Knowledge Worker Framework was created in response to a challenge. During a consulting assignment in the middle 1990s, a high-level manager indicated that a very popular framework looked promising and should be implemented across his IT organization. I indicated that it was too bad that the framework was not able to be successfully implemented, as specified, despite its obvious surface validity. He immediately asked why it wasn't able to be implemented. I responded by relaying how an attempt to implement the popular framework in a Federal Agency had failed for the following reasons:

- The various work products referenced in the cells were not sufficiently detailed.
- There was no cell to cell integration so that the work products could reference and employ each other.
- There was no framework-based detailed business information system methodology that could be followed.
- Metrics didn't exist across the work products that would support accurate effort estimates.

- There was no project, business function, or enterprise-wide work product database that could be used to capture, store, review, evolve, and accelerate other work products.

Almost immediately, the high-level manager asked that a valid framework be created and that its draft be delivered in a few weeks.

Rather than develop a new framework out of whole cloth, the framework was created inductively from an existing business information system's development methodology that had been time-tested and evolved for the prior 15 years. The methodology was fundamentally data-centric, had very detailed work steps, formally defined work products that were non-redundant, integrated, and interoperable one with the other, contained detailed metrics for work product estimation, and was supported by a multi-user work-product database that enable work product capture, interrelationship, reporting and evolution within projects, enterprise-functions, and across the enterprise itself.

The framework was set within the context of the knowledge worker as opposed to the manufacturing process worker. Hence, its name is the Knowledge Worker Framework. The overall architecture of the Knowledge Worker Framework is set out in Table 2. The architecture of the framework consists of six rows and six columns.

The rows, that is, the levels are essentially the same as the Zachman Framework. The six columns are divided into four sets: Mission, Machine, Interface, and Man. The Mission column of work products acts as the "Holy Grail" target of accomplishment. The Machine column work products are the computer-based mechanisms of reflecting Mission accomplishment. The Man columns are the human organizations and the human functions that, through the Machine assists, reflect Mission completion. The Interface column are the events, calendars, et al that set out the interactions between the Man column work products and the Machine column work products.

Work product creation generally proceeds from left to right for Mission, Database Object, and Business Information Systems. Thereafter, work product creation proceeds from right to left, that is, from Organization to Function. Finally, the Man and Machine work products are intersected through the Interface column work products. The reason for this left-right, right-left, and then intersection sequence is simple. The apolitical work is accomplished first and is the quickest and easiest to accomplish. Thus, almost immediately there are generated work products that can be reviewed and that have real and immediate value. Thereafter the Man column work products, which are certainly both political and style-based are accomplished and when done are interrelated with the already created Machine column work products. Work products across all six columns are related in a many-to-many fashion. This enables non-redundancy. Missions are expressed in the form of idealized "Holy Grails" outcomes without any hint of either who or how. That makes missions apolitical because they are not bound or constrained by any preconceived organizations and or human style-based strictures.

The first machine column, database objects, contains the work products for resource life cycles and mission-based database domains. It also contains all the traditional data models, which, in turn, also contain the appropriate processes tied to database object components. The database objects are tied to missions in a many-to-many fashion. Hence they too are apolitical.

<table>
<tr><th colspan="7">Knowledge Worker Framework</th></tr>
<tr><th rowspan="3">Level</th><th rowspan="3">Mission</th><th colspan="5">Man-Machine Interface</th></tr>
<tr><th colspan="2">Machine</th><th>Interface</th><th colspan="2">Man</th></tr>
<tr><th>Database Object</th><th>Business Information System</th><th>Business Event</th><th>Business Function</th><th>Organization</th></tr>
<tr><td>Scope</td><td colspan="6" rowspan="6">Identified and highly engineered work products that are captured, reported, and evolved during the execution of the business information methodology for one or more business information systems within projects, overall business functions, and across the enterprise. These work products are non-redundant, interrelated, are represented in the Metabase System’s database that is inherently multi-user and able to be deployed enterprise-wide.</td></tr>
<tr><td>Business</td></tr>
<tr><td>System</td></tr>
<tr><td>Technology</td></tr>
<tr><td>Deployment</td></tr>
<tr><td>Operations</td></tr>
</table>

Table 2. Knowledge Worker Framework.

The second machine column, business information systems are tied to database objects in a many-to-many fashion and exist solely to add/delete/modify the data contained in the database objects. Because business information systems are not tied to nor based upon organizations or human-based functions they are more long lasting and are largely devoid of any constraining styles.

The first-to-accomplish man column, organization, sets out the various organizations that are engaged in the accomplishment of the missions of the enterprise. This architecture supports multiple instances of essentially the same organization. “The” right one does not have to be chosen above all other equivalent organizations.

The second man column, function, describes either specially or generally the human functions that are performed by organizations regardless of whether these functions are ultimately assisted by “machine” work products. Like organizations, there can be multiple configurations of essentially the same set of human functions. An organization may perform different styles of essentially the same function. Similarly, a given function may be performed by multiple organization. Like organizations, “the” right function does not have to be chosen above all other similar functions.

The final column, Interface, relates human functions with their business information system assists within business events and various business accomplishment calendars.

The Knowledge Worker Framework columns, as shown in Figure 1, tell a story. The six columns are non-redundant, independent through many-to-many relationships, are integrated across the framework, and are detailed down through its rows of unfolding specification, implementation, deployment, evolution and operation.

The Knowledge Worker Framework work products captured, created, reported, and evolved across the six rows and columns of the cells are presented in Table 3.

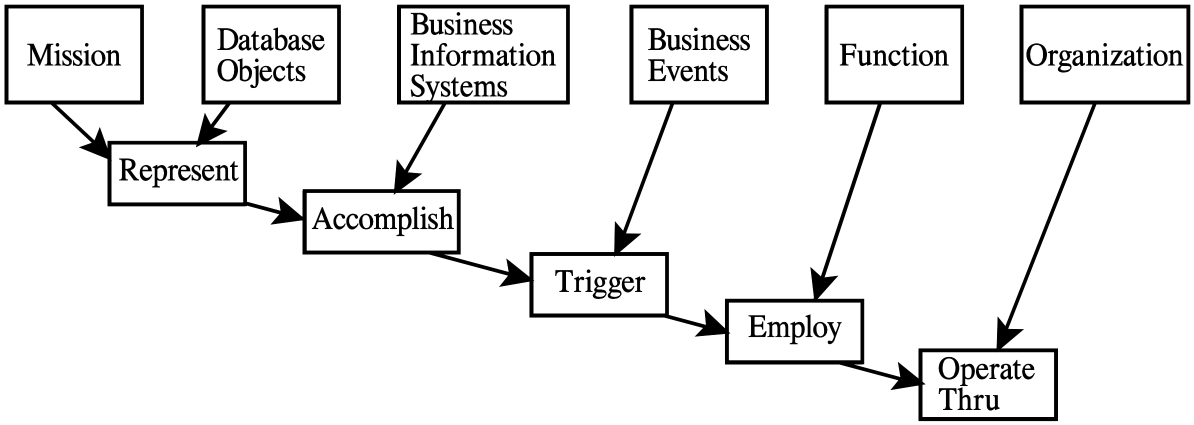

Figure 1. Interaction among Knowledge Worker Framework columns.

5.0 The Government Accountability Office's (GAO) Allocation of IT Errors

Shortly after the Knowledge Worker Framework was completed and presented to the high-level manager, he demanded to know why the Knowledge Worker Framework was any more valid than the one he initially asked about.

To respond to his question, Government Accountability Office reports of IT system disasters were reviewed. 13 different studies were selected mainly because they exceeded $100 million in cost. The reasons for IT failure were allocated to both the Knowledge Worker Framework and to the popular framework that was initially proposed. Only 10% of the GAO errors were addressed in the popular framework. In contrast, 100% of the errors were accounted for in the Knowledge Worker Framework. One reason for the discrepancy could be that the Knowledge Worker Framework was inductively created from an already proven and very detailed business information system development methodology. Table 4 presents the percent distribution of the GAO IT system errors across the Knowledge Worker Framework.

Several things are very interesting in this errors tabulation. First, 41% of the errors occur in the first two rows. These rows represent Requirements Analysis and Design. 50% of all the errors occur **after** a business information system is deployed. That is, within the Business Function and Organization columns for the System through Operations rows. Third, the actual percent of errors that occur during the actual IT development effort is just 5%.

These error percents go a long way to explaining the Standish Group's percents. If you have not successfully accomplished requirements analysis and design, how can you possibly know what is to be build, how complicated the building effort is to be, how much it will cost, and how long it will take? It's almost a "Duh" moment to conclude that on-time and on-budget projects are accidents.

Whitemarsh Knowledge Worker Framework							
		Man-Machine Interface					
		Machine		Interface	Man		
Deliverables	**Mission**	**Database Object**	**Business Information System**	**Business Event**	**Business Function**	**Organization**
Scope	List of Business Missions	List of Major Business Resources	List of Business Information Systems	List of Interface Events	List of Major Business Scenarios	List of Organizations
Business	Mission Hierarchies	Database Domains, and Resource Life Cycles	Information Sequencing and Hierarchies	Event Sequencing and Hierarchies	Business Scenario Sequencing, Hierarchies, and Use Cases	Organization Charts, Jobs and Descriptions
System	Policy Hierarchies	Data Elements, Specified Data Models and Identified Database Objects	Information System Designs	Invocation Protocols, Input and Output Data, and Messages	Best Practices, Quality Measures and Accomplishment Assessments	Job Roles, Responsibilities and Activity Schedules
Technology	Policy Execution Enforcement	Implemented Data Models and Detailed Database Objects	Information Systems Application Designs	Presentation Layer Information System Instigators	Activity Sequences to Accomplish Business Scenarios	Procedure Manuals, Task Lists, Quality Measures and Assessments
Deployment	Installed Business Policy and Procedures	Operational Data Models	Implemented Information Systems	Client & Server Windows And/or Batch Execution Mechanisms	Office Policies and Procedures to Accomplish Activities	Daily Schedules, Shift and Personnel Assignments
Operations	Operating Business	View Data Models	Operating Information Systems	Start, Stop, and Messages	Detailed Procedure Based Instructions	Daily Activity Executions, and Assessments

Table 3. Knowledge Worker Framework with Work Products.

Knowledge Worker Framework							
		Man-Machine Interface					
		Machine		Inter-face	Man		
Deliver-ables	Mission	Database Object	Business Infor-mation System	Business Event	Business Function	Organ-ization	Row Pct
Scope	5	2	3	1	3	4	18
Business	5	3	2	1	6	6	23
System	3	2	2	1	12	8	28
Technology	1	0	0	0	8	6	15
Deployment	0	0	0	0	5	5	10
Operations	0	0	0	0	3	3	6
Col. Pct	14	7	7	3	37	32	100

Table 4. Allocation of Percents of Tallied GAO IT System Errors.

In support of the disasters that follow from these GAO errors, Mark Vreeland wrote an article for TDAN entitled, Information Management & Governance Disciplines. (http://www.tdan.com/view-articles/5273).

While his article described the architectural and engineering disaster known to many as the Winchester Mystery House, the problems contained in the Winchester Mystery House have a direct connection to many of the IT systems described in the GAO studies. The errors, described in IT terms included, for example:

- No clear architecture design between application, technical and data architectures
- No metadata environment
- Multiple reporting tools
- Multiple enterprise resource planning (ERP) systems
- Multiple business intelligence tools
- No common data / information architecture
- Lack of standards for acquiring architecture components
- Business unit driven solutions with no consideration given to the enterprise architecture
- Incomplete architecture design for migrating data from operational environment to analytical environment
- Lack of standards for data integration
- Multiple systems for the same business function (i.e., two accounting systems, two payroll systems, multiple claims processing systems)
- No overall governance framework that encompasses the enterprise

6.0 Error Elimination through Key IT Processes and Strategies

From the prior sections, it was established that 41% of IT system errors occur during the first two phases of business information development. It is also presumed that errors that occur early in the cycle that go undetected have a significant impact on the overall cost of a project. A Google search produced a NASA paper that summarized a collection of studies that computed the overall increase in costs for fixing IT system errors based on when they were discovered. The NASA paper is located at:

http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100036670_2010039922.pdf.

The table from the study is presented in Table 5.

Statistics Source and Year	**Phase Requirements Issue Found**			
	Requirements	**Design**	**Code**	**Test**
J Boehm, 1981	1	5	10	50
Hoffman, 2001	1	3	5	37
Cigital, 2003	1	3	7	51
Pavlina, 2003	1	10	100	1000
McGibbon, 2003		5		50
Mean	1	7.3	2 5.6	177
Median	1	5	10	50.5

Table 5. Normalized Cost-to-Fix Estimates

Elimination of errors in the first the requirements and design phases will have the greatest positive benefit on project costs and expended time. A key objective then is to eliminate these errors at the earliest possible time. That would clearly be in the creation of the work products that are created within the first two rows of the Knowledge Worker Framework.

The six key processes that occur across the Knowledge Worker Framework along with the citation of their rows, columns, and GAO percents are:

- **Building and Employing Enterprise Architecture Models**:
  KWF: Scope and Business Rows, all columns. About 41% of all GAO IT errors.

- **Creating and Evolving Information Systems Plans**:
  KWF: System Row, all columns. About 8% of GAO IT errors.

- **Architecture and Engineering of Data Models**:
  KWF: Technology Row, database object column. Less than 2% of Errors

- **Performing Reverse Engineering of Legacy Systems and Databases**:
  KWF: Operations, Deployment, Technology and System Rows. Database Objects Column. Less than 2% of Errors

- **Forward Engineering Manufacture of New Systems and Databases**:
  KWF: Operations, Deployment, Technology and System Rows. Database Objects Column. Less than 2% of Errors

- **Employment Errors**:
  KWF: Systems through Operations Rows. Operations and Functions Columns. About 49% of Errors.

The focus clearly has to be on the first key process, Building and Employing Enterprise Architecture Models. An interesting problem however occurs during the development of the first two rows of work products. How do you know the work products are correct? While reviews are important, the most credible source of proof is implementation. When errors are observed during implementation, the work products of the first two rows can be corrected and the work products of the remaining rows can begin anew. However such a process can be very expensive because by the time a credible amount of a business information system is implemented too much time and resources have passed and been expended.

The focus clearly has to be on the first key process, Building and Employing Enterprise Architecture Models. An interesting problem however occurs during the development of the first two rows of work products. How do you know the work products are correct? While reviews are important, the most credible source of proof is implementation. When errors are observed during implementation, the work products of the first two rows can be corrected and the work products of the remaining rows can begin anew. However such a process can be very expensive because by the time a credible amount of a business information system is implemented too much time and resources have passed and been expended.

A strategy to accomplish, however, without expending even a few percent of the overall time and resources of a significant project is through iterations of:

- High level data modeling.

- Business information system generation based on the data models.

- Hands-on demonstrations with business-functional users.

- Subsequent revision of work products, high-level data models, and generated business information systems.

These steps are recycled until the functional users are satisfied that the high-level data models, generated business information systems, and functional behavior models meet their expectations and of course, fulfill the enterprise's missions.

The work products from the first two rows provide a great assist in this effort. The database domains from the database objects column contain sufficient information to support the quick development of first-cut data models. These data models can be easily exported into a data definition language form that can be read, for example into the Clarion (www.SoftVelocity.com) business information system generator. Clarion, armed with a data model, Clarion easily and quickly generates a first-cut version of a business information system for a particular business function. The work products from the business function column provide the behavior model that can be used to electronically trim/style the generated business information system. The organizations identified in the last column provide the ready-to-review audiences.

What is being reviewed through the use of the Clarion generated business information system generations are the missions, first-cut database object tables, business information systems, and business functions. After a demonstration, feed back is obtained and the work products in the first two rows are adjusted. Since there is virtually no effort expended in the development of the prototypes, they can be discarded and regenerated. What results is from three to five evolutions of the overall architecture that is thereafter implemented. By cycling this over several months, the following benefits are manifest:

- Significantly refined mission, organization, and function models.
- Good approximations of ready-to-implement data models
- Good approximations of the type, quantity, and complexity of the ready-to-implement business information systems.
- A collection of enterprise architecture models that are at Version 4 or 5 versus an untested version 1.

These iterated work products can be used to dramatically increase the accuracy and quality of business information system implementation efforts. Figure 1 from the Rise and Fall paper shows that the forecast and actual ratios do not trend towards 1.0 until about 60% into a project. From this paper's Figure 2, that would be about month 15 of a 30 month time line. That's well into implementation, and by that time the majority of the funds have been expended. Worse even more, all the time and effort has been expended on a Version 1, rather than on a Version 4 or 5.

In contrast, Figure 3 illustrates the iterative requirements development approach. The time lines are generally the same. What is critically different is that through the traditional approach the first operating version of the system is Version 1 and through the iterative approach, the first operating version of the system is Version 5.

That will largely eliminate the 41% of the first two rows of errors. These errors are eliminated because the functional uses will be reviewing and reacting to actual business information system executions (although not production class) instead of reviewing documents and drawings.

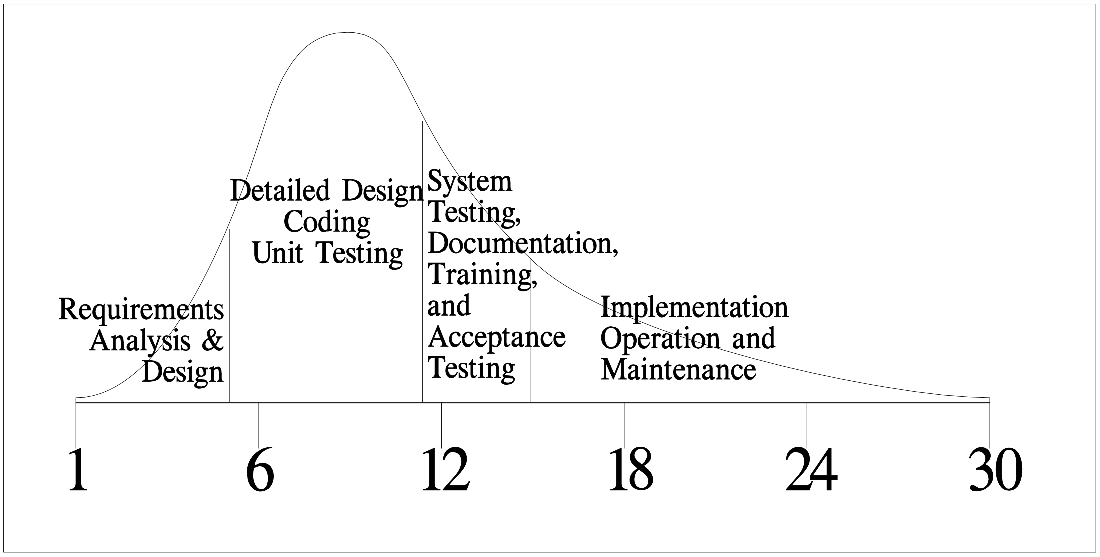

Figure 2. Traditional business information system development time-line by phase.

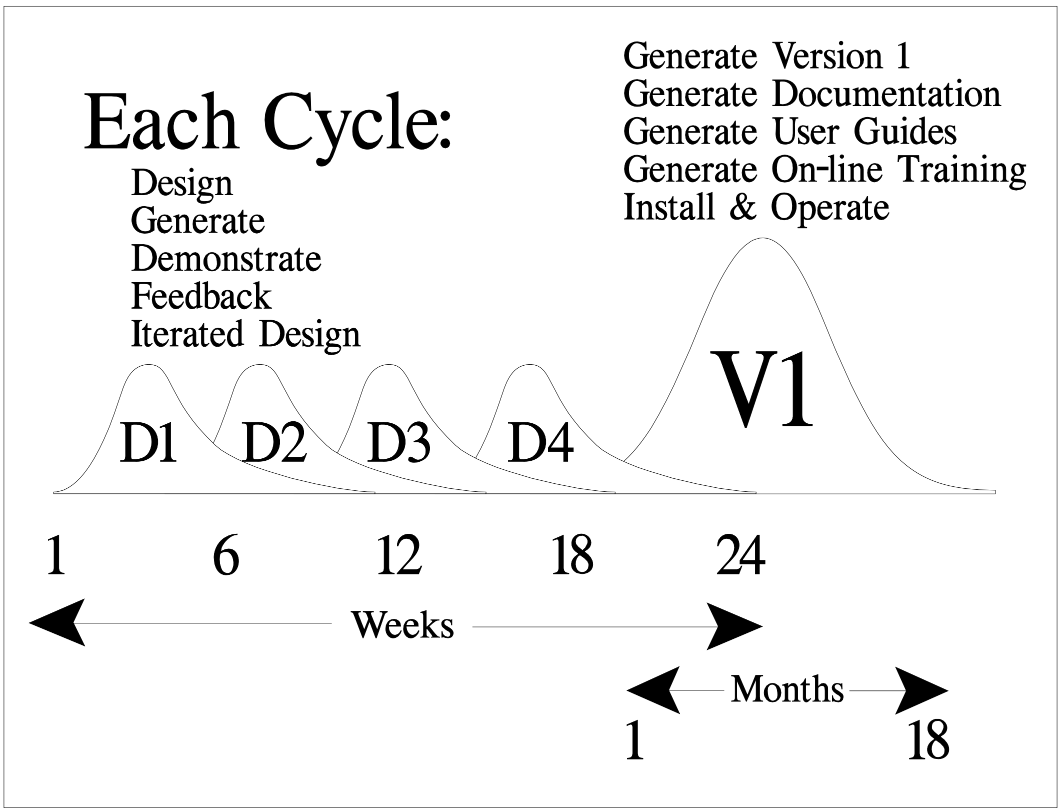

Figure 3. Iterative requirements development approach.

As to the other and more significant class of errors, that is those that occur after the business information system is deployed, they too will be largely eliminated because these very same functional users will be interacting with the evolving system during each of the design iterations. This will give a good heads-up to those persons charged with changing the work products such as organization charts, best practices, activity sequences, procedure manuals and the like. These changes will largely be known at the end of 18 to 20 weeks during the iterative requirements approach. Under the traditional approach, these changes would not surface until about 18 months after project start.

6.0 Key Work Products and Metabase System Cross Reference

A very critical and key component of this iterative requirements approach is the existence of the metabase system and its database of work products. These work products are identified in the Table 3, Knowledge Worker Framework. The metabase system functional modules that enable these work products to be captured, reported, and updates are set out in Table 6. The Table 7 contains brief descriptions of the Metabase System modules:

Metabase Software Module	Knowledge Worker Framework					
	Mission	Database Objects	Business Information System	Business Event	Business Function	Business Organization
1. Mission, Organization, Function Position Assignment	✔			✔	✔	✔
2. Resource Life Cycles		✔	✔			
3. Document			✔		✔	
4. Form			✔		✔	
5. Information Needs Analysis					✔	
6. Requirements Management		✔	✔		✔	✔
7. Use Cases		✔	✔		✔	✔
8. Business Information Systems		✔	✔		✔	
9. Data Elements		✔	✔			
10. Specified Data Model		✔	✔			
11. Implemented Data Model		✔	✔			
12. Operational Data Model		✔	✔			
13. View Data Model		✔	✔			

Table 6. Metabase System modules with respect to Knowledge Worker Framework.

Metabase Module	Description
Administrative Module	The Administrative module of the metabase system supports the creation of new metadata databases. Part of the creation of a new metabase database, is the loading of default values. Included as well are the creation of users with names and passwords. These new users are mapped to metabase modules and metabase databases. This enables a level of security. The metabase administrator can terminate a user's access to a given metabase database, and can delete a metabase user.
Business Information Systems	Business Information Systems are the necessary computer software systems triggered by enterprise business events instigated by functions. Business information systems are directly related to mission, organization, function, and databases. Business information systems are interrelated to each other including their calendar and business event execution schedules? Collectively, business information systems are the mechanisms that carry out

Metabase Module	Description
	the automation aspects of enterprise policy.
Data Element	Data elements are the enterprise facts that are employed as the semantic foundations for attributes of entities within data models of concepts (Specified Data Models), columns of tables within database models (Implemented Data Models) that support the requirements of business and are implemented through database management systems (Operational Data Models), that, in turn, are employed by business information systems (View Data Models) that materialize the database objects necessary for within the resources of the enterprise that support the fulfillment of enterprise missions. Data elements are derived through the specification of concepts, conceptual value domains, data element concepts, and value domains. Additionally, the data element model supports the definition of a full complement of semantic and data use modifiers that can be allocated to data element concepts, data elements, attributes, and columns. Because of these allocations, naming and definitions can be fully automatic.
Database Objects	Database Object Classes represent the identification of 1) the critical data structures, 2) the processes that ensure high quality and integrity data within these data structures, 3) the value-based states represented by these data structures, and 4) the database object centric information systems that value and transform database objects from one state to the next. Database Objects are transformed from one valid state to another in support of fulfilling the information needs of business information systems as they operation within the business functions of organizations. Database objects are encapsulated within database management systems (DBMS) so that they can be independent of any end-user environment or programming language.
Documents and Forms	Documents and Forms enable the structuring of both documents and forms into documents/forms, their sub-sections, and cells. Documents and Forms can be interrelated with network data structures. Both Document and Form section cells can be mapped to mission-organization-functions, and also to view columns.
Implemented Data Model	Implemented Data Models, are the data models of databases that are independent of DBMSs. These models, which are comprised of data structure components: schema, tables, columns, and inter-table relationships. Relationships are restricted to tables within a single schema. While each implemented database data model can address multiple concept data models from the collection of concept data models, each implemented data model should address only one broad subject. Every table column should map to a parent Attribute. Every column can be allocated both semantic and data use modifiers. There is a many-to-many relationship between the Specified Data Model and the Implemented Data Model.
Information Needs Analysis	Information Needs Analysis represents the identification, definition, and interrelationship of the information needed by various organizations in their functional accomplishment of missions and what databases and information systems provide this information?
Mission, Organization, Function, and Position Assignments	The Mission, Organization, Function, and Position Assignment module represents the identification, definition, and interrelationships among the persons who, through their positions, perform functions within their organizations that accomplish enterprise missions. Missions define the very existence of the enterprise, and that are the ultimate goals and objectives that measure enterprise accomplishment from within different business functions and organizations? Functions represent the procedures performed by enterprise organization groups as they achieve the various missions of the enterprise from within different enterprise organizations? Organizations represent the bureaucratic units created to perform through their functions the mission of the enterprise. Position Assignments

Metabase Module	Description
	represent the identification of both persons and position titles that are both interrelated and allocated to the various functions performed by organizations that achieve enterprise missions.
Operational Data Model	Operational Data Models, are the data models of databases that have been bound to a specific DBMSs. These models, which are comprised of data structure components: DBMS schema, DBMS tables, DBMS columns, and inter-table DBMS relationships. DBMS Relationships are restricted to DBMS tables within a single DBMS schema. While each operational database data model can address multiple implemented data models. Every DBMS Column should map to a parent Column. There is a many-to-many relationship between the Implemented Data Model and the Operational Data Model.
Requirements Management	The purpose of this Metabase System module, Requirements Management, is to provide: ● Identification and description of requirements. ● Interrelationship among different requirements. ● Relationship between requirements and other metadata artifacts. ♦ Mapping to Business Events ♦ Mapping to Business Information Systems ♦ Mapping to DBMS Columns ♦ Mapping to User Acceptance Test Steps ♦ Mapping to Database Objects ♦ Mapping to User Cases ♦ Mapping to Data Integrity Rules ♦ Mapping to Resource Life Cycle Nodes ♦ Mapping to Mission Organization Functions The Requirements Management module permits recording of the characteristics of the requirements associated with the enterprise. Each requirement can be described and interrelated with other requirements. Once requirements are identified and described, they can be allocated to any of the “mapping” items above. This permits enterprises to know the requirements needed by whom within the different organizations in the performance enterprise missions.
Resource Life Cycle Analysis	Resource Life Cycle of Analysis identifies, defines, and sets out the resources of the enterprises, the life cycles that represent their accomplishments, and the interrelationships among the different enterprise resource life cycles. Resource life cycle nodes represent the end-state data resulting from the execution of business information systems. The end-state data is represented through database object classes.
Specified Data Model	Specified Data Models are the data models of concepts. These models, which are comprised of data structure components: subjects, entities, attributes, and inter-entity relationships. Relationships can span interrelate entities within multiple subjects. Each data model should address only one concept such as a person’s name, or an address, etc. These concept data models can then be templates for use in developing database models (Implemented or Operational). Every entity attribute should map to its parent Data Element. Every attribute can be allocated both semantic and data use modifiers.
Use Cases	The use case module enables the creation, updating, and interrelationships of detailed function model specifications. The use cases contain use case networks structures, use case events, pre conditions, post conditions, special conditions, use case facts, and use case actors. These use case components are able to be interrelated with business information

Metabase Module	Description
	systems, database table columns, and mission organization functions, and persons functioning with positions.
View Data Model	The View data model represents the interfacing between operational data models and business information systems. View and their view columns can be characterized as Input and/or Output. Additionally, these views can be mapped one to the other on a view column basis and processes can be specified to define any appropriate data value transformation.

Table 7. Metabase System functional modules.

8.0 Key Work Products and Data Architecture Reference Model Cross Reference

Figure 5 provides a cross reference between the data models contained in the data architecture reference models. From this Figure it should become very clear that data models are the ultimate "associative" entity among work products. Tables 8 provides a cross reference between data architecture data reference models and other work products. Tables 9 provide cross references among the work products that are not part of the data architecture reference model.

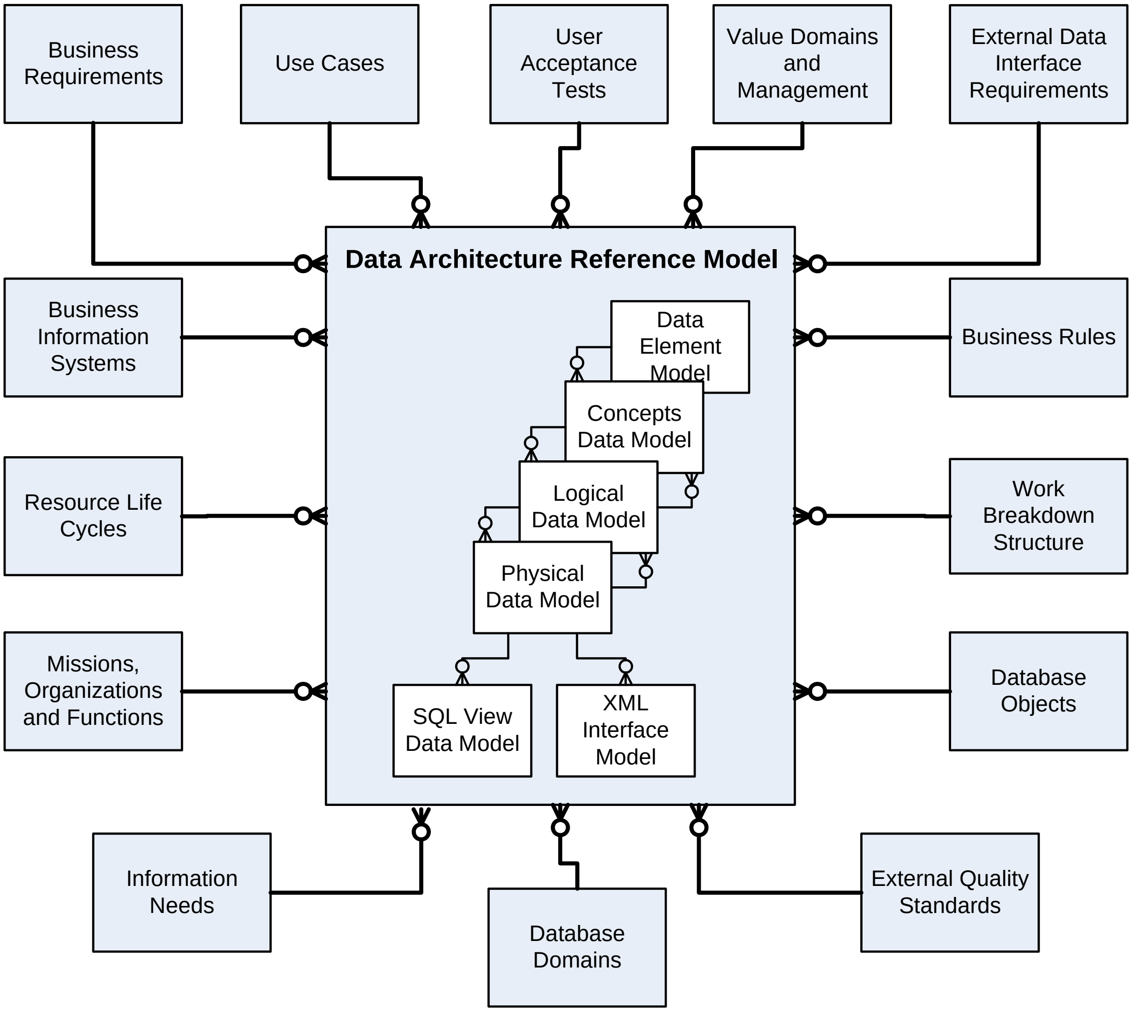

Figure 4. Cross reference between data architecture reference model components and business information system work products.

Work Products	Data Architecture Reference Model Component					
	Data Element	Concepts Data Model	Logical Data Model	Physical Data Model	View Data Model	XML Data Model
Business Information Systems				✔	✔	✔
Business Requirements		✔	✔	✔		✔
Business Rules	✔		✔	✔	✔	✔
Database Domains			✔			
Database Objects			✔			
External Data Interface Requirements				✔	✔	✔
External Quality Standards	✔	✔	✔	✔	✔	✔
Information Needs	✔		✔			
Mission Organization Functions			✔			
Resource Life Cycles			✔			
Use Cases			✔			
User Acceptance Tests				✔	✔	✔
Value Domains and Management	✔	✔	✔	✔	✔	✔
Work Breakdown Structure (WBS)			✔	✔		✔

Table 8. Cross reference between work products and data architecture reference model component.

From Figure 4 and Tables 6 and 7 it should become abundantly clear that virtually all work products are related to a number of other work products. Work on one work product clearly involves and affects others. If work products are accomplished in isolation, it is virtually assured that there will be conflicts, disconnects, redundancy, and specifications that are out of date, one with the other. Such problems cannot but help complicate and possibly even cause business information system projects to fail, be late, have conflicting data, business rules, and the like. It is very likely that these types of problems contribute to the Chaos statistics.

Work Products	Backward Reference from Involved Component to Column 1 Item													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1. Business Information Systems	na	✔	✔		✔	✔	✔	✔	✔	✔	✔	✔		✔
2. Business Requirements	✔	na	✔	✔		✔	✔			✔	✔	✔	✔	✔
3. Business Rules	✔	✔	na	✔	✔	✔					✔	✔	✔	✔
4. Database Domains				na					✔					
5. Database Objects	✔			✔	na				✔	✔				✔
6. External Data Interface Requirements	✔	✔	✔	✔	✔	na	✔	✔			✔	✔	✔	✔
7. External Quality Standards	✔	✔	✔	✔	✔	✔	na	✔	✔	✔	✔	✔	✔	✔
8. Information Needs	✔							na	✔	✔				
9. Mission, Organization Functions	✔	✔		✔	✔			✔	na		✔	✔		✔
10 Resource Life Cycles	✔	✔			✔			✔		na				✔
11. Use Cases	✔	✔	✔			✔	✔	✔	✔	✔	na	✔		✔
12. User Acceptance Tests	✔	✔	✔	✔	✔		✔					na		✔
13. Value Domain Management	✔	✔	✔				✔				✔	✔	na	✔
14. Work Breakdown Structure (WBS)	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	na

Table 9. Cross reference among work products.

9.0 Our Challenge

The final section of this paper contains the challenge that we must rise to if we intend to dramatically shift the challenged and failure statistics to the success column, and also short descriptions of a collection of metabase system centric projects that will go a long way to accomplishing the challenge's objective.

9.1 The Challenge

This paper concludes with our challenge: ***We, as IT professionals must proactively work to eliminate the high percents of IT system failures and challenges***. The key achievements of this challenge need to include:

- Adoption a framework such as the Knowledge Worker Framework that clearly identifies and defines each work product and precisely sets out their interrelationships.
- Recognition and proactively address the 90% or more of IT failure reasons that occur outside the formal domain of IT by ensuring that all IT projects holistically treat the entire spectrum of Knowledge Worker Framework work products appropriately within every project.
- Acquisition or development of a metadata management systems such as the Metabase System that can capture, store, report, and evolve all the IT related work products such that they are visible and are able to be used and re-used at a minimum across projects, and better across business functions, and ultimately across the enterprise.
- Engineering of work teams that are heavily populated with functional users while ensuring that all generated work products are non-redundant, integrated, and interoperable, and that they are stored and made visible through an enterprise metadata management system like the Metabase System.
- Ensuring that work accomplishment metrics are continuously captured so that real work plans and effort estimates can be quickly produced.
- Acknowledgment that enterprise data architecture models are the center of the IT universe and ensure that all IT projects ensure data consistency and interoperability.
- Ensuring that "local definitions" exist for all the business fact work products that exist for meta category value taxonomies, data elements, attributes, columns and DBMS columns so that naming and definitions can be automatic.
- Ensuring that all business information system analysis activities that detail use cases from business functions are cross-referenced to data models.
- Ensuring that all user acceptance tests are completely cross referenced to for example, use cases, requirements, mission-organization-functions, and database models.

If these accomplishments are achieved the following will almost certainly occur:

- Business Information System projects will be more accurate and features will be better known because the iterative requirements strategy will be eliminate requirements analysis and design errors that occur because the solution domain is not fully known prior to a project's plan, time-line, and resource estimate.

- Business Information System projects will have to be implemented in production systems only once because the vast majority of uncertainty will be eliminated.
- Business Information System projects will accelerate accomplishment because new ones will be able to be built upon pre-existing, well defined data and business information system work products that are enterprise mission-based rather than ever changing business organization and function based.

All in all then, productivity and quality will increase, and costs and risk will decrease to such levels that not only will the Chaos statistics of success grow towards 100%, but the fiscal benefits from doing this approach will exceed its costs.

9.2 Metabase System Projects

The following are Metabase System projects that will contribute the achievement of the challenge:

<table>
<tr><th>Metabase Project</th><th>Knowledge Worker Framework Columns</th><th>Metabase System Modules</th></tr>
<tr><td>1.0 Enterprise Mission Architecture</td><td>Mission</td><td>Mission Organization Function Position Assignment</td></tr>
<tr><td colspan="3">An Enterprise Mission Architecture project consists of creating hierarchically organizations of paragraphs that set out the idealized outcomes of the enterprise. These descriptions are idealized in that they are free from any identification of who and specification of how.
Properly done, mission descriptions are long lasting and change only when there is a significant change in the scope or purpose of the enterprise.

Once completed, mission models are detailed in terms of database domains, the organizations that accomplish missions, and high-level functional that represent illustrative processes that accomplish missions.
Database domain are noun-rich multi-level descriptions that illustrate the data required to represent the accomplishment of missions.
Missions are related in a many-to-many fashion with organizations. Mission-Organizations are related to high-level functions in a many-to-many fashion.
Missions models are considered complete when reviewers can “see” the entire enterprise in terms of its essential idealized accomplishments. Mission models are incomplete if any enterprise idealized accomplishment is missing.
Mission models are the cap-stone of many other models such as enterprise data architecture, database objects, resource life cycle analysis, and the like.</td></tr>
</table>

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
2.0 Database Objects (Conceptual)	Database Objects	Mission-Organization- et al module Database Objects Data Elements Implemented Data Model

The objective of a database object project is to identify high-level collections across a project, business function, or enterprise.

Database object classes consist of: 1) data structures of multiple collections of business facts , 2) the states that a database object proceeds through from initialization through termination, 3) the atomic processes that accomplish database table row insert, modify, or delete, and 4) the processes that transform a database object from one state to the other.

Database object projects are of two types: conceptual and detailed.

A conceptual database objects project begins with database domains that are set within the context of missions. Database domain are noun-rich multi-level descriptions that define the data required to represent the accomplishment of missions.

Each database domain’s set of nouns is examined to determine which nouns are 1) single value facts, 2) a collection of single value facts, or 3) multiple collections of facts.

A data element fact might be Salary or Zip code. These are stored in the data element metabase module.

A collection of facts might be Address, Evaluation, or Biographic Information. These are stored as candidate tables in a generic implemented database of the implemented database metabase module.

A multiple collection of facts might be Corporation, Purchase, Order, or Employee. These are stored as database object classes in the database object metabase module.

A database object classes are named multiple collection of facts such as Corporation, Purchase, Order, or Employee.

Database objects are often related to multiple mission-based database domains. Database object class data structures are detailed through the generic implemented data model tables. The columns of these tables are mapped to data elements. If implemented data models already exist, whole collections of tables can be mapped to a database object class. Once created the newly created database object classes can have its atomic processes, states, and database object transformations defined.

Database Object Classes can be related to each other through one to many relationships. From these relationships, the database object class is created. The value of the database object class diagram is to illustrate to management the overall architecture of data across a project, function, or the enterprise without being bogged down by the excruciating details that can exist in fully defined entity-relationship diagrams.

Once the conceptual database object project is completed, the detailed database object process is accomplished. That is, their tables, atomic processes, states, and database object transformations defined. It is often the case that whole business information systems focus exclusively on one or a few database object classes.

<table>
<tr><th>Metabase Project</th><th>Knowledge Worker Framework Columns</th><th>Metabase System Modules</th></tr>
<tr><td>3.0 Enterprise Data Architecture</td><td>Database Objects</td><td>Mission, Organization, et al
Data Elements
Specified Data Models
Implemented Data Models
Operational Data Models</td></tr>
<tr><td colspan="3">An Enterprise Data Architecture project consists of the complete engineering of the following data models: database objects, data element, specified, implemented, operational, and view. This project consists of subordinate projects which are: Database Objects, Enterprise Data Elements, Concepts Data Models, and Database Models.
Each of these subordinate projects can be done independently, or in conjunction one with another. Four examples serve to illustrate this top-down, bottom-up, and independent approach.
Operational data models can be built as the first step of reverse engineering across a collection of functionally related data models that are created through SQL DDL imports.
Implemented data models can be created by promoting just one operational data model and then promoting individual DBMS tables from other operational data models or even individual DBMS Columns.
Specified data models can be inductively created during an examination of a collection of Implemented Data Models.
Data element models can be inductively created through business fact promotions from a Specified Data Model project or an Implemented Data Model project.
Once these four data model layers are built, an implemented model can be quickly manufactured by first creating its schema, then importing collections of specified data model entities that become whole or partial tables, and finally creating relationships across these newly created tables.</td></tr>
</table>

<table>
<tr><th>Metabase Project</th><th>Knowledge Worker Framework Columns</th><th>Metabase System Modules</th></tr>
<tr><td>4.0 Enterprise Data Elements</td><td>Database Objects</td><td>Data Elements
Specified Data Model
Implemented Data Model</td></tr>
<tr><td colspan="3">An Enterprise Data Elements project consists of creating context-independent business fact semantic templates. Data elements are thereafter employed as semantic “rubber stamps” that can infuse business fact semantics into one or more attributes of entities within one or more specified data models, or one or more columns of tables within one or more implemented data models.
Enterprise data elements consist not only the data element specification itself, but also the contextual superstructure that enable viewing collections of data elements within overall concepts, conceptual value-domains, data element concepts, value domains, the interrelationship of value domain values, and the allocation of data elements to one or more classifications.
Data elements can be precisely allocated semantic and data use modifiers, that, in turn, enable automatic naming and automatic definitions. The semantic and data use modifier assignments enable the quick identification and subsetting of collections of data elements and their down-stream deployments within attributes, columns, and DBMS columns by selecting any of the semantic and data use modfiers.
Data elements can also be created through the business fact promotions from either specified data model attributess or implemented data model columns.</td></tr>
</table>

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
5.0 Resource Life Cycle Analysis	Database Objects Business Information Systems Business Events	Resource Life Cycle Analysis Business Information Systems Database Objects

A Resource Life Cycle Analysis project sets out the essential resources of the enterprise and the life cycle nodes of these resources. A resource might be Employee, and the resource life cycle nodes might be employee requirement, employee candidate, employee assignment, employee status change, and employee separation. Other resources include for example, facilities, finance, reputation, and products.

Each node represents a business-recognized state in the life cycle of the resource. The state is achieved through resource life cycle node assigned database objects and business information systems. Creation of the database objects and business information systems are accomplished in their respective metabase system modules.

Resource life cycles can be intersected by means of enablement vectors that signify that the accomplishment of one resource life cycle node enables the achievement of a different node on a different resource life cycle. The resultant diagram looks like a PERT chart. Assigned to the nodes are the relevant database objects and business information systems.

A key value of resource life cycle analysis is that entire collections of database objects and business information systems can be identified according to enterprise resources independent of business functions and business organizations. Additionally, this also enables the cross reference of business functions and organizations through their commonly employed resource life cycles.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
6.0 Concept Data Models	Database Objects	Data Elements Specified Data Models Implemented Data Models

A Concept Data Model Engineering project is the specification of collections of entities, attributes, and relationships contained within a specific business subject such as human resources, person, finance, logistics, manufacturing, or marketing.

Concept data models within the metabase system are called specified data models. That is simply because they are data models that are specified rather than either implemented as a database or operating via a DBMS. Concept data models are standardized data structure templates that can be used in an implemented database model manufacturing process.

Subjects can be hierarchically organized. Entities can also be hierarchically organized into subtypes. Attributes of entities are each related to a single data element. Relationships are expressed across entities through traditional primary and foreign keys.

Attributes can be assigned semantic and data use modifiers, and after their allocation, column names and definitions can be automatic. If an attribute’s ancestor data element already has assigned semantic and data use modifiers, the only ones able to be assigned are those that are specific modifier subsets. The same applies to assigned value domains.

Concept data models can be created inductively through promotion of one or more database tables from an implemented data model, can be created top-down by the identification of collections of subjects and the entities associated with those subjects, or be created as a single subject collection of entities via SQL DDL.

The value of specified data models is that they enable enterprise-wide view of all data across databases, business functions, or business organizations.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
7.0 Database Models	Database Objects	Data Element Model Specified Data Model Implemented Data Model Operational Data Model

Database Models represent are the models of databases in two different forms. DBMS independent and DBMS dependent. The first type is called Implemented (also called logical) and its triple is Schema, Table, and Column. The second is called Operational (also called physical) and its triple is DBMS Schema, DBMS Table, and DBMS Column. In both, relationships are expressed through primary and foreign keys.

Implemented data models and Operational data models are independent one from the other. They are not a logical form of a data model and then a physical form of the same data model. They can be related to each other in a many-to-many relationships. There can be DBMS tables that contain DBMS columns with ancestor columns from different implemented data models. That of course is typical for data warehouse data models.

Implemented data models can be created through the creation of a schema and the importation of entities and attributes from specified data models. Implemented data models can also be created through the promotion of an entire operational data model or just a subset of DBMS tables. Finally, implemented data models can be created by SQL DDL imports.

Implemented data model columns can be assigned semantic and data use modifiers, which, in turn enable automatic column name construction and definition generation. If a column's ancestor attribute and/or data element already has assigned semantic and data use modifiers, the only ones able to be assigned are those that are specific modifier subsets. The same applies to assigned value domains.

Operational data models are models of databases that are bound to a specific DBMS. These database models can be created through imports of SQL DDL. Initially upon import, these database models are automatically mapped to "unknown" ancestors. If these models are to be related to one or more existing implemented data models, a specific DBMS column is selected and its ancestor column from an implemented database model is selected and the relationship is created.

Operational database models can also be created by importing an entire schema and/or one or more tables from an implemented data. Relationship maps between the two levels of database models is automatically maintained. DBMS columns can have their value domains specifically identified, and if an ancestor column's already has an assigned value domain, the only ones able to be assigned are those that are specific subsets.

All the created database models can be exported into SQL DDL.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
8.0 Fact Name and Definition Management	Database Object	Administration Data Elements Specified Data Models Implemented Data Models Operational Data Models

A Fact Name and Definition Management project is an activity to regularize the names and semantics that are assigned to data elements, attributes, columns and DBMS columns. Data elements, attributes and columns can be assigned semantic and data use modifiers, and after their allocation, column names and definitions can be automatic.

When DBMS columns are created, their names can be automatically abbreviated. Each collection of abbreviations can be created within the Admin module. Each collection can be segregated by the business domain associated with the abbreviation collection. The process of abbreviation can either be arbitrary set or computer generated. In the case of computer generation, the abbreviation process consists of removing vowels and redundant consonants beyond the first letter of the word that is being abbreviated.

It is critical to understand that DBMS column names do not have to be changed from what is already deployed in legacy databases. That's because the true nature of a DBMS column is more important than its localized name.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
9.0 Fact Semantics Management	Data Elements	Data Elements

A Fact Semantics Management project focuses on the creation of semantic and data use modifiers that are assigned to data elements, attributes, columns. Prefix modifiers are semantic modifiers and modify that actual meaning of the fact's common business name. Suffix modifiers are data use modifiers and indicate the actual data type of the fact and the types of use within which the fact will be employed.

The sequence of applying the semantic and data use modifiers is established in the data elements module so that the manufacturing of a name is reliable, repeatable, and discriminating. No two facts can have the same name.

Each modifier has a localized definition that is employed during the process of creating an automatic definition.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
10.0 Use Cases	Business Function	Use Cases

A Use Cases project represents a detailing of a business function into process specifics and sequences. Use cases can contain pre-conditions, post conditions, special conditions, use case hierarchies, actors, and the like. Use cases can also be configured into networks.

There can be multiple use cases for a given function that are generally equivalent. Some use cases might be on-line presentation layer cases and other use cases can be for batch processing scenarios. Regardless, use cases are mapped to three different components that is, mission-organization-functions, columns of database tables, and business information systems that ultimately become the implementing mechanism for the use cases.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
11.0 Business Information Systems	Business Information System	Business Information System Resource Life Cycle Analysis View Database Object Use Cases Requirements

A Business Information Systems project consists of specifying the business information systems that need to be implemented. Business Information Systems can be hierarchically organized to as many levels as appropriate. They can be set against business event cycle and/or calendar cycle networks. Each business information system's characteristics is described.

Through other metabase projects, the complete business information system specification includes views through which the business information system operational database data is defined, resource life cycles that represent the roles business information systems play in the enterprise, database object information systems that are the instigating agents for transforming database objects, use cases that detail the processes and sequences that are to be accomplished within and across business information systems. All these specifications become the business information system's requirements.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
12.0 Information Systems Plans	Database Objects Business Information Systems	Database Objects Business Information Systems Resource Life Cycle Analysis

An Information Systems Plans project consist of the confluence of a number of components. Key among these are the resource life cycle node networks. To each resource life cycle node the appropriate set of database objects and the business information systems are mapped and sequenced.

The time-sequenced resource life cycle node network represents the sequence across the domain of database objects and business information systems of what has to be implemented and in what sequence.

The resource estimates assigned to the accomplishment of the resource life cycle node implied information system project is accomplished by applying standardized work breakdown structures, metrics that contain unit effort staff hour amounts, and the quantity of units that have to be constructed. These are collected and fed into the project management system that produces detailed work plans, PERT, Gantt, and Critical Path charts.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
13.0 Reverse Data Engineering	Database Objects	Data Elements Specified Data Models Implemented Data Models Operational Data Models

A Reverse Data Engineering project accomplishes the creation or updating of implemented data models, specified data models and data element models. The process starts with the importing of a number of operational data model SQL DDL scripts from one or more functional areas.

Once the collection is imported into the operational data model, these DBMS schemas are examined to determine the one that is the “least worst.” Assuming that a relevant implemented data model does not exist, it is promoted to be an implemented data model.

Thereafter, each remaining operational data model DBMS schema is examined to determine how it is reflected in the just created implemented data model. If a table is missing, it is promoted from the operational data model into the implemented data model. If only a column is missing, it too can be promoted from the relevant operational data model table.

Once the implemented data model is completed, the tables are reviewed to see if any collection should be elevated to be a specified data model. If so, the specified data model is created. It is very likely that the names of specified data models subjects and entities are more generalized than implemented data models.

At the conclusion of such implemented-to-specified mappings there will be a collection of specified data model templates that will enable a quick identification of how and where data structures are implemented across the enterprise’s missions, organizations, functions, business information systems, database objects, database models and resource life cycles.

During the effort of creating the implemented and specified data models, columns and attributes that are commonly used should be promoted to be data elements. Once a collection of data elements is created they can be reviewed to then create their data element superstructures.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
14.0 Forward Engineering	Database Objects	Data Elements Specified Data Models Implemented Data Models Operational Data Models

The Forward Engineering project accomplishes the creation of a database model. An implemented database model is created by first creating a schema. Entities from one or more subjects from the specified data model are imported into the implemented data model. The newly created column names can be revised to be more relevant to the implemented or operational database models.

Additional semantic and data use modifiers can be assigned to the columns so long as they represent subsets of previously assigned modifiers. The metabase system checks that automatically and assignment is not allowed to proceed unless the assignment is of a proper subset.

Additional value domains can be assigned to the columns so long as they represent subsets of previously assigned value domains.

This same process is followed in the creation of an operational data model from an implemented data model. In this case, tables from one or more implemented data model schema can be imported to become part of a single operational data model.

Additional value domains can be assigned to the DBMS columns so long as they represent subsets of previously assigned value domains.

Metabase Project	Knowledge Worker Framework Columns	Metabase System Modules
15.0 Requirements	Mission Organization Function	Requirements Mission-Organization-Function Business Information Systems Resource Life Cycle Analysis Use Case Operational Data Model Database Objects

A Requirements project consists of collecting and categorizing requirements that need to be accomplished in the completed database and/or business information system effort. Requirement categories can be hierarchically organized. The requirements themselves can be organized into a network. Once the requirements are collected they can be mapped to mission-organization-functions, business information systems, resource life cycle nodes, database objects, business events, and DBMS columns.

Requirements can be created top-down or bottom up. When a business information system project is started, requirements are often a very first task. These requirements are typically created without much regard to the ultimately existing database models, needed business information systems, or the business organizations and functions that must either exist or be modified. Commonly therefore, requirements exist as a single document.

During the accomplishment of a business information system project, requirements can be mapped to various work products that are created such as mission-organization-functions, business information systems, resource life cycle nodes, database objects, business events, and DBMS columns. This enables the substantiation of the need for specific requirements and enables requirements-based cross references.

This process of requirements management enables the top-down identification of requirements to be validated, and over time changed, and also the ability to discover and properly create new requirements in a bottom up fashion as the various work products are created.

10. Conclusions

The practical application of the points made in this paper include:

- The Success, Challenge, and Failure statistics are grossly unacceptable and need to be addressed.

- By the time the true cost and schedule is known, way too much time and money has been spent for consideration of any practical and cost effective correction.

- The Knowledge Worker Framework is expertly engineered to address the complete environment of business information development. The major work product categories (columns) from the Knowledge Worker Framework are interconnected with many-to-many relationships and result in non-redundant, integrated, and interoperable work products.

- Analyses of GAO reports on 13 $100+ Million IT systems showed that more than 40% of all failures start with insufficient requirements.

- Analyses of GAO reports on 13 $100+ Million IT systems showed that close to 50% of all failures occur after the IT system is deployed because of non optimized business organizations and functions.

- Analyses of GAO reports on 13 $100+ Million IT systems showed that only about 5% of all failures occur during the actual development of the IT system.

- To have IT system success, we must focus intensively on the more than 40% and the 50% of errors that arguably are outside of IT proper.

- The cost to fix IT errors dramatically increases as an IT development efforts progresses from requirements to design to implementation to post-deployment. Hence the most cost effective time to identify and correct IT system errors is during the requirements and design stages.

- Prototyping through business information system generators such as Clarion (www.softvelocity.com) is not only fast, it is also profoundly effective in validating and evolving requirements and design work products, thus eliminating a significant majority of challenge and failure errors. These prototypes will provide advanced insight into the required business organization and business function optimizations.

- Data models from data elements through concepts, logical, physical, view and XML are a significant common intersecting mechanism for a large variety of work products across the entire business information system development effort and across business information systems within the enterprise.

- The Metabase System is a comprehensive system for the capture, storage, reporting, and evolution of virtually all the key work products of a given business information system development effort. The Metabase System also enables work product cross referencing, auditability and traceability across the enterprise.

- If careful attention is not paid to these concerns, be light at the end of the tunnel will almost certainly be a train.

11. References

The following references to Whitemarsh materials provide a more detailed exposition practical application of the significant content of this paper.

The following documents are available free from the Whitemarsh website:

Free Whitemarsh Materials	
Paper	**URL**
Managing Data Models	http://wiscorp.com/sp/sp04.pdf
Enterprise Architectures	http://wiscorp.com/sp/sp08.pdf
Engineering and Managing Information Systems Plans	http://wiscorp.com/sp/sp11.pdf
Manufacturing Project Plans	http://wiscorp.com/sp/sp12.pdf
Earned Value Management	http://wiscorp.com/sp/sp15.pdf
Business Event Management	http://www.wiscorp.com/sp/sp16.pdf
RFP Development	http://www.wiscorp.com/sp/sp18.pdf
Data Modeler Architecture and Concept Of Operations Reverse and Forward Engineering Guide Metabase Module User Guides	http://www.wiscorp.com/metabase_demo.html
Comprehensive Metadata Management (short paper).	http://www.wiscorp.com/featured_papers.html
DAMA 2002 - Metadata Architecture for Enterprise Wide Data Sharing - Problem Specification DAMA 2003 - Metadata Architecture for Enterprise Wide Data Sharing - Problem Solution	http://www.wiscorp.com/DatabaseDesignInformation.html
Comprehensive Metadata Management	http://www.wiscorp.com/ComprehensiveMetadataManagement.pdf
Metabase Overview	http://www.wiscorp.com/Metabase.zip

Free Whitemarsh Materials	
Paper	**URL**
Metabase User Guides	http://www.wiscorp.com/MetabaseUserGuides.zip
Data Management Conferences	http://www.wiscorp.com/dama2002.zip http://www.wiscorp.com/dama2003.zip http://www.wiscorp.com/wrad2000.zip

The following documents are available for Whitemarsh Website Members. The URLs that follow provide descriptions of the pages. Members should log in and proceed to the appropriate page, e.g., Enterprise Database, find the book, paper, or course and perform the download.

Members Only Whitemarsh Materials	
Paper	**URL**
Data Management Program - Data Standards Architectures And Implementation Data Management Program - Engineering Data Management Program - Metadata Architecture For Data Sharing Data Management Program - Tag And Post Vs Data Standardization Data Management Program - Metadata Architecture For Data Sharing Data Management Program - Database Interface Architectures Data Management Program - Projects And Data-Asset Product Specifications Data Management Program - Work Breakdown Structures	http://www.wiscorp.com/EnterpriseDatabase.htm
Iterations of Database Design	http://www.wiscorp.com/DatabaseDesign.htm
Data Is Executed Policy	http://www.wiscorp.com/DatabaseProjects.htm

Members Only Whitemarsh Materials	
Paper	**URL**
Data Management Program - Work Breakdown Structures Knowledge Worker Framework Database Objects Managing Database - Four Critical Factors	http://www.wiscorp.com/wwmembr/mbr_products_edb.html
Work Breakdown Structures	http://www.wiscorp.com/wwmembr/mbr_products_dp.html
Data Architecture Classes Guidelines for Data Architecture Class - Data Warehouse	http://www.wiscorp.com/wwmembr/mbr_products_dd.html
Work Breakdown Structures Database Project Work plan Templates Information Systems Development Methodology Phases 1 and 2 Whitemarsh Project Estimating Work plan Development	http://www.wiscorp.com/wwmembr/mbr_products_dp.html