



**Whitemarsh**  
Information Systems Corporation

## *Manufacturing Project Plans*

Whitemarsh Information Systems Corporation  
2008 Althea Lane  
Bowie, Maryland 20716  
Tele: 301-249-1142  
Email: [Whitemarsh@wiscorp.com](mailto:Whitemarsh@wiscorp.com)  
Web: [www.wiscorp.com](http://www.wiscorp.com)

## **Table of Contents**

1.0	Objective .....	1
2.0	Topics .....	1
3.0	Understanding Work Breakdown Structures .....	2
3.1	Understanding WBS .....	4
3.2	Homogeneous Wn&vBS. ....	6
3.3	Heterogeneous Wn&vBS .....	9
3.4	Work Plan Summary .....	11
4.0	Highly Engineered Project Management .....	13
5.0	Standardized Project and Deliverables Templates .....	15
6.0	Assessing and Assigning Staff .....	18
7.0	Project Plan Constriction and Estimation .....	20
8.0	Recording Work and Generating Earned Value Reporting .....	26
9.0	Conclusions .....	27
10.0	References .....	28



## **1.0 Objective**

The objective of this paper is to present the Whitemarsh approach to engineering project plans through the use of collections of standardized work breakdown structures, earned-value proven metrics, and the management of these plans within a comprehensive metadata management system, Metabase, that additionally supports earned value based reporting.

This paper asserts that there is no real difference between real-product metadata and project-management data. Project management data is metadata because it represents an abstraction of real-projects. Both forms of these metadata overlap in content and purpose. Each however has a different use-perspective. Real-product metadata serves to identify, catalog, and manage metadata-based representations of deliverables (i.e., data and process models). In contrast, project-management data identifies, catalogues, and manages the projects that ultimately result in the creation of real-product metadata.

Consequently, this paper asserts that because of the overlap of purpose and content, both real-product metadata and project-management data belong in the same database and should be managed by a common metadata management system. For Whitemarsh, that means the Metabase System. As of the writing of this Short Paper, both the Metabase System and the Whitemarsh project management system exist but have not yet been integrated. The points of integration are obvious and were designed to be so. Additionally, the development environment employed by Whitemarsh enables the integration of these data and systems in just a few staff months.

## **2.0 Topics**

The topics in this paper include:

- Understanding Work Breakdown Structures
- Highly Engineered Project Management
- Standardized Project and Deliverables Templates
- Assessing and Assigning Staff
- Project Plan Construction and Estimation
- Recording Work and Generating Earned Value Reporting

More software projects have gone awry for lack of calendar time than for all other causes combined. Why is this cause of disaster so common? Five reasons come to mind.

- First, our techniques of estimating are poorly developed. More seriously, they reflect an unvoiced assumption which is quite untrue, i.e., that all will go well.
- Second, our estimating techniques fallaciously confuse hours expended with deliverables accomplished, hiding the assumption that men and months are interchangeable.



- Third, because we are uncertain of our estimates, software managers often lack the courteous stubbornness of Antoine's chef. <sup>1</sup>
- Fourth, schedule progress is poorly monitored. Techniques, which have been proven and routine in other engineering disciplines are considered radical innovations in Information Technology (IT).
- Fifth, when schedule slippage is recognized, the natural (and traditional) response is to add manpower. Like dousing a fire with gasoline, this just makes matters worse, much worse. <sup>2</sup>

### 3.0 Understanding Work Breakdown Structures

A work breakdown structure (WBS) is the breakdown of work. That's easy, but what's the definition of "work?" First of all, is it a noun or a verb? If it is a noun, a WBS is a breakdown of a product. If the work is a database schema, it's the schema, the tables, the columns within the tables, the data types of the columns, and the like. But, if work is a verb, then a WBS is a breakdown of processes such as Perform Assessment, Design Database, Engineer Hardware, Accomplish Testing, Commence Production, and Perform Maintenance.

In all actuality, the answer to the question above is, Yes. Work is both a noun and a verb. Consequently, both breakdown structures exist. Not only that, but more breakdown structures exist and they all have to be considered and employed appropriately to have successful projects. The set of breakdown structures include:

- Product WBS
- Organization WBS
- Finance WBS
- Methodology WBS
- Requirements WBS
- Computing Infrastructure WBS

---

<sup>1</sup> Written on the menu at Antoine's in New Orleans: "Good cooking takes time. If you are made to wait, it is to serve you better, and to please you." From Frederick P. Brooks, Jr., *The Mythical Man-Month*. Reading, MA: Addison-Wesley, 1975, page 13.

<sup>2</sup> Ibid, page 14.

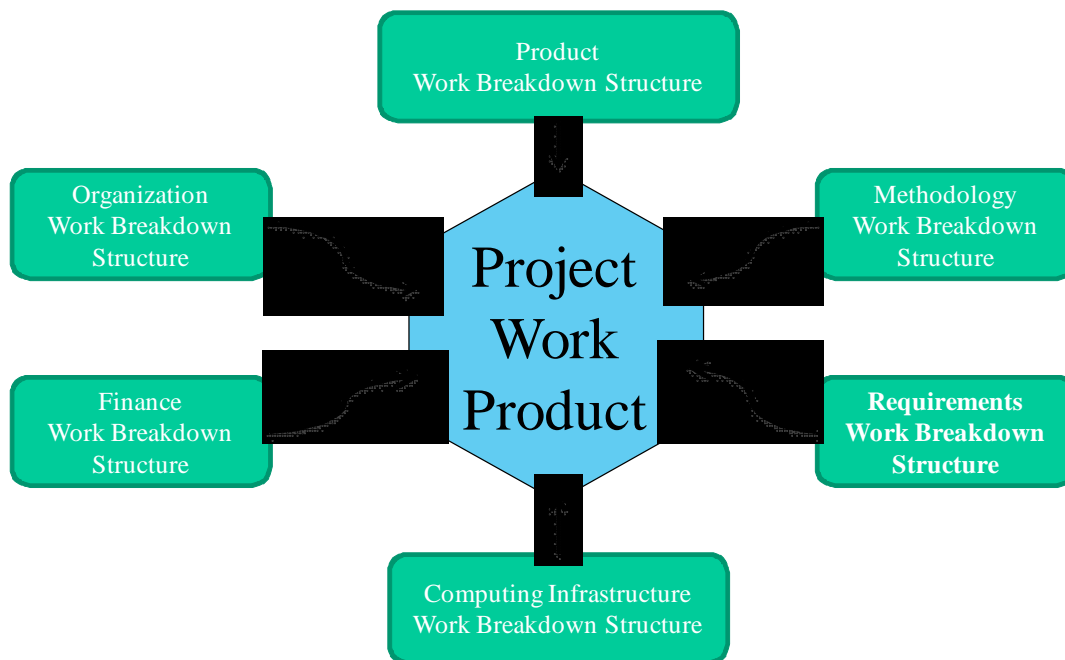


These six breakdown structures, all generally referred to as WBSs are involved in the creation of project work products. Figure 1 shows a general relationship. This figure illustrates that a given project work product fits within the six breakdown structures provided in Table 1.

All IT projects are developed through project plans that commonly employ work breakdown structures (WBS) of the classes listed above. This short paper concentrates on just two of the WBSs, that is, Methodology, which is hereinafter referred to as WvBS and the Product, which is hereinafter referred to as WnBS. An overall project plan involves not only these two WBSs, but also the following additional four items:

- PERT chart, a network that shows task interrelationships
- Unit effort estimates
- Factors affecting unit effort estimates
- Recording of real-work accomplishment

These four items are addressed in later sections of this paper.



**Figure 1.** Project work product setting within critical breakdown structures.



Breakdown Structure Class	Example Context Description
Product WBS	A product from the Product WBS, such as Columns of a Table within a Schema.
Methodology WBS	Accomplished by Task 2.2.2, Develop Data Model of Develop Logical Database of Conceptual Specification Phase.
Requirements WBS	A component of the Data Model of the Data Architecture within the Target Architecture
Computing Infrastructure WBS	Accomplished by the Data Modeling tool, xyz on the workstations within the servers of the 123 network node.
Finance WBS	Paid for by the Enterprise Data Infrastructure Development Project.
Organization WBS	Accomplished by the Data Modeling staff of the Enterprise Data Group of Corporate Infrastructures.

**Table 1.** Work Breakdown Structure Classes and descriptions

### 3.1 Understanding WBS

The term *work breakdown structure* is commonly employed in projects. Its first word, *work*, if not understood by all, can lead to significant confusion. Within the U. S. Department of Defense community, the word *work* is often seen as a noun. Thus, the term work breakdown structure implies a hierarchical decomposition of the actual product being delivered.

In Information Technology (IT), the term work breakdown structure is also commonly employed in projects, but it is used as a verb! Thus, in IT, the term work breakdown structure implies a hierarchical breakdown of the activities.

The misunderstanding is not so bad, however, because in the Continuous Flow Model illustrated in Figure 2, both types (verbs and nouns) are employed. Through this Continuous-Flow process, several unique features are present:

- All four processes are concurrently executing.
- Changes to enterprise resources occur in unison, periodically, and in a very controlled manner.
- A metadata management system, Metabase, always contains all the enterprise resource specifications: current or planned. Simply put, if an enterprise resource semantic is not within the metabase, it is not enterprise policy.
- All changes are planned, scheduled, measured, and subject to auditing, accounting, and traceability.



- All documentation of all types is generated from the metabase.

The phrases from Figure 2 that comprise the process bubbles are verbs, the set of WvBS. In the Metabase System, the storage location of the products of an Management Information Systems project, the breakdown is really a WnBS. While the two are not the same, they are definitely related, for example, *Plan the Project* and *Project Plan*.

There is, therefore, a strong correlation between activity lists (WvBS) and product lists (WnBS). Not surprisingly, if one project's product list is different from another project's product list, the two activity lists are also likely to be different. This leads to the conclusion that since there are different and valid product lists (WnBS), there must also be different and valid methodologies (WvBS). The appropriate pairing of a WnBS with an appropriate WvBS is given this notation: Wv&nBS.

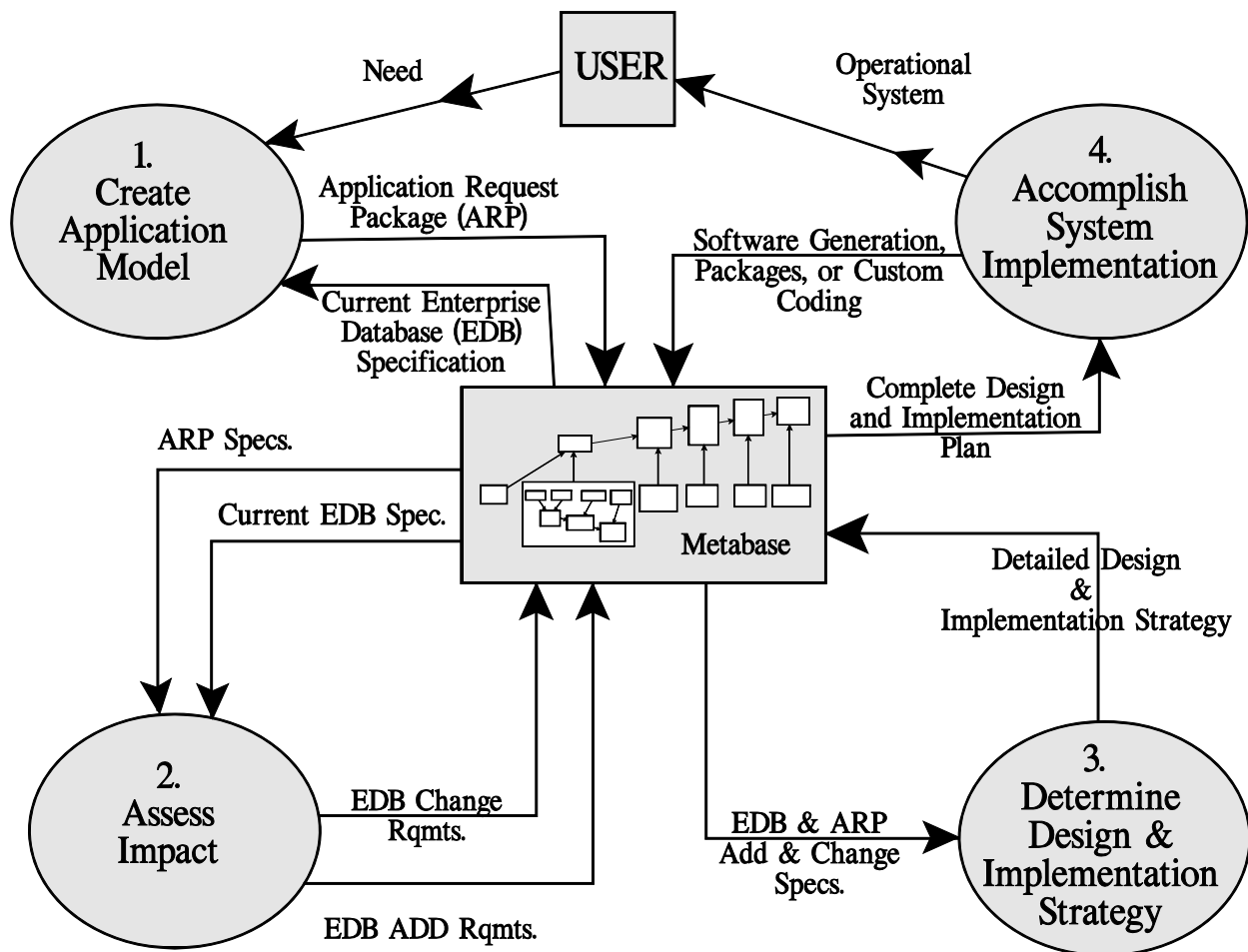


Figure 2. Continuous Flow Model.



*For any particular project, then, if its product list is fundamentally different from that of another project, the methodologies must also be different.*

In addition to pairing appropriate sets of Wn&vBS for specific projects, multiple customers may require, for example, MISs in different subject areas. For example, one client had an MIS project for Government Grant and Loan Accounting. Another client had an MIS project for Program, Planning, and Budgeting (PBS). While both projects are MISs, and both could have been developed through the same database project methodology, each client wanted to see *their own personalized* work plan. A Wn&vBS needed to be developed for each customer, that is, one for the Grants MIS and another for the PBS customer. While the customers may be satisfied, a contractor performing the work will appear to have undertaken very different projects when, in fact, the structure and format of the real work products and the methodology employed are essentially the same. Lost from the contractor will be any internal attempt to measure work accomplishment or performance across MIS projects, and any benefits from common training, use of a multiple-project repository, and the like.

These losses can be avoided if the contractor first has its own Wn&vBS for MISs. From this internal MIS Wn&vBS, the contractor can build relationships between its own Wn&vBS and that of the customer. Figure 3 presents a diagram that portrays such a set of relationships. Such an arrangement permits contractor methodologies to be evaluated, improved, and audited over many years and against many projects.

### 3.2 Homogeneous Wn&vBS.

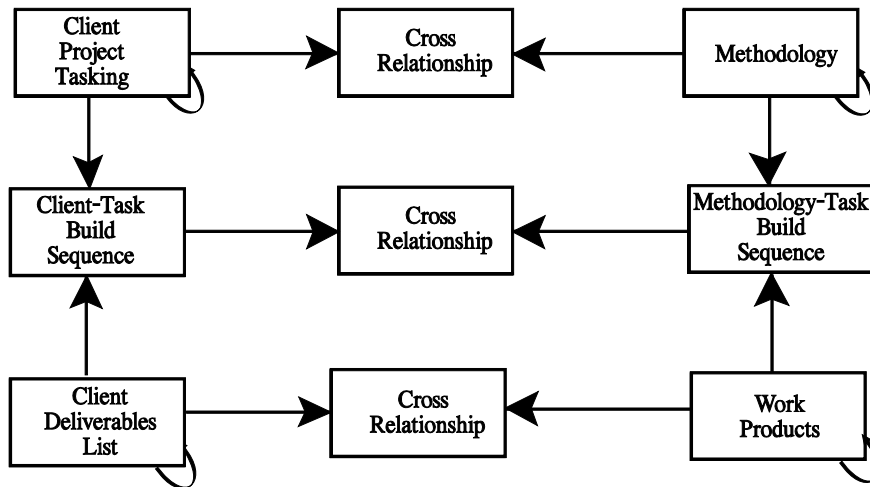
In Figure 3, a project's methodology is represented by the methodology meta-entity and recursive relationship on the top-right. The product list of the methodology is represented by the work-products schema meta-entity and recursive relationship on the bottom-right. As steps of the methodology produce product, intersection meta-entities (called the *Methodology-Task Build Sequence*) are created.

The *Methodology-Task Build Sequence* intersection table is more than just an intersection table because it must have a sequencing all its own so that the methodology and products can be accomplished in a specific project-logical order. Figure 4 illustrates a set of Wn&vBS. The WnBS is on the left side and shows the breakdown of the Conceptual Design. Conceptual design may be the name of the overall product of the conceptual design phase. In a previous phase, Preliminary Analysis, there are other products such as mission model, high-level E-R diagrams, evaluation criteria, and so forth. In this hierarchy, three of the products, that is, logical model, physical model, and interrogation, are shown. Contained within the logical model are the E-R diagram, objects, and data elements.

On the right side of the figure is a WvBS. It shows the verb sequence of *Build Conceptual Design* down through three of the steps within *Build Logical Database*. The intersection of the WnBS and the WvBS is shown at the bottom. Among the attributes contained







**Figure 3.** Interrelationships between standard and client work breakdown structures.

in the Build Sequence table would be sequence, start date and time, and expected stop date and time. Other tables that could be attached to the build sequence table could be various time cards from staff members performing work. And, related to the various rows from the WnBS table could be the actual work products themselves. For example, attached might be an entity-relationship diagram entity called Accounts Payable.

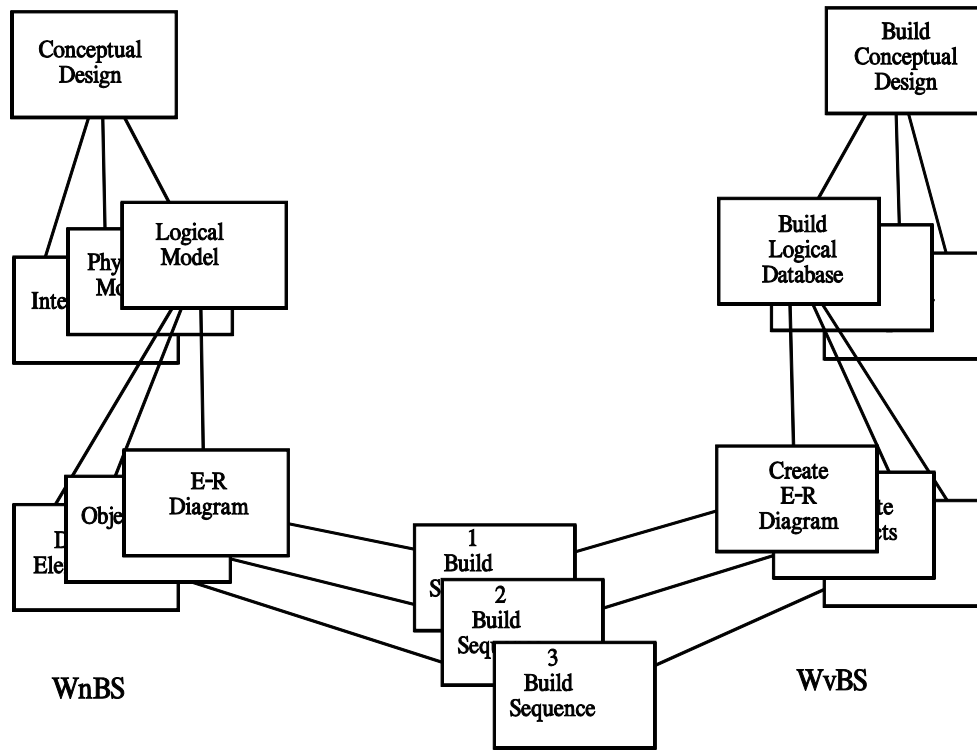
Analogous to the way the methodology and work products are organized and interrelated on the right side of Figure 3, the client's project tasking and deliverables list are represented on the left side of Figure 3. When requests for proposals (RFPs) are issued, the buyer often issues what is on the left. *If the contractor is a quality contractor, the right side also already exists.* How to intersect the two? The intersection records, all labeled cross-relationships, provide the explanation.

Once the two Wn&vBS are created, a sophisticated contractor can quickly tell whether the RFP is one of quality. Assuming that the Wn&vBS of the contractor is valid, there will be a good match between the buyer's Wn&vBS and the seller's. If the match is not exact, four possibilities exist:

- The buyer is attempting to buy less than what is needed.
- The buyer is attempting to buy more than what is needed.
- The seller would build less than what is needed.
- The seller would build more that what is needed.

Regardless of which alternative exists, the discrepancies have to be examined and resolved before a quality proposal is generated and before a quality contract is consummated.





**Figure 4.** Illustrated instance interrelationships between WnBS and WvBS.

*Contract performance disputes are almost always based on unknown mismatches between the client's and the contractor's Wn&vBS.*

Every project has a product list that is more or less unique. For projects with homogeneous product sets, for example, a human resources system and an operating system, the specific product list for the HR system is different from the product list for an operating system. Included in the differences would be the types of products, the types of tests, the interface products, and so forth. In this case, some of the products are likely to be the same in form but very different in content.

Additionally, there may be a difference in the significance of one product over another. For example, extremely detailed logic design is required before an operating system is coded, but only logic sketches are required for the HR system that is 4GL-based. Similarly, the database design section for the HR system requires significant effort, while the effort may be nonexistent for an operating system.

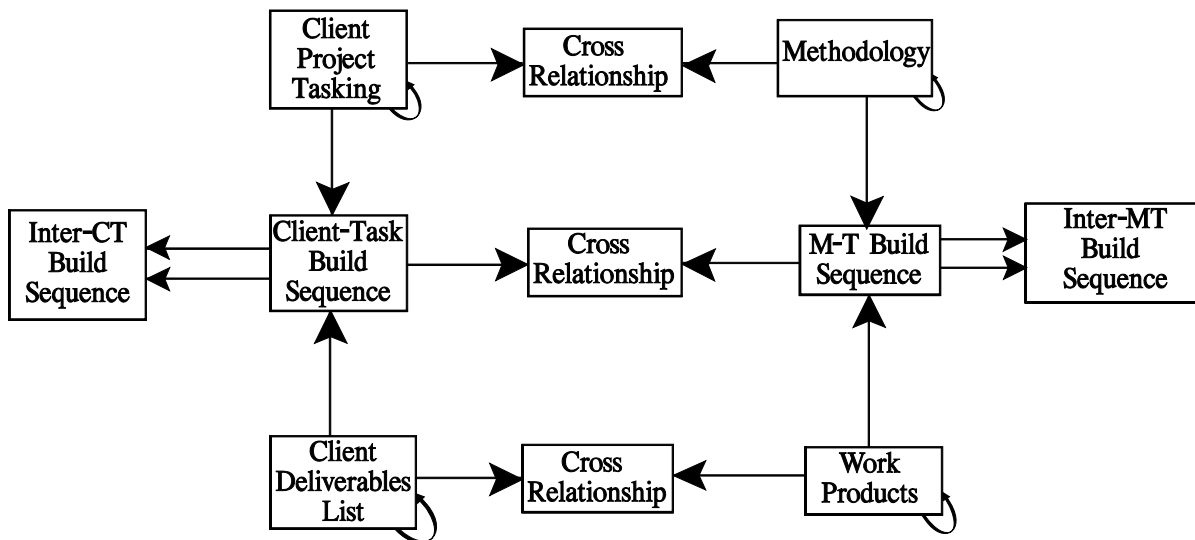


### 3.3 Heterogeneous Wn&vBS

For heterogeneous projects, for example, a command and control system for the military, the product list includes software, hardware, training, documentation, quality measures, standards adherence, and so forth. Each of these products has a unique product list, that is, a unique WnBS. Furthermore, a command and control system for the U. S. Air Force would be somewhat different than one for the Navy or the Army. A critical issue, then, is how to provide a unified Wn&vBS to the client for diverse product sets (WnBS) and, by implication, diverse process sets (WvBS).

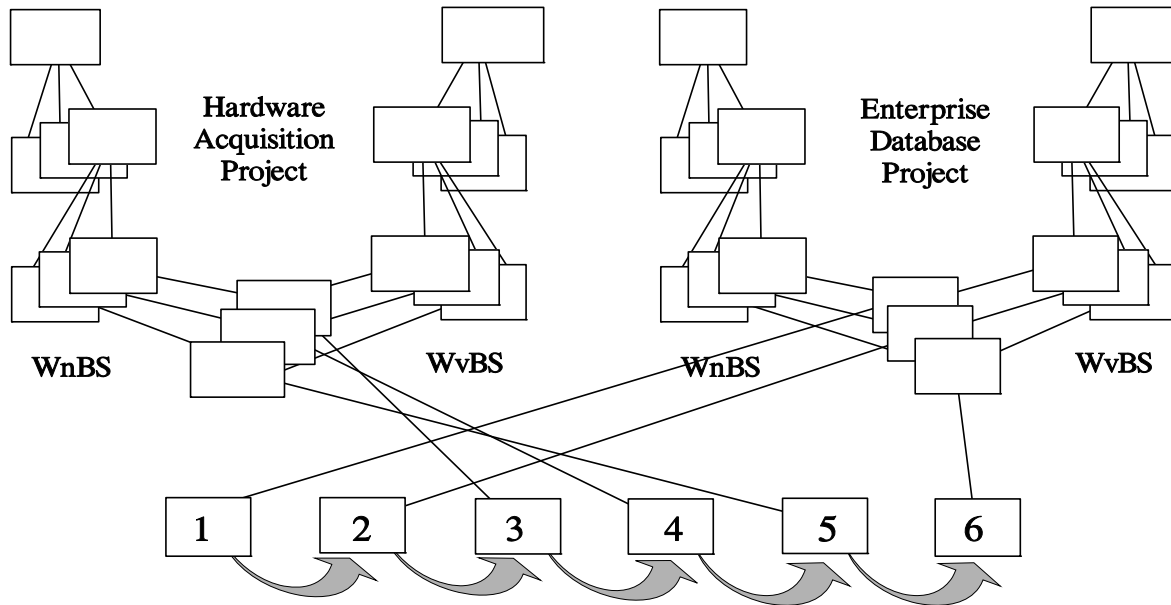
Figure 5 shows a data structure to handle heterogeneous Wn&vBSs. The center of the figure is the same except that there are multiple hierarchies, one for each methodology. Similarly, there would be multiple subsets of the repository schema for each different product type. The additional entities allow inter-WnBS and inter-WvBS relationships. Such relationships allow projects to identify, for example, the relationship between an HR data update subsystem, which is a product within an HR WnBS, and a particular computing hardware subsystem from an appropriate hardware WnBS. If the client's RFP is of quality, it too will have similar structures as depicted on the left side of Figure 5.

Figure 6 illustrates a heterogeneous set of Wn&vBS. The project on the right side of the figure is a particular enterprise database project. From Figure 3, there could be a multiple of such projects underway. There would then be multiple sets of the right side of the diagram. Several might be for the concurrent development of detailed data models for particular aspects of an initial high-level E-R diagram effort.



**Figure 5.** Interrelationships between multiple standard and client work breakdown structures.





**Figure 6.** Interrelationships between Heterogeneous WnBS and WvBS/

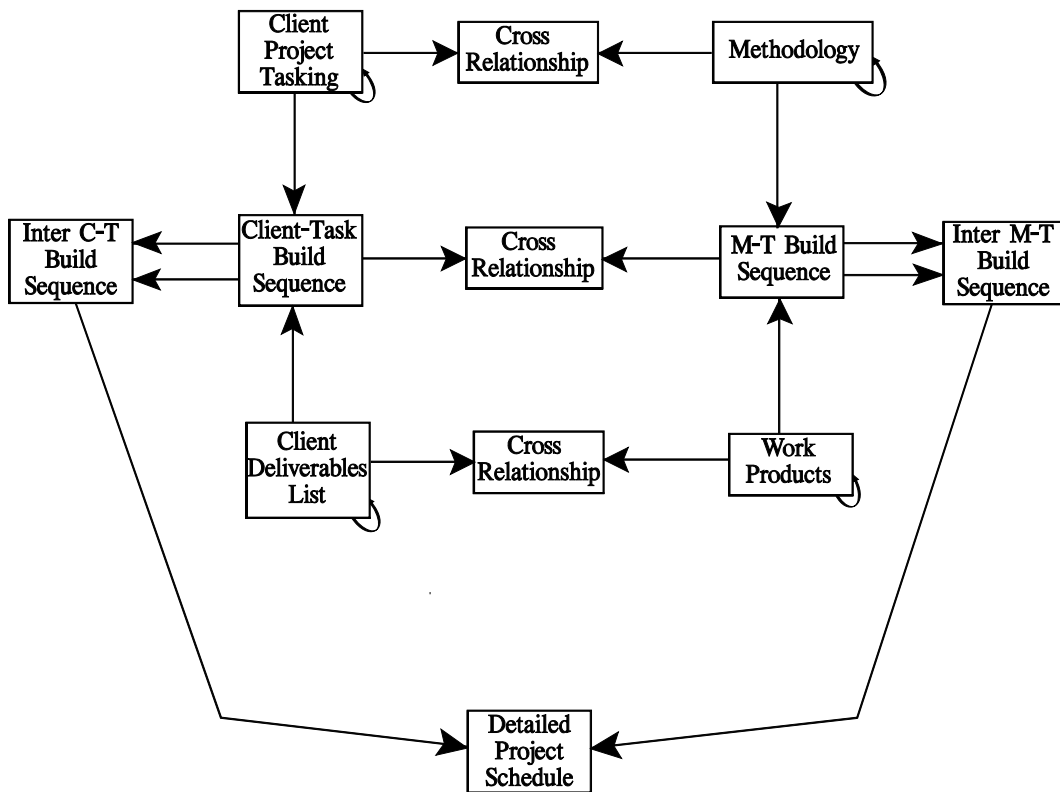
For example, there might have been the development of an overall E-R diagram that included the finance area, and then three to six smaller projects to develop detailed data models for accounts payable, procurement, receivables, general ledger, and cash flow management. Each of these projects would have their own WvBS and would commonly contribute to different and to shared instances of the same WnBS.

There might also be, as depicted on the left side of the figure, a project underway to configure the overall set of hardware to accommodate an entirely new finance system. This, of course, cannot be done until the data and processing requirements of the various enterprise database projects are completed.

Figure 6 thus shows how the Wn&vBS for each project type would be created, and then, as illustrated on the bottom of Figure 6, the timing sequence of when all these project types should be finished.

Once all the various Wn&vBSs are developed, an overall schedule, satisfactory to both the client and the contractor, must be created. Figure 7 shows the meeting of the minds between the two parties. The two middle meta-entities at the ends show the client schedule and the contractor schedule. And, the meta-entity at the bottom shows the matching between the two schedules, or the detailed project schedule.





**Figure 7.** Interrelationships between Wn&VBS, contractor, client schedules, and overall schedule.

### 3.4 Work Plan Summary

The approach to enterprise-wide database must be focused on only one objective: the successful completions of database projects within and across the missions, organizations and functions of the enterprise.

A quality database project life cycle provides a clear cut path, and a common-sense approach, to database projects. Using quality life cycles enables a project's costs and schedules to be accurately projected and managed. The controls imposed by a quality life cycle enables the knowledge worker to identify what products are due, when they are due, and under what quality measures the products are judged. Because each product is well defined, project members can focus their work efforts. In short, knowledge work is accomplished more quickly and more accurately.

A quality life cycle reduces project anxiety as each next step is already set down and is a logical consequence of the prior step. Again, sharply focused targets translate into quality work.



The methodology life cycle, along with all the estimates, knowledge worker product specifications, actual products, and worker time cards indicating progress, must be stored in the Metabase System database. This ties project planning to product deliveries—in short, real project management and control. Because the Metabase System is database-centric, cross-reference reports are easy to generate.

A quality database environment represents a complete solution to database needs. It represents the time-tested, rigorously defined products necessary for organizations to accomplish database projects successfully, the first time, and within budget.

The amount of work in creating a project plan and schedule, and controlling and tracking a project through its entire life cycle, should not be underestimated. Prudent project managers spend about 20-45 percent of the total project's cost on this task alone, depending on the overall size of the project. This means that on a ten staff-year project of real work, at least another two staff-years should be spent on its planning, scheduling, and ongoing control. Too little time causes:

- Bad initial estimates due to poor work assessments
- Inaccurate progress reporting during project execution
- Under delivery of product, given required quality
- Under delivery of quality, given required product

The next sections of this paper provide a highly engineered strategy that has been implemented to satisfy the following requirements:

1. Enterprise visibility of all projects.
2. Standardized project, task, and deliverables templates to ensure work product sharing and comparative analysis.
3. Standardized work environment factors so that true accomplishment can be measured and productivity improvements can be realistically accomplished.
4. Standardized staff assessments, assignments, productivity measures to ensure fair and accurate assessments of staff quality and progress.
5. Deliverables based work reporting so that earned-value assessments of progress can be made.



#### 4.0 Highly Engineered Project Management

Highly engineered project management requires a database-centric system to capture and manage the data critical to effective project management. Organizations taking an enterprise-approach must have a centralized database application for project management to avoid the proliferation of stove-pipe based projects.

As stated at the outset, there are significant overlaps in the metadata managed by metadata management systems such as the Metabase System and the data managed by project management systems. In addition to the overlap and shared objectives and processes, there is a critical need to have an enterprise perspective on both versus a stove-pipe perspective.

For example, there are not only overlaps in organizational and staff data, the very objects managed by a project management system, the deliverables, which are in fact, IT's deliverables, that is, enterprise architecture components, data models, process models, business information system models, and the like. Because of these overlaps, and because of shared goals and processes, an underway extension of the Metabase system is to have Project Management as a Metabase System module.

The project management module's database's design is depicted in Figure 8. It consists of a number of entities. All these entities are traditional and are interconnected through one-to-many relationships except for those entities that show a one-to-many relationship from the entity to itself. Organization (upper right) contains such a relationship. This relationship means that the entity contains subordinate organizations. For example, an Information Technology organization contains the Information Resource Management organization, which in turn may contain the Data Administration organization, and Database Administration organization.

The eight entities that are recursively are:

- Contract
- Deliverable
- Deliverable Template
- Organization
- Project Template Type
- Resource
- Task
- Task Template

The entities from Figure 8 are also divided into six distinct clusters, which are:

- Contracts, organizations and contract [staff] resources
- Resource and Resource Life Cycle Node
- Project, Deliverable, and Task Templates
- Project Staff

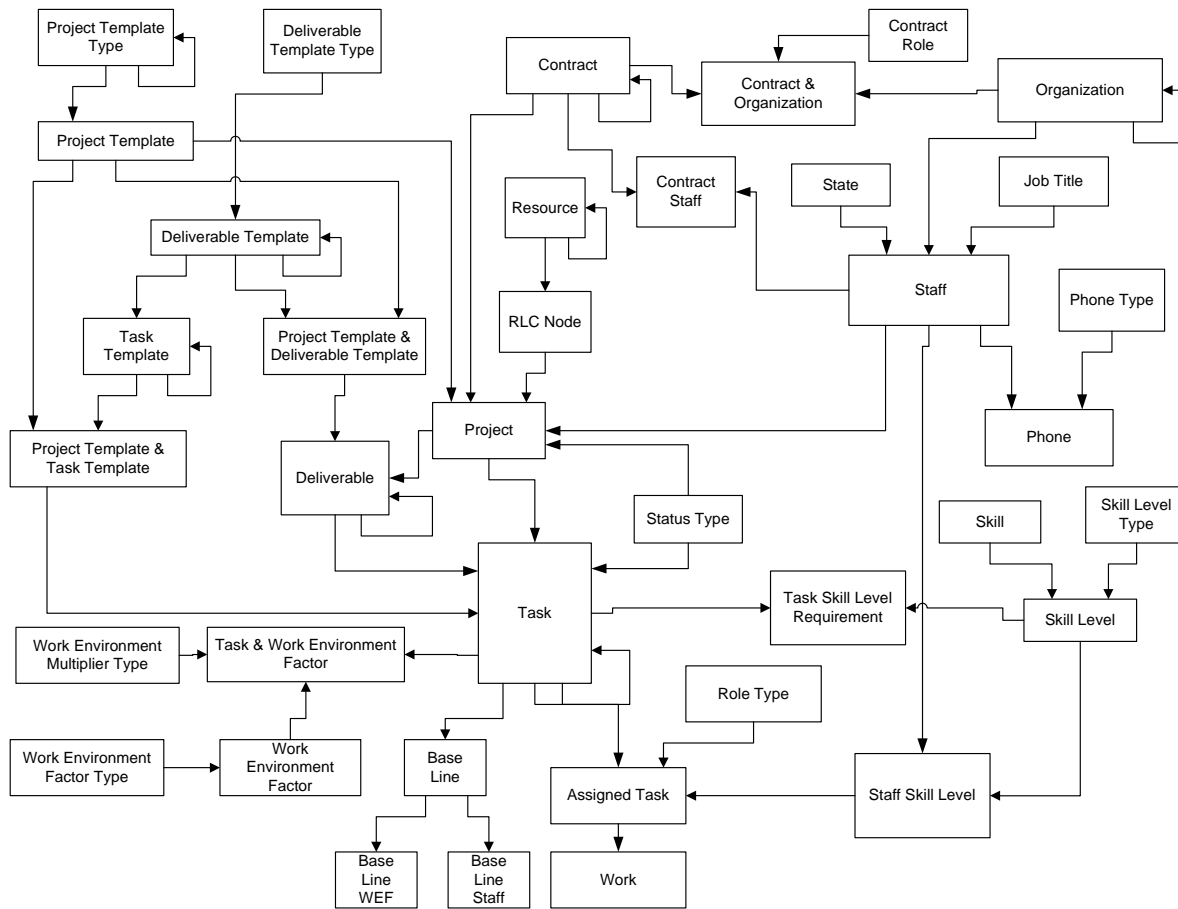


- Project Building and Estimation
- Project Work

In general, the *Contracts, Organizations, and Contract [staff] Resource* cluster of entities represent the environment within which projects take place.

The *Resource and Resource Life Cycle Node*<sup>3</sup> entity represents the target of the project, that is, the area of the business benefitted by the project. For example, for manufacturing, finance, human resources, or land use planning.

The *Projects, Deliverable, and Task Templates* entity cluster enables the definition of the templates employed in the actual building of projects. Defined across the enterprise, these templates enable standard project execution and accomplishment measurement.



**Figure 8.** Data model for a highly engineered project management database.

3

Resource and resource life cycle node has the exact same definition as it does within the Whitmarsh metabase and also the process of Resource Life Cycle Analysis of Ron Ross.





The *Project Staff* entity cluster enables the inclusion of the staff as resources for a contract, and also permit the specification of the specific types and performance ratings of skills that a person may bring to a specific project.

The *Project Building and Estimation* entity cluster represents the entities that support building projects. Projects and associated tasks are initially created through the use of the Project Deliverables, and Tasks Templates. Once projects and associated tasks are created, they are modified by attaching work environment factors and specific skill-level based staff assignments. Only then can task and project resources be computed.

Finally, as task work is accomplished, the *Project Work* entity is valued. As actual work is accomplished, it can be reported through any of its related entities.

### 5.0 Standardized Project and Deliverables Templates

The standardized Project and Deliverables templates involve the following entities from Figure 8:

- Deliverable Template
- Deliverable Template Type
- Project Template
- Project Template and Deliverable Template
- Project Template and Task Template
- Project Template Type
- Task Template

The purpose of these entities is to pre-define classes of projects that are accomplished one or more times. Even when a project is expected to be accomplished only once, it must first be defined as a project template. The reason is simple: No matter how many times you may protest to the contrary, a project is never done only once. Further, the effort to create a project from the template requires the push of only one button. Hence there is no real cost, only a real benefit.

A Project Template can be seen as being within a hierarchical class of projects. This gives rise to the notion of project template types. While clearly the strategy for identifying hierarchical classes of project template types is subjective, the following project template types are illustrative:

- 1 Training
  - 1.01 Needs Assessment
  - 1.02 Development
  - 1.03 Execution
  - 1.04 Evaluation
- 2 Information Technology



- 2.01 Needs Assessment
- 2.02 Tool Development
- 2.03 Component
  - 2.03.01 Requirements
  - 2.03.02 Specification
  - 2.03.03 Implementation
  - 2.03.04 Evolution
- 3 Metabase
  - 3.01 Requirements
  - 3.02 Component
    - 3.02.01 Development

These classes of project templates may mirror the various sections of a mission statement and may take on several generic prefix “forms” such as: plan..., execute..., and evaluate....

Once project template classes are identified, the individual project templates within each class can be identified. For example, the following project templates would be suitable for the project template type, Metabase Component Development.

- Enterprise data architectures
- Data elements
- Specified data models
- Implemented data models
- Operational data models
- Business events
- Database domains
- Missions
- Organizations

The names for these project template examples were all taken from the Knowledge Worker Framework and other Whitemarsh materials. The goal here is to classify the possible types and kinds of projects that are likely to be accomplished within the organization. This classification does not, however, relate to an enterprise’s resource, that is, Finance, Human Resource Development, Distribution, Manufacturing, etc. Rather, these project templates serve as the mechanism to classify projects that address, for example the development of an enterprise data architecture model within the enterprise resource, Human Resources, or the development of an Operational Data Model for a data warehouse data architecture class database, Retail Sales Management.

Associated with each project template there are Deliverable Templates and Task templates. These are independent of Project Templates so that standard Deliverable Templates and Task Templates can be developed and employed within Project Templates. For example, every Project Template that creates a Knowledge Worker product is likely to contain a delivered



product presentation. For any given presentation, the deliverable is likely to be the same: that is, the presentation's materials. Additionally, the work plan to create the presentation is also likely to be the same.

To create the Project Template, Presentation, the Deliverable Template and Task Template merely have to be associated to the Project Template by creating two intersection entity instances, Project Template & Deliverable Template, and Project Template & Task Template. This same multi-use strategy is likely to apply to Project Reviews, Testing Scenarios, User Guide Development, Project Plans, Phase Plans, and the like.

The deliverable template entity is recursive, and thus can contain subordinate deliverable templates. For example, a deliverable template may be the Enterprise Data Architecture. Contained within it would be the Database Domains, Database Objects, Data Elements, and Specified Data Models. Within Specified Data Models would be the Entities, Attributes, Primary Keys, and Foreign Keys. In this regard, the set of meta-entities from the Whitemarsh metabase are clearly first-rate candidates for deliverable templates.

In addition to deliverable templates, there are also deliverable template types that represent the classification of deliverable templates. For example, there might be a deliverable template type such as:

- Process models
- Data architectures
- Business information system architectures
- Plans
- Analyses

Each of the deliverable template types is a way to broadly classify the different types of deliverable templates to which each belongs.

Task templates are recursive and represent the natural hierarchical decomposition of tasks. For example, there might be a task template to Define Database domains. The subtasks within that task template might include Identifying, Describing, Reviewing, and Finalizing the Database Domains. The complete set of tasks and subtasks for the Whitemarsh database project methodology is provided in a Whitemarsh book, *Work Breakdown Structure*.<sup>4</sup>

Collectively, the project, deliverable, and task template entities enable the standard definition of the types of projects that are performed within the enterprise. By standardizing the project types, management and status reports can be created.

Deliverable templates contain an attribute for unit effort hours. This value represents the quantity of staff hours required to complete the deliverable under normal conditions. A

---

<sup>4</sup>

Great care must be exercised as to the level of detail for tasks. If they are too shallow, ambiguity results. If they are too deep, resentment and strangulation occurs. In the Whitemarsh database project methodology, for example, the table of contents of the work breakdown structure represents the right level of detail by which a Whitemarsh database project should be specified and managed. The Table of Contents is 14 pages. The full work breakdown structure is 125 pages.



deliverable, for example might be a defined data element, and the deliverable unit effort for this might be 2 staff hours.

The deliverable template entity also contains another attribute that indicates whether the work associated with producing the deliverable is divisible. For example, it is not likely that the work to create a defined data element is divisible. Therefore, no more than one person should be assigned the task of creating a single defined data element. Similarly, task templates contain an attribute that indicates whether the task is divisible. The task (but not the task template) contains an attribute to indicates the quantity of deliverables to be produced. If the task template indicates that the task is divisible, and if the quantity of delivered units is more than one, then the work is divided as evenly as possible among all assigned staff. The rules for work division are these:

<b>Deliverable [Template] Divisible</b>	<b>Task [Template] Divisible</b>	<b>Task deliverable quantity</b>	<b>Assigned work divided?</b>
No	No	N/A	No
No	Yes	more than one	Yes
Yes	Yes	one	No
Yes	Yes	more than one	Yes
Yes	No	N/A	No

Whenever work is not divisible, the work’s elapsed time is computed to be that of the slowest working person. Other assigned persons are presumed to be “watching” the expert teach or lead the others in the production of the deliverable. Hence, the work proceeds at the rate of the slowest person, not the fastest.

## 6.0 Assessing and Assigning Staff

Figure 8 has the following entities related to assessing and assigning staff:

- Job Title
- Phone
- Phone Type
- Skill
- Skill Level
- Skill Level Type
- Staff
- Staff Skill Levels
- State



The staff entity contains the necessary information to identify persons associated with projects. Staff can be associated with more than one project. The staff member's name, address and phone numbers are entered. In addition to this basic information, hourly costs are stored. These exist as direct hourly costs and indirect hourly costs. Direct hourly costs are typically those associated with salary and benefits. Indirect hourly costs are those associated with assigned overhead such as office space, tools, management and supervision, and the like. When staff costs are computed for a project, both types of hourly costs are employed to produce subtotals and then overall staff cost.

The staff member's skill levels are identified from a list of skills and the skill level types. The skills should relate to the type of activities performed. Skills might include for example:

- Data Administration
- Data Quality Control
- Database Administration
- Documentation
- Logical Database Design
- Management
- Physical Database Design
- Programming
- Quality assurance and quality control
- Requirements analysis
- Systems Analysis
- Teaching
- Testing
- Training program development

With respect to each skill, there can be distinct types that characterize the skill level achieved for the skill. The following skill level types are suggested:

- Expert
- Journeyman
- Novice
- Unskilled

A skill level is a skill of a certain performance type. For example, unskilled data modeler, novice data modeler, journeyman data modeler, or expert data modeler. For each combination of skill and skill level type, a multiplier is affixed. This multiplier field represents an assessment of the relative speed that work can be performed. For example, if the norm is journeyman, the Skill Level Multiplier might be classified as 1.0. If a novice performs the task 30% slower, the Skill Level Multiplier might be 1.3. For an expert who performs the work 20% faster, the Skill Level



Multiplier might be 0.8. Finally, an unskilled worker might take twice as long and would have a Skill Level Multiplier of 2.0.

Once a complete set of skills, skill level types, and skill levels with their multipliers are created, they can be associated with staff. A staff member may have more than one skill, and be an expert at one and a novice at another. Once a person through their staff skill level is assigned to work on a project task, the staff skill level multiplier either speeds up the work, has no effect, or slows it down. This assignment capability provides project managers the ability to explain exactly why two projects that are the same in all respects except for the skill levels of the assigned staff will take two different quantities of staff hours.

## **7.0 Project Plan Constriction and Estimation**

Figure 8 has the following entities related to Project Plan Construction and Estimation:

- Assigned Task
- Base Line
- Deliverable
- Project
- Role Type
- Status Type
- Task
- Task & Work Environment Factor
- Task Skill Level Requirement
- Work Environment Factor
- Work Environment Factor Type
- Work Environment Multiplier Type

To create a project, the following steps must be accomplished:

1. Create the project record
2. Choose the project template
3. Choose the project's status
4. Choose the project's resource
5. Select the project's contract
6. Enter the project's description
7. Choose the project's leader
8. Create basic information for each task
9. Choose the appropriate work environment factor for each task
10. Assign the tasks to project staff members
11. Choose the role that each project staff member performs on a task



12. Press the Compute Project Resources button on the project record
13. Record the project's base line

In Step 1, the project record is created on the screen that contains the list of existing projects. Upon insert, a new [but null] project is created and is presented for further update.

Step 2 requires the selection of the project's project template. A set is presented and one is chosen. Step 3 includes supplying the project's title, and a status that is chosen from the status type entity list. Statuses should proceed through a life cycle. The following sequence of statuses are suggested:

1. Proposed
2. Deleted
3. Approved
4. Scheduled
5. On-going
6. Terminated
7. Completed
8. Reviewed
9. Finalized

Step 4 sets the project within the context of a particular Resource Life Cycle Node. Thus, it is related to a specific collection of databases and business information systems, and exists within the overall life cycle of the Resource.

In Step 5, the appropriate contract under which the task is performed is selected.

In Step 6, a 250 character description of the project can be included. While this seems short, it must be remembered that there are many other name and description attributes that in combination serve to fully describe both the project and its context. Included for example are all the deliverables and their templates, tasks and their templates, contract, and resource.

In Step 7, the project's leader is chosen from a staff list. At this point, the only additional piece of information that can be added at the project level is the project's start date. Once this is entered, all remaining work must be done at the individual task level. This is done in steps 8, 9 and 10.

Once the project's basic information is captured, the project's basic information is essentially complete, and the auto-build of the project can start. Auto-build is the manufacturing step. Project classes are identified, defined, and include deliverables, and unit effort. Once refined, these projects are made into templates. Thereafter, these Project Template are available to manufacture projects.

Auto-build is accomplished by highlighting the newly created project and instigating the project build process. Immediately, all the deliverable template records and task template records associated with the identified project template record are copied into the project's deliverable and task entities. At that point, the lists of deliverables and tasks can be displayed



Once the project's auto-build step is complete, the project's tasks can be updated. This process, that is, Step 8, is started by highlighting the project from within the project list and instigating a change. The project's update window is then presented. Each project task is changed by first highlighting a specific task in the project's task list window and invoking the change function.

Under the strategy set out in this approach, that is, auto generating project details through the use of predefined project, task, and deliverable templates no task may be deleted from, nor can any additional tasks be added to any generated project. If there is additional work in the nature of additional tasks, the project's template must be changed. That is, new tasks added to or existing tasks deleted from that task template from within the project's template.

The ability to add or delete tasks from a project has specifically been removed from Whitemarsh project management because to allow additions or deletions would in effect be defeating the project management strategy: standardization of project specification, accomplishment, and measurement. If task lists derived from a template are allowed to be modified then eventually all templates will be irrelevant as all projects will ultimately be uniquely accomplished.

Great care must therefore be taken to ensure that the task lists at the project template level are sufficiently comprehensive but not too detailed. The exact details of how a task is accomplished should be left to the ingenuity of the person performing the work. That is what distinguishes the skill types, inexperienced, novice, journeyman, and expert. Having tasks at such a level of detail that every five-minutes of effort is both specified and controlled is suffocating, insulting to individual creativity, and absurd. Whitemarsh project management is for knowledge workers, not for process or manufacturing-line workers. That is a profound difference.

As an example, if the project is an IT project that creates a database application, and if the Whitemarsh methodology, which has a 125 page 3300 task work breakdown structure, is employed, these 3300 tasks must not be stored in the project management database. But, if by mistake they were, then it would take forever to estimate such a project and it would also be impossible to manage because the task detail is too low. Instead, only the high level tasks should be stored. This represents about 450 tasks. That's a 8:1 reduction.

Better still is to divide the database project into at least six smaller projects, one for each phase. In that case, the six projects would represent only about 70 tasks each. That's much easier to estimate and to manage. Even within a given phase there can be further subdivisions that makes evolution and maintenance easier. Suppose there were four different approaches to the definition of a database table. Each could be a separate task template with its own set of deliverables. That would allow for the appropriate picking and choosing.

The only tasks that can be changed are those that are "leafs." A request to change a non-leaf task results is an error. In this Step 8, the basic task information can be adjusted. At the outset, the task shows the related task template, deliverable template, and the deliverable template's unit effort hours. This data is automatically brought from the task template. Automatically loaded as well is the task sequence number, whether the task's work is divisible,





and the task's title. The user is able to change the task's title and description to something more stylistic, detailed, or appropriate. Notwithstanding, the task should still be seen as an implementation of a task from the task template.

Entered on the task window is the current status (1=Proposed through 9=Finalized), and the quantity of deliverable units that the task expects to produce. For example, 125 defined data elements. Once the quantity is entered, the task level unit effort hours to be automatically computed. If a quantity of zero is recorded, the task will effectively not participate in the determination of the project's resources. The task's start date should also be entered and as it is used to determine the task's completion date after work environment factors and staff are assigned. Note however, that the task's start date is "overruled" when the schedule for the entire project is computed.

In Step 9, the appropriate work environment factors that govern the tasks are picked. Once picked, the task screen shows chosen work environment factors. When a new work environment factor is needed, an exiting set is presented. Choices should be made that relate to the work environment for that specific task. Each chosen work environment factor has an its effect on the unit effort hours computed for the task.

The staff hours in the task templates assumes normal work environments for each task. While the definition of the work environment factor types and the specific factors is clearly under the control of the project management administrator, the following suggestions are offered:

<b>Factor Name</b>	<b>Factor Value</b>	<b>Factor Description</b>
Reviews conducted by the client	1.00	No effect
	1.00	Reviews, that is, walk thru's are conducted by the client as scheduled
	1.10	Reviews are conducted but by inexperienced reviewers
	1.20	Reviews are generally not conducted
Equipment available for analyst/programmer	1.00	No effect
	1.00	Workstation connected with shared case/metabase environment
	1.10	PC with stand-alone case tool environment.
	1.25	PC with no case/metabase environment
	1.30	For no equipment available to the staff, except through an administrative person
Equipment outages	1.00	No effect



Factor Name	Factor Value	Factor Description
	1.00	If the equipment and all required software is available for at least 6 business hours each day
	1.16	For each hour below the average of six that the equipment is unavailable
Extent of user contact	1.00	No effect
	1.00	If the users are available within half day request to review interim products (work in progress, but not a deliverable).
	1.10	If users are available but only within 2 business days of request
	1.20	If users are available but only within 4 business days of request
	1.30	If users are available but only within 6 business days of request
	1.40	If users are generally not available

In Step 10, staff members are assigned to the task. The list of currently assigned task members appears on the task screen. Changes in these assignments can be accomplished through inserts, changes or deletes.

On an insert, a list of staff members and their associated skills is presented. When a specific staff member is chosen, so also chosen is the staff's skill and skill level multiplier.

Once a staff member has been assigned to a task, Step 11 consists of picking their appropriate role. Suggested roles might be:

- Developer or contributor
- Manager or leader
- Reviewer

Once all the work environment factors and staff have been assigned to a task, the resources associated with that task can be computed. This is done through three processes:

- Compute Factored Hours
- Compute Elapsed Hours
- Compute Completion Date

When the Compute Factored Hours process is executed, the standard hours from the task's task template is multiplied by the multiplicative summation of all the work environment factors producing Factored Hours.

When the Compute Elapsed Hours process is executed, the staff work hours for each assigned staff member is computed as the Factored Hours multiplied by the staff's Skill Level



Multiplier. The elapsed hours for a task is determined as the longest effort for any one participant in a task. The field, staff assigned hours, is the commutative quantity of all staff work hours.

When the Compute Completion Date process is executed, the start date of the task is employed to then compute the completion date according to these rules:

- Duration Days are computed by dividing elapsed hours by hours-per-workday. The Whitemarsh project management software's basic reference data can be changed. It includes, for example, hours-per-workday. If a task has a duration of 24 staff hours and if the hours-per-workday are 6, then the staff-days is equal to 4.
- The total staff-hours for a project is the sum of all the staff hours.
- The completion date depends on whether a task which contains other tasks is marked as serial or parallel.

If the containing task is marked as serial, then the completion date for its contained tasks is computed to be start date for the first contained task plus the duration days for every other contained task. Weekends and holidays are excluded.

If the containing task is marked as parallel, then the completion date for its contained tasks is computed to be the start date for the first contained task plus the duration days for the longest contained task. Weekends and holidays.

If a graphical representation of activities is required, the Whitemarsh task and resource data can be exported via the Microsoft Project export format and be imported into a project management system like Microsoft Project, or other third-party PERT an Gantt chart creation tools.

*Great care must be taken not to try to make project plans perfect. Too much time will have been spent to accomplish a result that will, in just a few days to a week become outdated. Too many internal and external factors cause project plans to become "broken" including such trivial causes such as sick-leave, layoffs, slipped physical product delivery schedules, to say nothing of significant causes such as enterprise mission redirection, change in database management systems, replacement of computing hardware.*

*An analysis of 13 multi-hundred million dollar U.S. Government General Accountability Office IT projects showed that 95% of all late IT projects were caused by errors/changes the occurred outside the control of IT. Thus, getting the overall project plan within 10% is certainly close enough. Additionally, forcing*



*conformance to a perfect-plan that has been autocratically declared not subject to change is simply bad management and bad execution.*

In step 12, the overall project resources and end-dates are computed. Once all the work to each task has been completed, the Compute Project Resources process is executed. At that point, the project's factored hours, staff hours, and elapsed hours are all computed. The start-date and end-dates for each task are recomputed beginning with the project's start date.

Unless otherwise indicated, tasks within a project are processed in a hierarchically serial manner. If a project contains multiple tasks, or if a task contains other tasks, and the schedule-parallel indicator is parallel, then the contained tasks within the project or within specific tasks are scheduled in parallel. That is, the elapsed time for a containing project or task is the elapsed time of the longest contained tasks. That set of tasks becomes the critical path. Once all the scheduling is accomplished, the project's end date is computed.

Once a project is scheduled, the Check Overload process can be executed. This process triggers an analysis that determines if any staff member is overloaded for a task. It determines this by summing the staff members assigned hours across task schedules. If the sum is greater than the allowed work hours for the period, then the overload indicator is set. A staffing allocation report is generated and it shows the project and task level of staff loading by staff person across time. The granularity of time used to assess overloading is one week.

Step 13 enables the project manager to record a baseline for the project. A baseline is the dated permanent recording of key project, task, resource and schedule information. When the baseline is recorded, a new set of the information is created for every task in the project. Collected as well is the key information from the work environment factors and the staff assignments on the date of the baseline.

Once project planning is finished, the project's baseline is created by invoking the Baseline process on the project update window. The data values for all of the above attributes are collected on a task by task basis and stored into the baseline entity. Baselines support progress reporting. Since a key attribute of the baseline entity is the baseline date, multiple baselines can be created and the baseline information can be described and tracked during the life of the project.

## **8.0 Recording Work and Generating Earned Value Reporting**

Figure 8 has only one entity, Work that is directly related to the recording of work and supporting earned value reporting. This entity records the hours expended by an assigned staff person including a description of the work performed. Work is recorded in terms of the assigned task. A window that contains four interlinked lists is presented. The first list contains the projects. The second, linked with projects is the contained tasks. The third is the task's contained assignments, and finally the fourth contains the work records. If a new work record is to be



added, the Insert button is pressed. The recorded work includes the start and end dates, hours worked, quantity of deliverable units completed, and the description of the effort.

As a task is completed, its status should be changed. As a project is completed, the project's status should also be changed. Reports perform project accomplishment analysis by schedule, status, hours, and earned value. Earned value analysis computes the percent of a task being complete based on the deliverable units accomplished. If 100 units are proposed and 50 are complete, then percent complete is judged to be 50%. Earned value date adjustments mean that the quantity of hours expended by summing work hours to complete a task results in projections of the hours needed to complete the task. When earned value rescheduling is performed the task's end-date is recomputed. On the project level, earned value analysis and rescheduling accomplishes the rescheduling of the entire project. This results in new task and project dates.

## **9.0 Conclusions**

The enterprise-level project management approach in this paper clearly enables:

- Concurrently managing disjoint projects
- Managing generally uncontrolled resources
- Enabling maximum re-use of past efforts
- Incorporating learned experiences
- Not requiring a full-time project planner
- Supporting what-if resource allocation scenarios
- Enabling management to know about and view all projects and resources across the enterprise
- Supporting the presentation of projects individually, or from the perspective of a business-defined set of priorities
- Enabling multiple, concurrent, but differently scheduled projects against the same enterprise resource<sup>5</sup>

---

<sup>5</sup>

An enterprise resource represents an essential component of the business. Resources provide the business context for projects. For example, staff, money, contracts, equipment, and facilities. More  
(continued...)



- Supporting single projects that affect multiple enterprise resources
- Permitting projects that develop completely new capabilities, or changes to existing capabilities within enterprise resources

Because Whitemarsh project management system is implemented as a database application, it supports the following types of reports:

- Projects and project statistics of a certain project template
- Projects and project statistics within certain [business area] resources
- Projects and project statistics by deliverable types
- Projects and project statistics by organizational units
- Projects and project statistics by specific project staff members
- Projects and project statistics by certain types of skills
- Projects and project statistics according to certain status types
- Projects and project statistics according to certain work environment factors

## 10.0 References

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

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:

Paper	URL
Comprehensive Metadata Management	<a href="http://www.wiscorp.com/ComprehensiveMetadataManagement.pdf">http://www.wiscorp.com/ComprehensiveMetadataManagement.pdf</a>
Metabase Overview	<a href="http://www.wiscorp.com/Metabase.zip">http://www.wiscorp.com/Metabase.zip</a>
Whitemarsh Data Modeler, Architecture and Concept of Operations	<a href="http://www.wiscorp.com/MetabaseDataModelerArchitectureandConceptofOperations.zip">http://www.wiscorp.com/MetabaseDataModelerArchitectureandConceptofOperations.zip</a>

---

<sup>5</sup>(...continued)

on resources, resource life cycles and resource life cycle networks is found in the Whitemarsh Knowledge Worker and Information Systems planning books, papers, and courses.



## Manufacturing Project Plans

---

Information Systems Planning: Book, Course, and Presentation (short and long) – samples	<a href="http://www.wiscorp.com/EnterpriseDatabase.htm">http://www.wiscorp.com/EnterpriseDatabase.htm</a>
Knowledge Worker Framework: Book, Course, and Presentation (short and long) – samples	
Database Architecture Classes: sample	<a href="http://www.wiscorp.com/DatabaseDesign.htm">http://www.wiscorp.com/DatabaseDesign.htm</a>
Resource Life Cycle Analysis: Paper	<a href="http://www.wiscorp.com/MetabaseProducts.htm">http://www.wiscorp.com/MetabaseProducts.htm</a>
Database Project Work Breakdown Structure – sample	<a href="http://www.wiscorp.com/DatabaseProjects.htm">http://www.wiscorp.com/DatabaseProjects.htm</a>
Resource Life Cycle Analysis Metabase Module User Guide	<a href="http://www.wiscorp.com/metabase_demo.html">http://www.wiscorp.com/metabase_demo.html</a>
Metabase System (Free Version) Request form	<a href="http://www.wiscorp.com/freemb.html">http://www.wiscorp.com/freemb.html</a>

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.

Paper	URL
Data Management Program - Metadata Architecture For Data Sharing	<a href="http://www.wiscorp.com/wwmembr/mbr_products_edb.html">http://www.wiscorp.com/wwmembr/mbr_products_edb.html</a>
Data Management Program - Database Interface Architectures	
Data Management Program - Projects And Data-Asset Product Specifications	
Data Management Program - Work Breakdown Structures	
Knowledge Worker Framework Database Objects	
Managing Database - Four Critical Factors	
Data Architecture Classes	<a href="http://www.wiscorp.com/wwmembr/mbr_products_dd.html">http://www.wiscorp.com/wwmembr/mbr_products_dd.html</a>
Guidelines for Data Architecture Class - Data Warehouse	
Iterations of Database Design	



## Manufacturing Project Plans

Paper	URL
Work Breakdown Structures Database Project Work plan Templates Information Systems Development Methodology Phases 1 and 2 Whitemarsh Project Estimating Work plan Development	<a href="http://www.wiscorp.com/wwmembr/mbr_products_dp.html">http://www.wiscorp.com/wwmembr/mbr_products_dp.html</a>
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  Knowledge Worker Framework Database Objects  Managing Database - Four Critical Factors	<a href="http://www.wiscorp.com/wwmembr/mbr_products_edb.html">http://www.wiscorp.com/wwmembr/mbr_products_edb.html</a>
Information Systems Planning: Book, Course, and Presentation (short and long)  Knowledge Worker Framework: Book, Course, and Presentation (short and long)	<a href="http://www.wiscorp.com/wwmembr/mbr_products_edb.html">http://www.wiscorp.com/wwmembr/mbr_products_edb.html</a>

