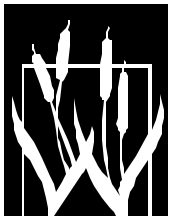

Whitemarsh
Information Systems Corporation

Whitemarsh Project Management: Architecture and Concept of Operations

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

Table of Contents

1.0 Project Management in the Knowledge Worker Environment

Whitemarsh project management is designed to help enterprises manage knowledge worker products individually or an enterprise-wide basis. Figure 1 presents the Whitemarsh Knowledge Worker Framework. Any cell from the framework is likely to spawn many projects to engineer, develop, execute, evolve, or evaluate the knowledge worker products implied by the cells.

The examples in this paper are drawn from Information Technology (IT) projects, which naturally are within the Knowledge Worker Framework's "machine" columns. Specifically, IT projects exist within the Systems through Operations rows of the Database Objects and Business Information Systems columns. There is nothing, however, within Whitemarsh project management that restricts its use to these cells or these columns. All projects, regardless of the Knowledge Worker cell from which the project proceeds, have deliverables, task lists, staff assigned, and schedules. And, without proper project management, all knowledge worker projects exhibit the same characteristics: over budget, under specified, delivered late, and failing to meet organizational expectations.

1.1 Why Project Management is Important

Project Management is important because almost all enterprises suffer from one or more of the following problems:

- Inaccurate estimates
- Conflicting priorities among projects
- Inability to deal with varying levels of work conditions, staff skills, and the like
- No intra- and inter-project reporting

Simply put, a common lament is that while there are projects everywhere, the ability to effectively manage these projects on an individual or enterprise-wide basis is nowhere.

For example, studies[1] [2] [3] by have shown that many, if not most, knowledge worker projects exhibit these characteristics: over budget, under specified, delivered late, and fail to

---

[1] The Standish Group. *CHAOS: 1998: A Summary Review:* 1999. The Standish Group International, Dennis, MA 02638.

[2] Matson, Eric. Speed Kills (the Competition). Fast Company (August 1996). Page 1. Web: www.fastcompany.com/online/04/speed.html.

[3] Strassmann, Paul A. *40 Years of IT History*: 1997. Page 6. Web: www.strassmann.com/pubs/datamation1097/index.html

meet organizational expectations. While not all reasons for failure can be laid at the foot of project management, too many can. Among the underlying reasons are invalid work plans, insufficient time for requirements changes, and inexperienced or mis-allocated staff resources.

Whitemarsh Knowledge Worker Framework

		Man-Machine Interface				
		Machine		Interface	Man	
Deliv-erables	**Mission**	**Database Object**	**Business Information System**	**Business Event**	**Business Function**	**Organizatio n**
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
Busine ss	Mission hierarchies	Resource Life Cycles	Information sequencing and hierarchies	Event sequencing and hierarchies	Business scenario sequencing and hierarchies	Organization charts, jobs and descriptions
System	Policy hierarchies	Database object models and specified data models	Information system designs	Invocation protocols, input and output data, and messages	Best practices, quality measures and accomplishme nt assessments	Job roles, responsibilities , and activity schedules
Tech-nology	Policy execution enforcemen t	Implemented data models	Information systems application designs	Presentation layer information system instigators	Activity sequences to accomplish business scenarios	Procedure manuals, task lists, quality measures and assessments
Deploy-ment	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
Oper-ations	Operating business	Application interface data models	Operating information systems	Start, stop, and messages	Detailed procedure based instructions	Daily activity executions, and assessments

Figure 1. Knowledge Worker Framework

The United States Government' General Accounting Office (GAO) has been studying IT projects[4] [5] for a number of years, and a review of 10 GAO studies clearly shows that the main reasons why IT systems fail has nothing to do with IT. Again, while not all reasons are specifically related to project management, some of the reasons have to do with critical components of project management. And again, these are invalid work plans, insufficient time for requirements changes, and inexperienced or mis-allocated staff resources.

In response to the need for project planning and management, methods and computer based project management systems have been created. PERT (Program Evaluation Review Technique) was created and employed within the United States Department of Defense in the early 1960s. The PERT chart supplemented the (Henry) Gantt chart and the Critical Path Method (CPM). The PERT chart is typified by a left-to-right network of activities that enforce sequence and precedence. The Gantt chart is a top-to-bottom hierarchical list of activities set against time. A CPM chart is a subset of activities from a PERT chart that takes the longest time. Hence, the critical path. Some Gantt charts simulate PERT charts by serializing some activities. Additionally, once a PERT chart is resource loaded, it can be displayed left-to-right against a time-scale.

In the 1960s through the1970s, project management software operated on large mainframe computers and supported the planning and management of very expensive projects such as Naval ships, bridges, highways, and large IT projects. Because of the project's size, one or more full-time project planners were justified.

With the advent of the PC in the early 1980s, PC packages such has Harvard Project Manager arose. This brought project management to the small project. However, few projects could afford full-time project planners. Because the underlying project planning methods had not changed, and because every project was seen as a one-off effort, PC based project management had become a most dreaded activity.

As a consequence of market pressures and corporate mergers, two classes of project management systems remain today:

- PC based or low-end packages
- Server based or high-end packages

PC based project management systems are typified by Microsoft's Project (www.microsoft.com) or Time Line Solution's product, Time Line (www.tlsolutions.com). Server based project management systems are typified by Primavera (www.primavera.com) and Welcom Software (www.welcom.com).

---

[4] Gorman, Michael M. Knowledge Worker Framework: 1999. Web: www.wiscorp.com

[5] United States Government Accounting Office. Managing Technology: Best Practices Can Improve Performance and Produce Results,: 1997 (GAO/T-AIMD-97-38). Washington, D.C. (web: www.gao.gov)

While the high-end packages are designed for very large, complex project's of thousands of nodes, and while the low-end packages are well suited for scheduling a single project of relatively simple complexity, both the high end and low end solutions do not really address the problems associated with:

- Disjoint projects
- Management of generally uncontrolled resources
- Repeatability of projects
- Incorporation of learned experience into the project estimation cycle

Many knowledge worker projects involve persons from within different organizations over whose time the project manager may not have direct control. Thus, the best the project manager can do is to request participation and to create approximate schedules that show deliverables from these non-controlled participants.

If the knowledge worker project manager creates elaborate project schedules based on many layers of intricately crafted activity networks, then while they look magnificent the instant they are first created, these project plans cannot withstand assaults from all the schedule conflicts. Once these assaults are underway, the project manager has to continuously adjust the layers of project activity networks, resource estimates, parallel and serial paths, etc. Soon the project manager's life is consumed by project management rather than project accomplishment. The dilemma then becomes:

- Accomplish the project, or
- Plan the project's accomplishment.

All too often, project planning is discarded because the project management system, initially thought to be the savior from chaos actually had become another source of chaos. The castle of project management becomes the project manager's dungeon wherein time is the dungeon master, the PERT chart is the shackles, and the schedule is the rack.

To be successful at Knowledge Worker project management, an approach must:

- Concurrently manage disjoint projects
- Manage generally uncontrolled resources
- Enable maximum re-use of past efforts
- Incorporate learned experiences
- Not require a full-time project planner
- Support what-if resource allocation scenarios
- Enable management to know about and view all projects and resources across the enterprise
- Support the presentation of projects individually, or from the perspective of a business-defined set of priorities

1.2 Continuous Flow Process

IT projects are accomplished within distinct development environments. The two most common are: discrete project and release. The discrete project environment is typified by completely encapsulated projects accomplished through a water-fall methodology.

In release environments, there are a number of different projects underway by different organizations and staff of varying skill levels over which project managers may not exert draconian control. Once a large number of projects are underway, the ability of the enterprise to know about and manage all the different projects degrades rapidly. That is because the project management environment has been transformed from discrete encapsulated projects into a continuous flow process of product or functionality improvements that are released on a set time schedule. Figure 2 illustrates the continuous flow process environment that supports releases. The continuous flow process environment is characterized by:

- Multiple, concurrent, but differently scheduled projects against the same enterprise resource[6]
- Single projects that affect multiple enterprise resources
- Projects that develop completely new capabilities, or changes to existing capabilities within enterprise resources

There are four major sets of activities within the continuous flow process environment. The user/client is represented at the top in the small rectangular box. Each of the ellipses represents an activity targeted to a specific need. The four basic needs are:

- Need Identification
- Need Assessment
- Design
- Deployment

The box in the center is the metabase. Specification and impact analysis is represented through the left two processes. Implementation design and accomplishment is represented by the right two processes. Two key characteristics should be immediately apparent. First, unlike the water-fall

[6] 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 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.

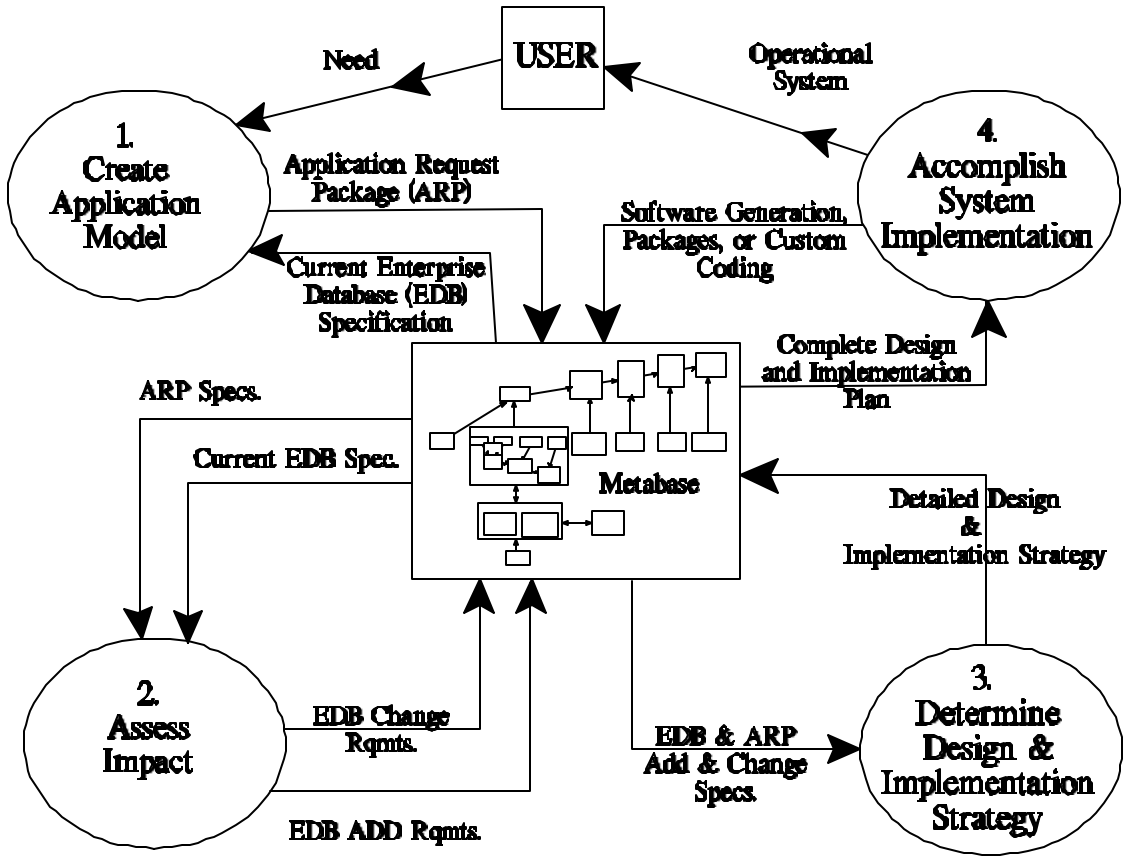

Figure 2. Continuous Flow Process

approach, the activities do not flow one to the other. They are disjoint. In fact, they may be done by different teams, on different time schedules, and involve different quantities of products under management. In short, these four activities are independent one from the other. Their only interdependence is through the metabase.

The second characteristic flows from the first. Because these four activities are independent one from the other, the enterprise evolves by means of releases rather than through whole systems. If it evolved through whole systems, then the four activities would be connected either in a waterfall or a spiral approach, and the enterprise would be evolving through major upgrades to encapsulated functionality within specific business resources. In contrast, the release approach causes coordinated sets of changes to multiple business resources to be placed into production. This causes simultaneous, enterprise-wide capability upgrades across multiple business resources.

In the first process, Create Application Model, the user provides requirements to an application's analyst, who interrogates the existing metabase about the existing enterprise resource. If the user's request can be handled by existing enterprise features, it is. Otherwise, the application's

analyst formulates the requirement into an application request package (ARP). The application request package is represented as a need and includes a series of brief requirement statements in terms of additions to or modifications of an existing enterprise database.

Because this is a continuous-flow model, an application request package may be handled by either an existing enterprise resource capability, or those in implementation (process 4), or, in time, through process 3 (design) or process 2 (impact assessment).

At predetermined intervals, for example, monthly or quarterly, all the application request packages are bought into the second process, Assess Impact. During this process, all the application request packages are examined in unison against the then current enterprise resource specifications. Because of the continuous-flow nature of this process, the enterprise resource specification may have changed during the time when a set of application request packages is being batched. When the enterprise resource add or change requirements are formulated, they reflect the state of the enterprise resources at the time it is to be changed. The form of an enterprise resource add and/or change requirements package is similar to that of the application request package. When the impact of a change is assessed, there may also be changes to existing facilities. These changes must be accomplished as well. Thus, when the final enterprise resource add or change package is developed, it contains these additional change requirements.

The third process, Determine Design & Implementation Strategy, determines the actual detailed specifications of the systems changes. The detailed design of the change is in the form of specific changes to existing enterprise systems (manual, automated, or mixed).

The final process, Accomplish System Implementation, performs all the normal implementation activities. At the end of test and integration, the release is accomplished.

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.
- The 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.

1.3 Relationship to Other Metabase Components

Figure 3 presents a high level diagram of the domain of the metabase. As illustrated by the diagram, persons, through their role within an organization perform functions in the accomplishment of enterprise missions, have information needs. These information needs reflect the state of certain enterprise resources such as finance, people, and products that are known to the enterprises. The states are created through business information systems and databases. Prior to computers, business information systems and databases were entirely manual. Today, most, if not all are automated in some manner.

Knowledge work is accomplished in support of the enterprise's missions. Projects cause knowledge work to be accomplished. Some enterprises accomplish all knowledge work through projects. Other enterprises employ projects only to establish or change knowledge work environments. In either case, Whitemarsh project management can greatly enhance the ability to have efficient, reliable, and repeatable projects.

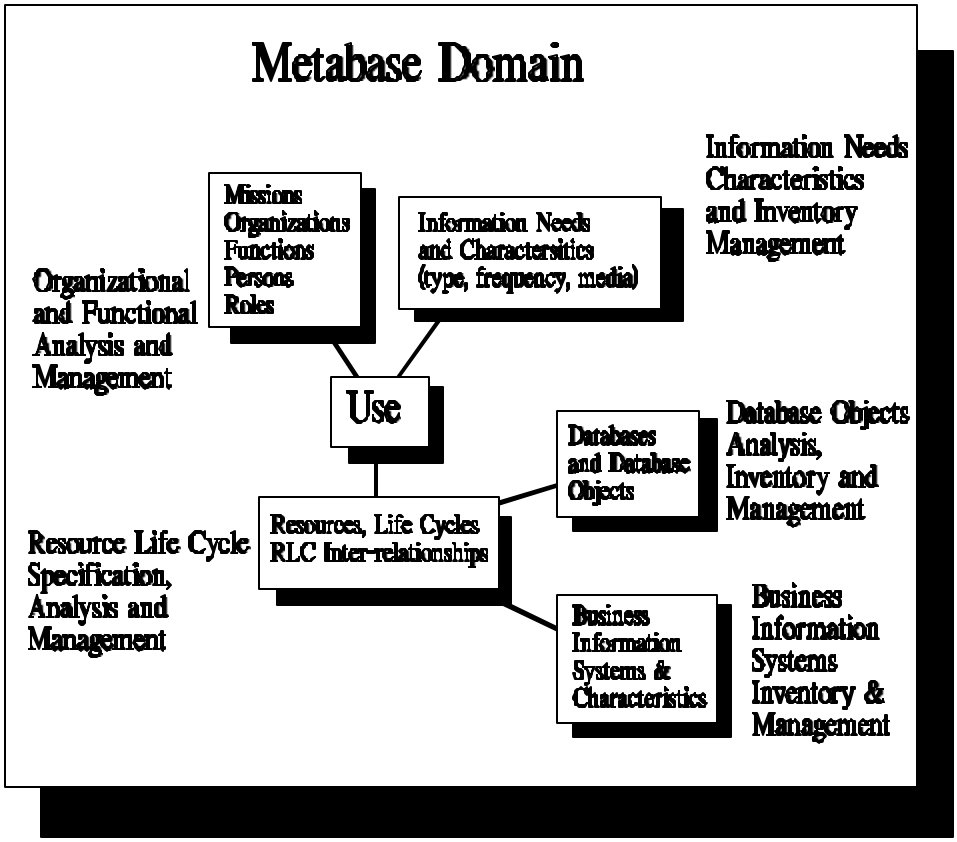

Figure 3. Overall Metabase Environment

1.4 Whitemarsh Project Management Environment

Whitemarsh project management is based first and foremost on its database design. The general "life cycle" of Whitemarsh project management is:

- Employ project, deliverable, and task templates to plan projects
- Plan and estimate projects in a gross way and accommodate different work environment factors
- Staff projects and generate schedules
- Record progress towards deliverable accomplishment
- Re-plan projects as needed
- "Learn" from actual durations from accomplished deliverables

Whitemarsh project management does not, however, support the creation of:

- Very precise parallel and serial networking of projects or tasks
- Very detailed and precise scheduling
- Gantt, PERT or CPM diagram production

These three activities are the proper activities of both low-end and high-end project management systems. In support of these systems, the Whitemarsh project management system generates output data files that can be input into these systems. These systems can then be used to create very precise schedules, activity diagrams and the like. Whitemarsh holds however, the proposition that if accomplishing these three activities were THE basis for successful project management then there would be no need for Whitemarsh project management. While very precise schedules, activity diagrams and the like are important, it seems clear that these features have little or nothing to do with project management success.

In contrast, Whitemarsh believes that project management success is predicated on different activities, which are:

- Continuous optimization of repeatable projects ,
- Accommodation of various work environments and factors within these environments,

- Adjustment of project schedules based on differing staff and skill levels, and
- Capturing actual work accomplishment metrics that support earned value analysis and reporting.

The project management database design employed by Whitemarsh has been implemented several times as the basis for project management over the past 10 years. Whitemarsh believes therefore that the "design bugs" are worked out. Whitemarsh project management serves the need of the independent project manager who has to accomplish the definition, management and reporting of diverse and possibly disjoint projects with staff of varying skill levels within mixed work environments that are generally not within direct control. Whitemarsh believes that this type of knowledge worker environment is the rule, not the exception.

2.0 Whitemarsh Project Management, A Difference in Kind

A key difference between the Whitemarsh project management approach and others is that the Whitemarsh approach concentrates on the management of "nouns" while other project management approaches focus on the management of "verbs." Clearly, since there is no one sacred, perfect way to produce a deliverable (i.e., the nouns), if the focus of project management is to identify and control the "methods" (i.e., the verbs) by which deliverables are produced, then to have enterprise-wide project management and/or to have enterprise-wide metrics, the enterprise must first carve-into-stone the processes by which work is done. Not only is this impossible, it is highly undesirable. It is impossible because it is inconceivable that there is only one way to accomplish any product. It is undesirable because it is insulting to project staffs to presume to control their every technique, process and step. Not only can't it be done, no one will allow it to be done.

In contrast to managing "verbs," Whitemarsh project management manages "nouns," It does this by collecting the quantities of resources expended to produce deliverables. Whitemarsh project estimates are therefore based on the staff hours required to produce deliverables rather than to accomplish tasks.

This technique enables different styles of project management to be employed or be set one against the other by comparing the resources expended to produce deliverables. There might be one project template for mainframe development, another for micros, and finally a methodology for web-based systems even though all the deliverables might be essentially the same. Alternatively, there might be multiple project templates that produce the same set of deliverables to serve the needs of different styles or techniques as might be the case for the data-driven and process driven approaches.

Additionally, the Whitemarsh project management approach enables enterprise-wide project reporting in terms of the cost and effort to produce deliverables versus the accomplishment of activities. As work techniques improve, either through the increased skill of staff, or through the adoption of different techniques, the efforts remain comparable because it is the quantity of resources expended to produce the deliverables that are compared rather than the activities, which are no longer able to be compared because they are now different, that produce the deliverables.

To illustrate, when you go into a grocery store and buy an apple, the cost is expressed in terms of the product you are buying, the apple. While you may wonder how much the various activities cost that ultimately produced the apple, fundamentally, you probably do not care. When you go to five different stores and compare the cost of apples (given a standard for equating quality), again you are only comparing the cost of the deliverable, the apple. If one store spends 10% for transportation and another spends 8%, you probably don't care. It's the final cost of the apple that matters, nothing else. So also should it be with project management. The only thing that should matter is the final cost of the deliverable. Nothing more, and nothing less.

If however you are a wholesale apple buyer that deals with a co-operative and by contract, you have to pay every apple grower the maximum cost incurred by any one member of the cooperative, then you have a real incentive to look "behind" the costs of the deliverables (the apples)

to find the different underlying processes that make the costs different. Even then, the goal then is to find the lowest-cost set of activities, and to then highly recommend that set of activities to all members of the cooperative so that your costs for the deliverable–as a buyer–will go down. So, while there may be an interest in activity-sets, they are not the driving force. So too with Whitemarsh project management wherein the cost of deliverables rather than the cost of methods is the driving force.

Whitemarsh project management enables the melding of the project templates by combining selected:

- Task templates–that is, the enterprise's techniques, methods or work breakdown structures that have been proven of the years to accomplish work the most cost effective manner.

- Deliverable templates–that is, the enterprise's specifications of and unit effort metrics required to accomplish the components of its Knowledge Worker products.

The resulting Project Templates are then specially tuned into "real" projects by determining the quantity deliverables, and then affecting the resulting "norm" estimates through:

- Work environment factors–that is, the effects from varied work environments on the creation of deliverables according to certain task templates.

- Staff–that is, the effects from persons and their varying types and degrees of skills on the rate of production of deliverables according to the task templates.

Collectively, these four project management components are an exemplary use of the database fundamental, *define once, use many times*. Whitemarsh believes it has achieved the ability to have maximum reuse with minimum original, one-off effort.