"Everybody knows how to find the facts that prove the conclusion they have already reached."

Peter Drucker

Good Project Management – Good Management

By: Gary Rinehart
Decision Interface

White Paper – August 2005

Table of Comments

Introduction

Managing projects is at the heart of much of the work of the information technology (IT) organization. New software, new hardware, new systems, new tools and new delivery approaches are all accomplished through projects. Yet the state of the art in many organizations for this fundamental skill is very low. Often, lack of such a fundamental skill is stonewalled because to admit to needing better project management is like the sailor admitting to a fear of the sea. Many are, few confess.

World class information technology organizations get things done through projects – consciously!

Projects come in a variety of forms and shapes.

Replacing the accounting system installed in 1987 or upgrading the central computer site are projects with a lot of structure, a known and expected result and a highly predictable end-date.

Working with the marketing organization to improve the effectiveness of advertising and promotion campaigns is a project with an outcome difficult to predict and a schedule basically unknown even to those involved.

Projects differ also in visibility. A new accounting system will be visible to many parts of the organization. The upgrade of the computer system should be generally a nonevent to all but the team who worked three weekends and a holiday to assure the "effortless" change.

Project Management -- Step by Step

There are six steps to any project:

- Defining what "done" is before you begin
- Building a team that can succeed
- Estimating the resources required
- Planning the tasks -- in short intervals
- Building a control and visibility process
- Defining a clear procedure for completion

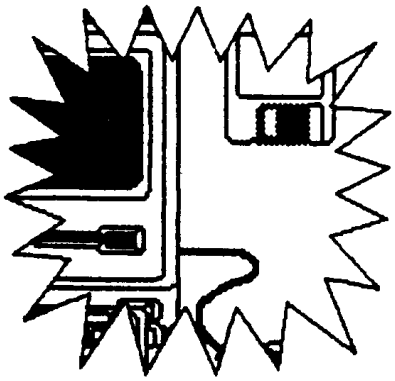
For a simple, highly structured, low technology project involving only a few people and having a short duration, one pass through these six steps is usually enough. "Build a wall in the computer room to separate the printers from the rest of the equipment" would be an example of a one-pass project.

"Total renovation of the applications portfolio" is an example of a project that might require multiple passes through the six steps. The first pass might be a project to define just what was meant by "total renovation."

A project isn't a project just because someone says it is -- at least not in terms of planning a project and the application of the six basic steps in developing the plan. Project managers need to logically subdivide large efforts into projects that have a high probability of meeting schedule, budget and specification.

Before going much further, a brief definition of these six steps is in order:

1. Defining Done before You Begin



It is nearly impossible to be successful if you do not have a grasp on the details of success. What does success look like? You, the project manager, and others must have a clear understanding of "done" so success will be recognized when it is accomplished.

For simple projects, "done" is the completion of the total objective. "Install a new wall in the computer room." For more complex projects, we are tempted to define "done" as the total renovation of all the applications in the portfolio.

Guess what that specification will look like! Many trees will be sacrificed to produce a document attempting to describe renovation of all applications. Many weeks or months will pass before such a specification can be prepared, and more yet before real work can start.

"Done" for complex projects is better defined as a series of much smaller "dones." Make the first sub-project define just what renovation means and which applications are included.

2. Building a Team That Can Succeed

Projects are executed through teams. Team members are the sole resources of the project, and are therefore the key indicator of success. Good teams are successful most of the time. (Research has shown that teams that have the respect of the organization are successful four times more frequently than teams that lack organizational respect.)

Team members must come from those organizations that have a vested interest in the project. The selection of a team member sends a strong message regarding importance of the client or user to the project team and to the IT organization.

The project manager also must be carefully chosen and must fit the nature of the project. Technical skill and the ability to talk with the technical staff are keys for a project that depends on new technology. "Boardroom" skills and the ability to translate the project status and objectives into multiple frames of reference (gestalts) are vital for the project manager of a highly visible, low structure project (such as implementing a Business Intelligence data mart).

3. Estimating the Resources Required

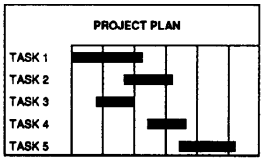
If executives understand nothing else about information systems, they understand that information systems have the potential of consuming resources quickly and often without a trace of benefit.

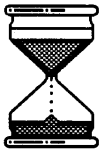
The successful project team understands the resources required to be successful. The successful project team also understands that a contingency fund or "kitty" is also vital to success. Projects change. Changes seldom reduce cost.

The resource estimates come in at least four categories: personnel costs, capital costs, supplies and services costs and, finally, external contract costs. Personnel costs are the salaries and related expenses of having a person work on your project. Capital costs are those involved in acquiring computers, buildings and other expensive, physical assets. Supplies and services would cover computer time, communications costs, paper, pencils and other "everyday" costs of doing business. Contract costs are those that involve obtaining significant material or service items from outside contractors, specifically for the project. A custom computer program, a customized communications processor or a specialist in the design of ERP systems are examples of contract items.

Avoid the temptation to assume some crucial skill or material is free ("it is part of overhead," "she will work on our project and not charge her time" or "we will use the surplus raised flooring from the old engineering computer room"). Butterflies may be free, but crucial project resources that seem to be free during planning time always turn out to be available only at full cost when it is implementation time. There is a reason for the adage "you get what you pay for."

4. Plan the Tasks -- in Short Intervals





Complex projects are sets of less complex projects. Taking this idea further, until the decomposition of the project reaches the point of a single kind of work or work done by an individual is the objective of this step. This step breaks the project described as "total renovation of the applications portfolio" down until tasks such as "develop a data model of project invoicing" are reached.

Then we develop some quantitative measurements or progress indicators to this task so that later we can denote progress and be assured it is moving towards "done." My bias is to never have a task last longer than two weeks without a quantitative measure. "Invoice data requirements" report might be a mid-task measurement leading to "product invoicing data model submitted to data administration" as a measure of this task being "done." With two-week intervals between measurements, it is hard to be surprised by more than a two-week slip in the project. Oracle's implementation methodology takes

the task level to a specific deliverable. For each task, a deliverable must be completed before that task is considered complete.

5. Building a Control and Visibility Process

Having defined "done," built a team, estimated the costs and planned the tasks, the hard work begins. A marketing expert was quoted as saying, "No product plan withstands the first contact with the market." So it is for project plans. Team members change jobs, personnel costs escalate, requirements shift, free becomes full cost, the purchased application is full of "vapor-ware," and schedules are delayed.

Taken in small increments, none of these "normal" events mortally wound a project. If left to accumulate, even the smallest deviations from plan eventually add up to huge delays and cause projects to be considered failures.

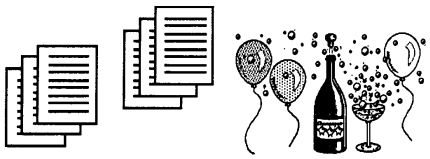
The control and visibility process involves meetings of the team (my bias is weekly meetings) and written reports of planned progress versus actual progress, planned resource consumption versus actual consumption and adherence to specifications (and expectations).

Each deviation is considered, and the impact on the project is determined. Proposed changes to the project are treated as binary events, either in or out, no middle ground.

As changes start to be made to the project, the value of the "kitty" starts to be realized. Uncommitted resources now are committed. Impact on the overall cost and schedule, as seen externally, are minimal. Team confidence increases as the changes are handled without serious compromise to the overall definition of a "done" project.

Never manage a project without a "kitty"!!

6. Defining a Clear Procedure for Completion



Finishing a project involves three elements: a definition of what "done" is (prepared in the first step and kept up to date along the way), a procedure for demonstrating that the project has met these requirements, and agreement upon who has the authority to agree that the requirements have been met.

This last step is so often the fatal element in what is otherwise a successful project. Often the technical side of the project team is given responsibility to develop the details of the demonstration of compliance to specification.

The tests of compliance are met and yet the project is not accepted by the client or the users. The project languishes at this "almost" state until it fades away. Why?

Different gestalts!

The authority for acceptance was incorrectly given to the individual verifying that the test plan was successfully accomplished. That the test plan was irrelevant to those with the real authority (the client) was never considered.

Clients are often far easier to please than technicians. If the new system meets their expectations (or is only just better than what they are now using), success is declared. The customer is almost always right, if for no other reason than the authority of being the customer (client or user).

Project Artifacts

The dictionary defines an ***artifact*** as evidence of human skills "usually of a primitive nature."

Projects need a set of artifacts. Not only do the teams working on the projects need some tangible evidence of progress, but also the artifacts should provide the management and "significant others" with confidence that their good is being served.

Project documentation needs four elements:

- A spec
- A plan
- A schedule
- Progress reports

The plan for a highly visible and highly structured project often is quite large. It would

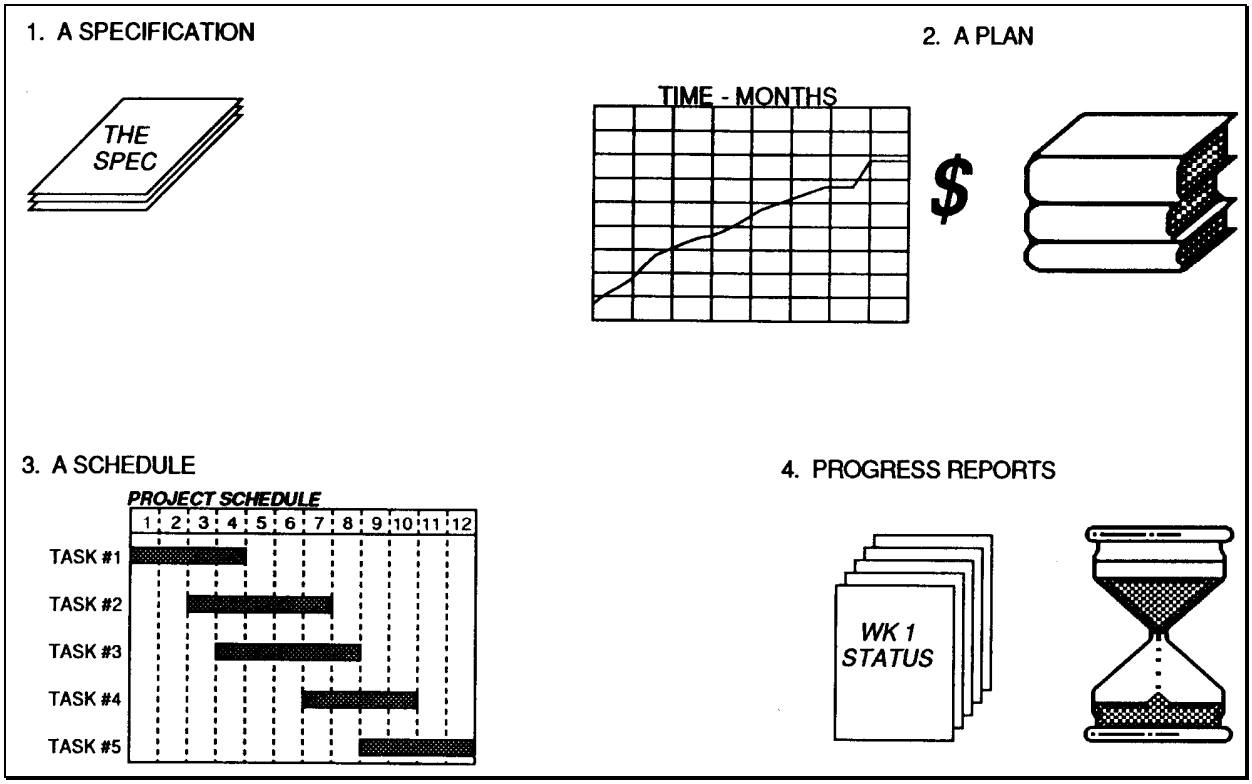


Figure 1: Project artifacts

include a detailed set of tasks, a highly detailed description of the resources required, and a comprehensive test and demonstration plan, along with a full description of the impacts on the organization.

As the project becomes less structured and the technology becomes a greater unknown, the plan becomes less precise and necessarily smaller in volume.

The specification follows the same size patterns as the plan (and sometimes is a part of the plan). The purpose of the specification is to describe the result of the project: "What will it look like when it is done?"

The schedule ties the tasks described in the plan to the results described in the specification.

Less is more.

Especially for low structure projects, small projects or projects depending upon new technology.

If the results can only be generally predicted, don't deplete the forests trying to force certainty by describing the uncertainty in detail.

The same goes for schedules. Highly structured projects deserve full schedules. For those projects with only vague expectations, describe and schedule the next one to three months in detail, the rest of the project in general. Update the schedule weekly, monthly or quarterly by rolling out the horizon (maintain the one-to-three-month horizon) as needs dictate.

Project Status Reports

Weekly, five aspects are updated:

1. Project description (any new information regarding the spec?)

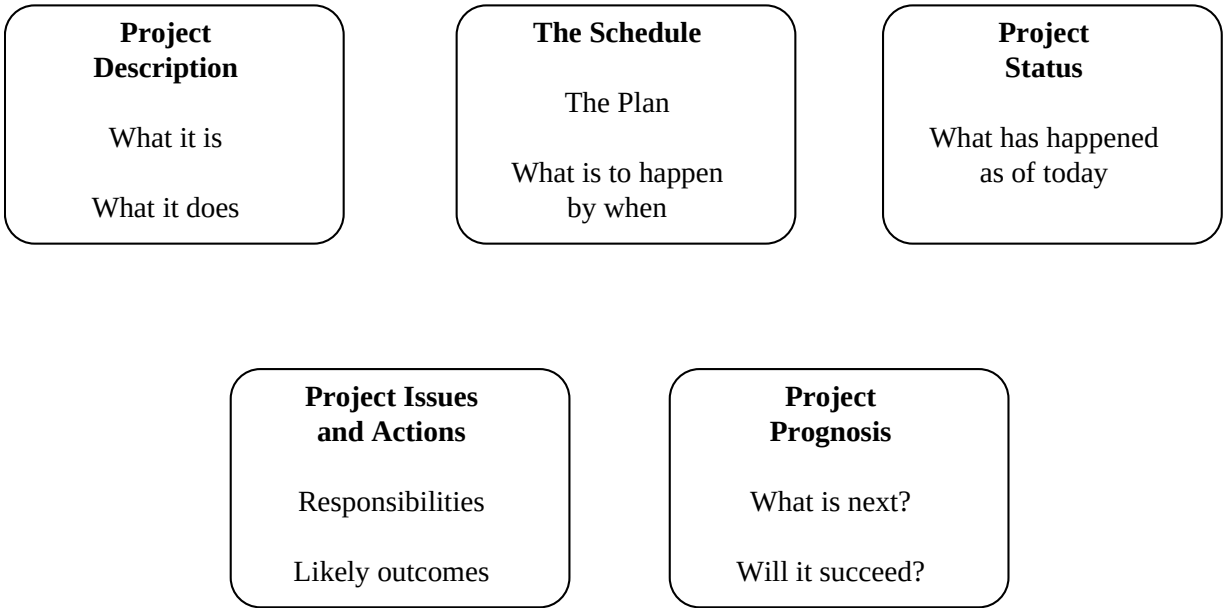


Figure 2: The five-age report

2. Project schedule (current planned sequence of events)
3. Project status (actual performance versus plan, schedule, and spec)
4. Issues, consequences, actions, responsibilities, and projected outcome
5. Prognosis (what to expect during next period)

Five topics, five pages, updated by the project team, reported to two, three or four levels above the project team with real feedback to the team keeps projects and IT management in touch and focused on real work.

Summary

Information Technology organizations do work via projects.

Projects come in all shapes and sizes. Unlike stockings, one size of project management ***doesn't*** fit all projects. The detail and the rigor used to develop the six steps in the basic project management approach, the project artifacts, and the project leadership style vary with the nature of the projects.

Real, regular reporting by the people actually doing the work to those who plan and are measured by the work of IT is key to healthy IT groups.

Glossary

Accounting: Process of capturing and displaying labor and costs expended.

Active Projects: Those projects that are currently being acted upon or are completed. Projects can not be in a simulation mode.

Active Projects Due Report: Exception report showing only those projects that are due for completion within a user specified time frame.

Actual Start Date: Date that a unit of work (Task or Project) was actually started.

Administrative Manager: Individual who has the administrative for resources (may also serve as a project manager).

Algorithm: Method of solving a problem. I.E.:

SA - Schedule to availability

ST - Schedule to target

LA - Linear to availability

LT - Linear to target

Analysis of completed programs: Exception report showing only those programs that are completed along with the percent of effort of each category of work (task groupings)

Audit Trail: Report displaying transactions and diagnostics.

Budget Cost: Forecast cost to complete a unit of work (Task or Project)

Calendar Days: Refers to all days appearing on the calendar as opposed to work days.

Completed Projects Analysis Report: Exception report showing only those projects that are completed.

Completion Date: Date that a unit of work (Task or Project) was actually completed.

Constraint: Any item that restricts or limits the scheduling of work.

Control: Process of monitoring and taking effective action to correct problems.

Controlling: See Control

Cost: Dollar expenditure to complete a unit of work

Dates:See start date, target date, etc.

Days to Work: The work days on a task or project.

Dependency: Indicates that a unit of work may not start until all predecessor units of work are complete.

Diagram: See Flowchart

Dollars estimated: See Budget Cost

Duration: Facility to specify the elapsed time in calendar days required for the completion of a task.

Estimated Hours: Forecast of the hours necessary to accomplish a unit of work (Task or Project).

Evaluating: Examination of the performance of resources, projects, managers, etc.

Executive Summary Report: High level report showing all active work.

Flowchart: Graphic diagram of dependent relationships of units of work (Tasks or Projects).

Full Project Identifier: Project identifier containing all hierarchical elements.

Hierarchical Project Identifier: An identifier that references an element of the full project.

Hours estimated: See Estimated Hours

Hours Per Day: Facility to specify the number of hours to schedule each day by task or project.

Individual Performance Record Report: Evaluation Report showing actual performance versus accepted budgets and schedules by resource.

Initial Plan: Original definition of work to be performed including task selection, estimations, dependencies, priorities, FTC.

Linear to Availability: Calculate the number of work days and schedule the resulting hours for every day until the total hours are exhausted. The availability of the resource is never to be exceeded; therefore, the target date may be exceeded. If start and target dates are not provided, schedule to availability should be used. (hours per day = hours to finish/(target date - start date).

Linear to Target: Provides the same facility as linear to availability except the hours per day calculated will be used regardless of resource availability. The work should never be scheduled late (unless the target date was already passed), but resources may be overscheduled.

LT: See Linear to Target

Lump Sum Dollars: Ability to enter direct costs expended (purchase materials, consulting services, computer costs, etc.)

Model: A mode of representation that records all observations (perceptions) made of the system under study, and predicts its behavior in various circumstances (alternatives).

Monitor: Checking work-in-process to achieve an awareness of potential and actual problems.

Network: Collection of projects interrelated by precedence notation.

Network Diagram: Flow diagram of the various projects within a network depicting their precedence relationships one to another.

Order of Execution: See Topological Order

Outside Services: Work performed outside of the organization.

Over Budget Report: Exception Report showing only those projects that are over budget.

Overtime Rate: Rate applied to labor expended on overtime.

Percent Assigned: Facility to specify resource availability by task or project (1-200% of standard daily availability).

Pert Estimates: Method of estimating hours based on the formula:

$X = (A + 4M + B)/6$ where:

X = Resulting Estimate

A = Optimistic Time Estimate

M = Probable Time Estimate

B = Pessimistic Time Estimate

Plan: Process of arriving at estimates, resource assignments and schedules that best satisfies management's objectives.

Plan Date: Date submitted to specify first date to be planned in the planning reports.

Plan Flag: Facility to specify acceptance of schedules and simulation mode among other options.

Planning: See Plan

Precedence Audit Trail: Report showing those projects and tasks that have inconsistencies in precedences, priorities, start dates and target dates.

Processing Order: Sequence of the various units of work considering priorities, start dates, target dates, topological order and other such user defined constraints that determine the order in which scheduling is to be performed.

Program: See Project

Project: Collection of tasks to be performed which is assigned a project identifier and description.

Project Description: Description to define a project.

Project Diagram: Flow diagram of the various tasks within a project depicting their performance relationships one to another.

Project Identifier: An identifier used to reference a project.

Project Leader: See Project Manager

Project Manager: Individual who has the responsibility of planning and control of a project.

Project Number: See Project Identifier

Project Precedence: See Project Predecessors

Project Predecessors: Those projects that must be complete prior to the start of a project.

Project Priority: Priority specified at the project level indicating the order of scheduling between projects that execute in parallel.

Pseudo-Resource: Resources that are not uniquely identifiable as a person or machine (programmer, payroll analyst, engineer, etc.) where actual assignment is postponed.

Resource: Performer of work (person, group of people, machine or work station).

Resource Audit Trail: Report showing all resource definition transactions with diagnostics when appropriate.

Resource Contention: Competition of various projects and tasks for the services of resources.

Resource Description: Description to define a resource.

Resource File: Library of resource identifiers, descriptions, costing rates, scheduling availability, skills and managers.

Resource Identifier: An identifier used to reference a resource.

Resource Number: See Resource Identifier

SA: See Schedule to Availability

Schedule: Forecast start and target dates.

Schedule to Availability: Indicates that the work is to be scheduled to the full availability of the resource, but never to exceed that availability (the target date may be exceeded).

Schedule to Target: Indicated that the work is to be scheduled to the full availability of the resource. However, never schedule any work beyond the target date (the resource may be overscheduled).

Simulation Mode: Software Computer program(s) used to accomplish a data processing objective.

Software Engineering: A set of disciplines and procedures facilitating the development of software that satisfies requirements in a way that is easy to maintain.

ST: See Schedule to Target

Standard Rate: Rate applied to labor expended in the standard work day (contrasts with overtime rate).

Start Date: Date that a unit of work (Task or Project) actually started or may begin in the future.

Structured Projects: Projects that are interdependent.

Structured Tasks: Tasks that are interdependent.

System: Any configuration organized by activities placed in a network, that is understood to be coherent by an observer, and that has a common objective.

Target Date: Forecast completion date of a unit of work (task or project).

Task: Smallest measurable unit of work that is used repetitively in various projects.

Task Audit Trail: Report showing all task definition transactions with diagnostics when appropriate.

Task Description: Description entered by the user to define a task.

Task File: Library of task identifiers, descriptions and default task dependencies.

Task Identifier: An identifier used to reference a task.

Task Number: See Task Identifier

Task Precedence: See Task Predecessors

Task Predecessors: Those tasks that must be complete prior to the start of a task.

Task Priority: Priority specified on tasks indicating the order of scheduling between tasks that execute in parallel.

Topological Order: A sequence that is consistent with the dependencies of the units of work.

Turnaround Time Document: Generated time document for each resource to record time expended, task completion, comments, etc.

Turnaround Planning Document: Exception turnaround document showing only those projects and tasks with exceptional conditions (late, overscheduled, early, simulation mode). This document also provides facilities to input data to affect the scheduling in subsequent planning cycles.

Work Days: See Days to Work