

# **The Project Management Dashboard: A Management Tool For Controlling Complex Projects**

**Project Management White Paper Series--#1001**

**John Aaron, Milestone Planning And Research, Inc.**

**5/14/01**

## **Overview**

The Project Management Dashboard (PMD) is a customized Project Management Information System containing a variety of quality based project tracking and control metrics. The PMD enables the project manager to clearly monitor the “vital signs” of a project, to identify problems early and to trigger corrective actions in a timely manner. While every project is unique and requires its own specific set of controls, certain common informational needs exist across projects. This paper describes some of the common kinds of techniques and metrics that project managers should utilize in order to monitor and control progress on complex, large-scale projects. These methods are particularly applicable to projects with aggressive schedule objectives and cost constraints.

The selection, creation and successful utilization of appropriate project metrics provide several benefits to the project manager. First, metrics focus the team on those activities that are most important. The adage “what gets measured, gets done” reflects a time tested principle of good management. Secondly, project metrics clarify and sharpen the manager’s understanding of the true status of a project which, in turn, enables earlier notification and correction of problems. All complex projects have problems periodically, and the earlier problems are identified and corrected, the less costly the problems will be. Thirdly, metrics give project team members a daily to weekly report card on key project dimensions. This creates a more routine environment for team members allowing them to spend more time on planned work and less time reacting to crises.

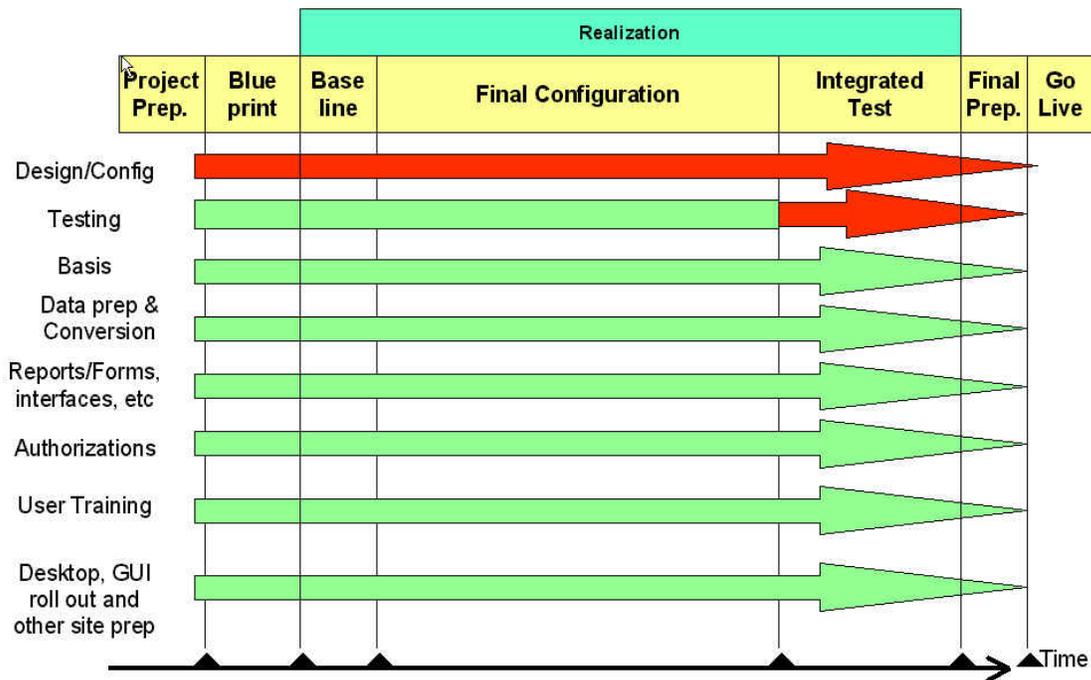
In this paper, the author demonstrates the principles of PMD within the context of managing SAP-ERP implementations. The dashboard concept is applicable and useful for managing any complex project. With only slight modification, the reader can adopt the principles directly to his/her project.

## **The Concept Of Managing Phases By Quality and Schedule**

The first step toward building a Project Management Information System is to understand the project life-cycle of the project that is to be managed. The life-cycle identifies the names of the project phases as well as the deliverables that are produced within each phase. In the case of SAP, the commonly used methodology called ASAP includes a project life-cycle component. The ASAP life-cycle mirrors a “waterfall” development approach that includes distinct phases for requirements definition, design, build, test and release to a production system. In ASAP terminology the life cycle includes distinct phases for project preparation, blueprint, realization, final preparation, go live and support.

The diagram shown below helps one visualize the management challenge in ASAP terms. The phases of the ASAP implementation life-cycle are shown across the top. The various tracks of work that cut through each phase are shown along the side. The project management challenge is to ensure that all of the key deliverables within each track for a particular phase meets both schedule, cost and quality objectives.

## ASAP Phases And Tracks Of Work



*Red denotes the usual critical path*

Three important considerations exist when project managing under the ASAP life-cycle. These are: 1) the completion of a phase constitutes a high level milestone for the implementation, and a phase should not be considered complete unless the work products for the phase have been sufficiently completed, 2) phases should be given challenging but achievable schedule objectives, adequate time needs to be budgeted to complete a phase for a given scope of work and set of resources, and 3) the quality of critical work products within a phase should be reviewed and verified before a phase is considered complete.

The essence of the above three principles is simply this: the schedule and budget goals of the project will ultimately be put at risk when project teams do not complete the *critical* work sufficiently in a phase and push the incomplete work into later phases.

Several reasons exist that explain why incomplete work will cause problems when it is allowed to pass to a subsequent phase. First, when incomplete work is pushed forward, it generates costly rework. Incomplete work and open issues are very much like defects on a production line that get increasingly expensive to correct. Secondly, the pushing of work into a later phase will force the team members to try and accomplish too many things at once during that later phase. This normally creates chaos on the project and drastically reduces the productivity of team members causing delays.

This sounds a bit obvious, but in practice the complexity of a large-scale project combined with the schedule pressure can impede the practice of this common sense principle. In reality the project manager is often faced with situations in which a phase exit milestone is approaching yet work is not nearly complete. In such situations a dilemma arises for the project manager because it is not usually economical to stop projects at review points to give time for those who are behind

to catch up. Rather, a good project manager will try to raise the visibility of the incomplete work early enough so that remedies can be put into effect that preserve schedule adherence of the project.

When early detection of incomplete work is not possible, the project manager has less chance for recovery. As a project approaches a phase exit review the project manager would not necessarily *stop* a project to wait for those who are behind to catch up, but he/she must *encumber* the project by making issues and unfinished work highly visible. This has the desired effect of creating a crisis and prevents those people who are behind from moving forward until their work is complete. The high visibility of issues and unfinished work serves as a forcing function that enables team members to focus on quick resolutions.

Clearly, project managers can be trained to effectively deal with a phase exit crisis as described above, but the better option is to avoid the crisis in the first place. A more effective approach is to provide project managers with an early warning mechanism that detects when work is falling behind across all tracks of work and that triggers early corrective actions. To achieve early warnings, however, the project manager must have an effective project management information system.

### **The Inadequacy Of Task Lists and Gantt Charts on Complex Projects**

Generally, the use of traditional project planning and control tools such as Task Lists and Gantt Charts is not adequate to control a large-scale project. All projects need project plans, and project planning tools (like MS Project) are quite useful for generating schedules especially at the sub-team level. However, these kinds of tools alone are usually inadequate for tracking and controlling complex and integrated projects.

Task lists and Gantt charts are necessary, but the sole reliance on a tool such as MS Project makes for a very weak project information system that can fail to give the project manager adequate warning when deliverables are falling behind plan. The statusing of task lists with “percentage complete” information and actual start/finish dates provides utility as a checklist. However, without data consolidation and trend analysis the tool fails to provide unambiguous warnings when things are going wrong. The sheer number of line items on a project plan can obfuscate trends and mask problems. The sole reliance on task lists and Gantt Charts can lead to the condition whereby the project manager finds that he/she is approaching a major milestone (i.e. review point) and behind schedule with little hope of recovery.

A good way to avoid this kind of dilemma is to support project plans with early warning indicators consisting of critical path analysis and negative float calculations along with metrics that are grounded in trend analysis and phase exit criteria. Such a project management information system provides the project manager with unambiguous warnings across all tracks of work on a project. It operates very much like the warning lights on an automobile dashboard--thus the name *project management dashboard*.

## **Building A Project Management Information System**

### **A) Critical Path Analysis and Negative Total Float**

Critical path analysis has been in use since the 1950s. It requires the project manager to assign duration estimates to all activities of a project plan and then to link the activities together with predecessor-successor relationships to form a network. Then, a forward and backward pass is performed to generate a calculation of total float for each activity  $A_{ij}$  so that:  $TF_{ij} = LFT_{ij} - Dur_{ij} - ES_{ij}$ . In other words, the Total Float of an activity equals the Late Finish Time of an activity, minus the duration of the activity, minus the Early Start Time of an activity.

The utility of this approach arises from its data reduction power as well as from its unambiguous indication of warnings. As activities on the plan fall behind, negative total float values are calculated. The activities containing the most negative float form what is known as the *critical path*.

Example 1 below illustrates a basic task list that is typical of a project plan. This example shows a suggested layout of the project plan and includes the critical metric called Total Float. The necessary information that should be displayed on a project plan is as follows:

- The activity ID number and name
- Remaining duration in days
- Percent complete
- Baseline start
- Baseline finish
- Actual start
- Actual Finish
- Total Float

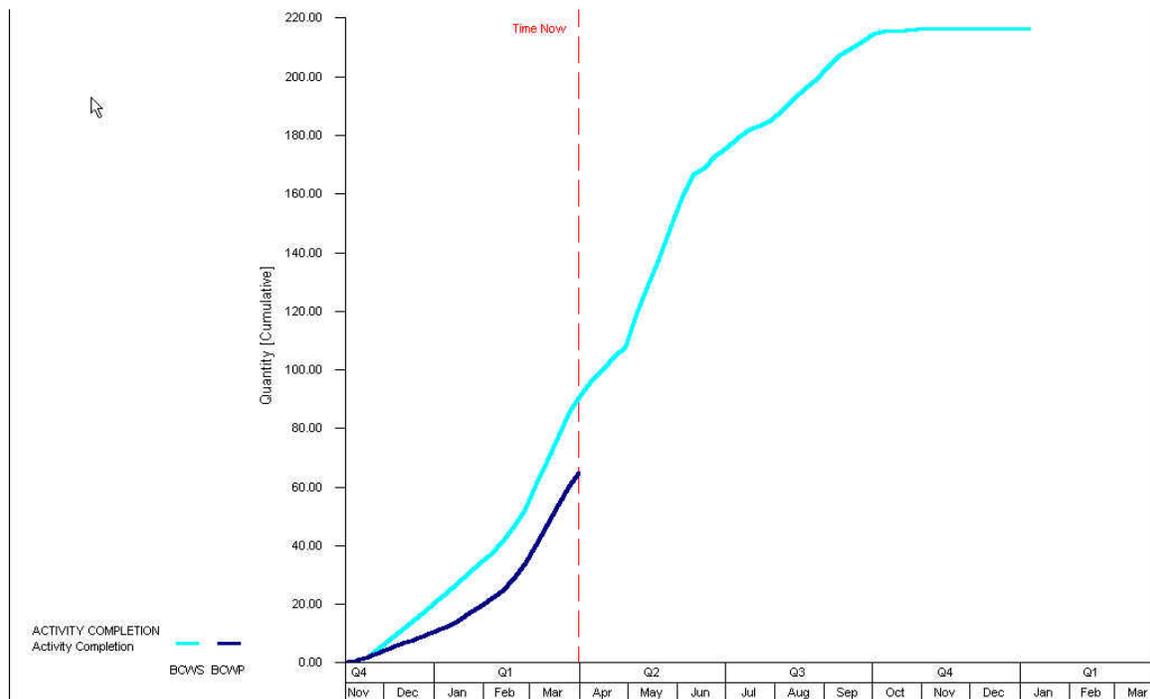
Baseline start and finish dates refer to the planned dates. These dates are then compared to actual dates to identify variances. The project manager must maintain the plan on at least a weekly basis by updating remaining duration, percent complete, actual start and actual finish. From this information total float is calculated. Negative total float on an activity shows the days behind that the activity is running in relation to the targeted end date of the project. Activities that are connected showing the greatest amount of negative float form the “critical path”. Project managers normally manage by exception. So, those activities with large float variances are noted for follow-up and corrective action.

### Example 1

Activity ID	Activity Desc.	Remaining Dur.	Progress Percent	Baseline Start	Baseline Finish	Actual Start	Actual Finish	Total Float
0.10	Reports and Forms	45d	77	28Feb00	28Aug00	03Jan00		-9d
0.11	Integration Testing	86d	33	28Feb00	01Oct00	15Apr00		-7d
0.11.1	Test Cases	5d	96	28Feb00	01Jun00	15Apr00		-7d
0.11.1.01	Review test cases	4d	87	28Feb00	06Apr00	15Apr00		-6d
0.11.1.02	Create and prepare test cases	5d	83	07Apr00	18May00	15Apr00		-7d
0.11.1.03	Master data prep for testing	5d	50	19May00	01Jun00	15Apr00		-7d
0.11.2	Iteration 1	24d	77	01Jun00	05Jul00	02Jun00		-3d
0.11.2.1	Build QA Client	0	100	01Jun00	04Jun00	02Jun00	05Jun00	0
0.11.2.2	Load Data and prep master data	0	100	05Jun00	08Jun00	05Jun00	07Jun00	0
0.11.2.3	Iteration 1 Testing	15d	0	09Jun00	29Jun00			-7d
0.11.2.4	Validation	4d	0	30Jun00	05Jul00			-6d
0.11.3	Iteration 2	25d	0	06Jul00	04Aug00			-6d
0.11.3.1	Build QA Client	3d	0	06Jul00	10Jul00			-6d
0.11.3.2	Load Data and prep master data	3d	0	11Jul00	13Jul00			-6d
0.11.3.3	Iteration 2 Testing	12d	0	14Jul00	31Jul00			-9d
0.11.3.4	Validation	4d	0	01Aug00	04Aug00			-9d
0.11.4	Iteration 3	23d	0	07Aug00	05Sep00			-9d
0.11.4.1	Build QA Client	3d	0	07Aug00	09Aug00			-9d
0.11.4.2	Load Data and prep master data	5d	0	10Aug00	16Aug00			-9d
0.11.4.3	Iteration 3 Testing	10d	0	17Aug00	30Aug00			-9d
0.11.4.4	Validation	4d	0	31Aug00	05Sep00			-9d
0.11.5	Iteration 4	86d	0	01Sep00	01Oct00			55d
0.11.5.1	Build QAS Client	3d	0	01Sep00	05Sep00			55d
0.11.5.2	Load Data	4d	0	06Sep00	11Sep00			18d
0.11.5.3	Iteration 4 Testing	16d	0	12Sep00	25Sep00			-9d
0.11.5.4	Validation	5d	0	26Sep00	01Oct00			-9d

Example 2 is an extension of the plan shown in Example 1 above. Here the project manager creates a classic S-Curve that graphically illustrates the performance of a project--actual activity completion rates versus planned activity rates. In this case the measure is the number of activities on the project plan that have been completed, and the scope of the project (or sub-project) is about 200 activities. The graph augments the float information shown in Example 1 because it highlights all activities not just those with negative float. In this example it suggests that the actual performance has been running behind planned performance and that corrective action is underway. The project scheduling tool called *Open Plan* from Welcom Software is the tool of choice to generate this metric. A similar graph could be created from MS Project and Excel.

**Example 2**



**B) Adding Quality And Schedule Metrics To The Information System**

In 1993 the author delivered a paper with two colleagues from Tellabs, Inc. at the Annual Symposium of The Project Management Institute<sup>1</sup>. The paper described the Quality Gate approach to managing large-scale software projects. The concept remains highly relevant for managing large-scale, complex projects such as today’s ERP implementations.

By definition a quality gate is a “collection of completion criteria and sufficiency standards representing the satisfactory execution of a phase of a project plan”. To construct a quality gate on a project the project manager must first identify the essential deliverables for each phase of the project. Then metrics must be constructed for each criterion that point to the level of completion. The quality gate concept serves as the basis for a project management information system.

For example some critical completion criteria for the SAP-ERP Blueprint Phase would be:

- All “to be” process maps complete
- All questions from the Q&Adb complete
- Design review complete
- All design issues closed
- Development environment ready to begin configuration

The five criteria shown above represent only a small subset of the criteria actually needed to complete the Blueprint phase of an actual implementation, but they are useful for demonstrating the project management principles of the quality gate.

In a real situation the detailed project plan for the Blueprint phase (i.e. the MS Project schedule) may contain hundreds of activities, but the project manager can *reduce* the measurement problem down to only those items (say 20 or so) that are absolutely vital to the completion of the phase. Many of those vital 20 or so items may be complex deliverables, however, that contain hundreds to thousands of sub-deliverables. So, for these vital sub-deliverables the project manager should count completions. The counting of the sub-deliverable completions serves as a major quality assurance check, and it gives a clearer picture into the overall schedule adherence of the project as well.

To illustrate this point we return to the simple example above using the five criteria shown. We see that statusing some of the deliverables involve straightforward “yes” or “no” decisions. For example, the activities “design review complete” and “development environment ready to begin configuration” are either done or they are not. The manager can status these items on a project plan without too much guessing. On the other hand, the activities “to-be process maps complete” and “all questions from the Q&Adb complete” are complex deliverables that require further analysis. Therefore, the project manager must break down these deliverables into more detail and measure progress in more depth to gain an understanding of the true status of the deliverables.

Following this approach, the project manager would set up a measurement process for the critical deliverables of the phase that might look as follows:

<b>Deliverable</b>	<b>Responsible</b>	<b>Criterion for completion</b>	<b>Complete to date</b>	<b>% complete</b>	<b>Target Finish</b>
“All to be” process maps complete	<b>Design Teams</b>	450 maps	320 maps approved	71%	9/01/01
All questions from the Q&Adb complete	<b>Design Teams</b>	3220 questions in scope	2991 answered	93%	9/15/01
Design review complete	<b>Design Teams</b>	Review complete	Done	100%	8/20/01
Design Issues Closed	<b>Design Teams</b>	55 issues	44 issues resolved	80%	9/29/01
Dev	<b>Basis Team</b>	Environment	Done	100%	9/18/01

Environment ready		ready			
----------------------	--	-------	--	--	--

On a weekly basis the project manager would update the metrics for the current phase of the project. If it becomes apparent that actual completion rates are not progressing according to plan, the project manager raises the visibility of the variance and initiates a corrective action process. The kind of measurement approach shown above supports and complements the traditional critical path methods and S-Curve charts of activity completions that are based upon performance to the baseline project plan. Yet, it brings additional clarity and insight into the actual health of the project. The reporting approach can be broken down into even further detail to give status on the performance of each sub-team on the project as required.

One ingredient that must be added to the above table is trend analysis. Trend analysis combines the schedule dimension with the quality dimension and gives the manager an early warning indication as things fall off track. The techniques shown on the following pages are hypothetical yet realistic and combine trend analysis with phase exit criteria to form at least a partial view of an effective Project Management Information System. When used with task lists, total float and activity completion S-Curves, these metrics give the project manager adequate insight into the true performance of the project as well as early warnings. When we consolidate these measures into a single sheet, we have a working Project Management Dashboard so to speak.

Based upon the author's experience the project manager should review the performance of these kinds of critical metrics for all teams on a weekly basis. (Note: integration test metrics should be reviewed daily). In cases where actual performance does not meet planned performance, the team leaders should create remedies and take corrective action to ensure that actual performance catches up to planned performance.

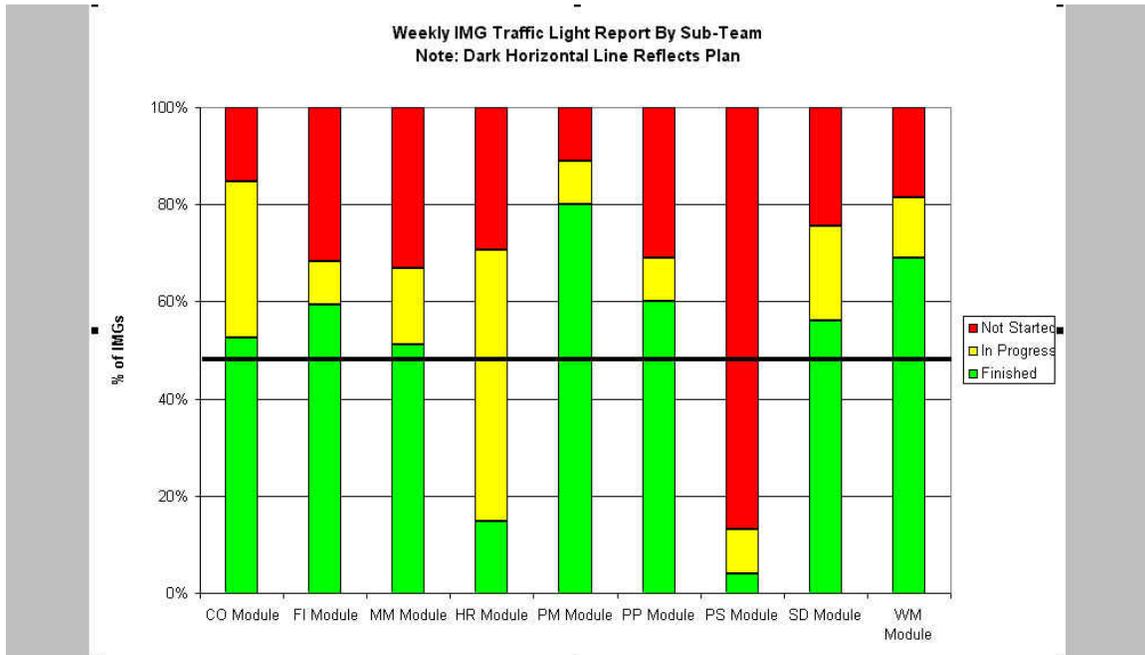
In many cases team leaders will run into tough issues that are beyond their ability to resolve alone. These issues should be documented in an issue log that is reviewed publicly each week. The project manager and management sponsors of the project must help the team leads secure timely resolution of issues.

While every project is unique and requires a tailored set of metrics The following examples are typical of the metrics required on an SAP-ERP implementation during the Realization phase. The examples shown are by no means exhaustive, but they should provide the reader with an understanding of the approach.

### **Additional Examples**

Example 3 below shows weekly tracking of progress for the customization of the SAP system during the Realization phase of the project. Here the metric is the status of IMG's completed---actual versus planned. The chart shows in traffic light fashion (i.e. green equals finished, yellow equals in-progress, and red equals not started) where each team stands with the configuration items of the system on a percentage basis. In this example each team (module) should have completed about 48% of its configuration based upon the baseline schedule. The 48% objective is denoted by the solid horizontal line. In this case the project manager should note that the HR and PS teams are behind plan. Recognition of this variance should prompt a discussion between the project manager and the HR and PS team leaders. The two team leaders should explain their proposed remedies and planned corrective actions during this discussion. Any issues that are contributing to this variance should be entered on the issue log.

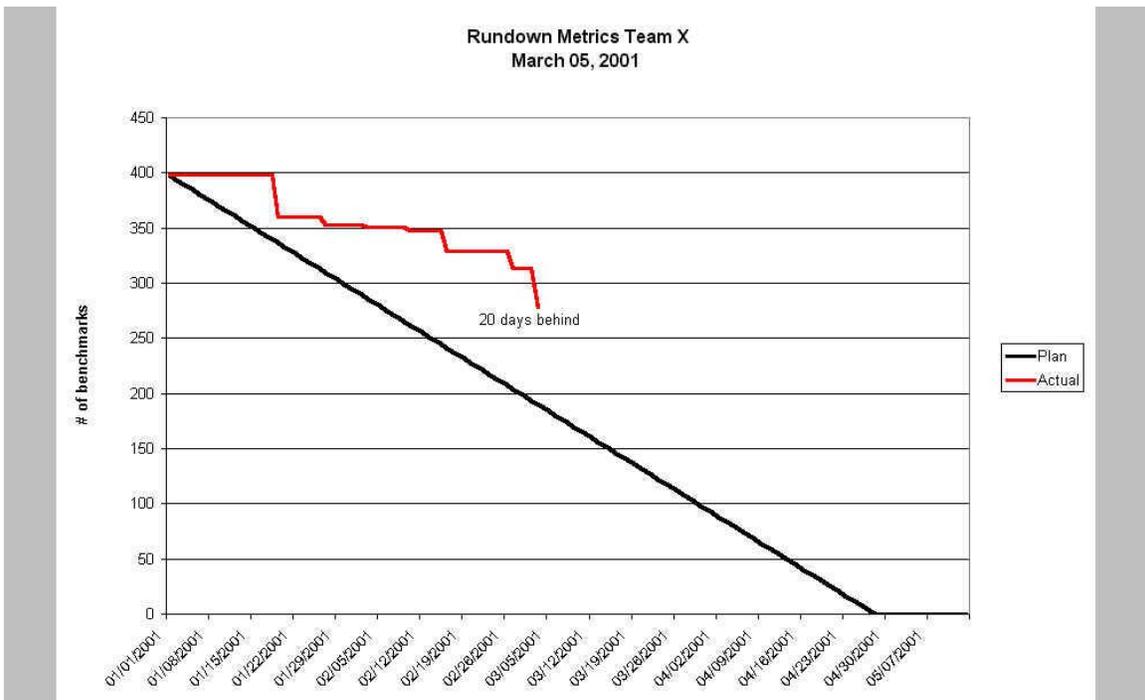
### Example 3



Example 4 below shows a metric that describes transactional readiness to perform integration testing for a particular functional team. The metric is based upon the number of SAP transactions that are in scope based upon the Business Process Master List (i.e. BPML) as generated from the ASAP Q&A database. The example illustrates a project containing a hypothetical scope of 100 SAP transactions. Following the ASAP methodology each transaction must be configured, documented, unit tested and prepared for integration testing. Thus, each transaction must meet four benchmarks which translates to 400 benchmarks total that must be completed. The metric shown below illustrates how a “linear rundown curve” can be utilized to determine progress for a particular team.

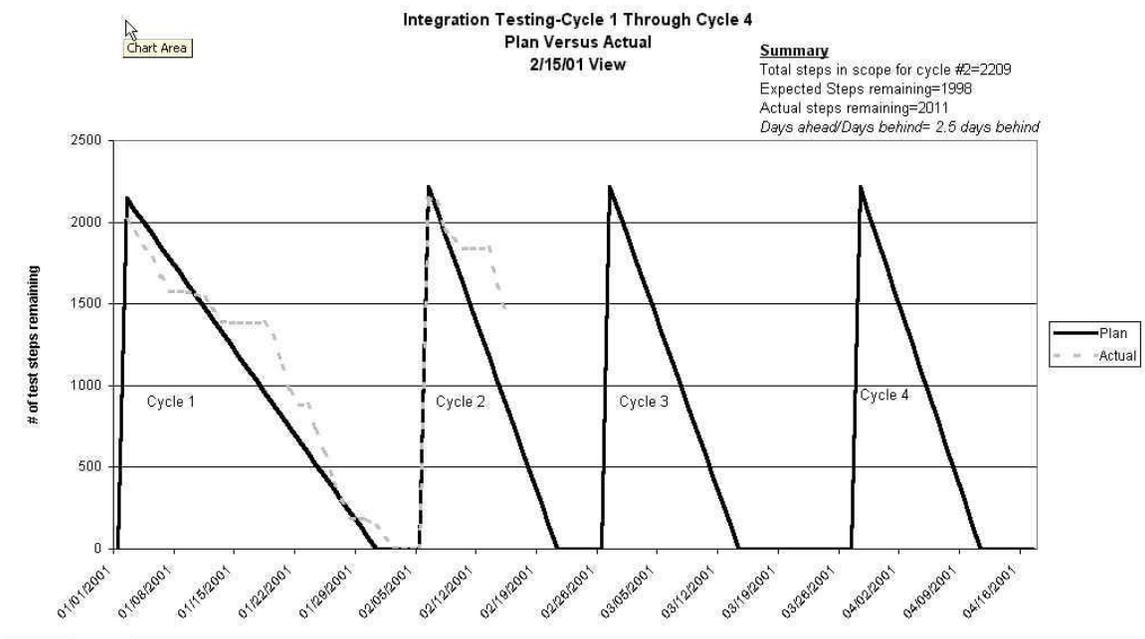
Example 4 exemplifies a common situation in which a team is behind plan. As is often the case on projects this team is behind plan due to a lack of staffing (i.e. headcount) assigned to the project. Here again the metric identifies a variance which leads to a discussion between a project manager and a team leader. The job of the project manager is to spot the variances/exceptions to plan and to raise the visibility of these exceptions to the accountable team leader. Similarly, the job of the team leader is to take responsibility for working with team members to develop remedies and take corrective action. In this case the appropriate corrective action is to assign more resources to the project and to work extra hours (or more focused hours) until the team gets back on track.

### Example 4



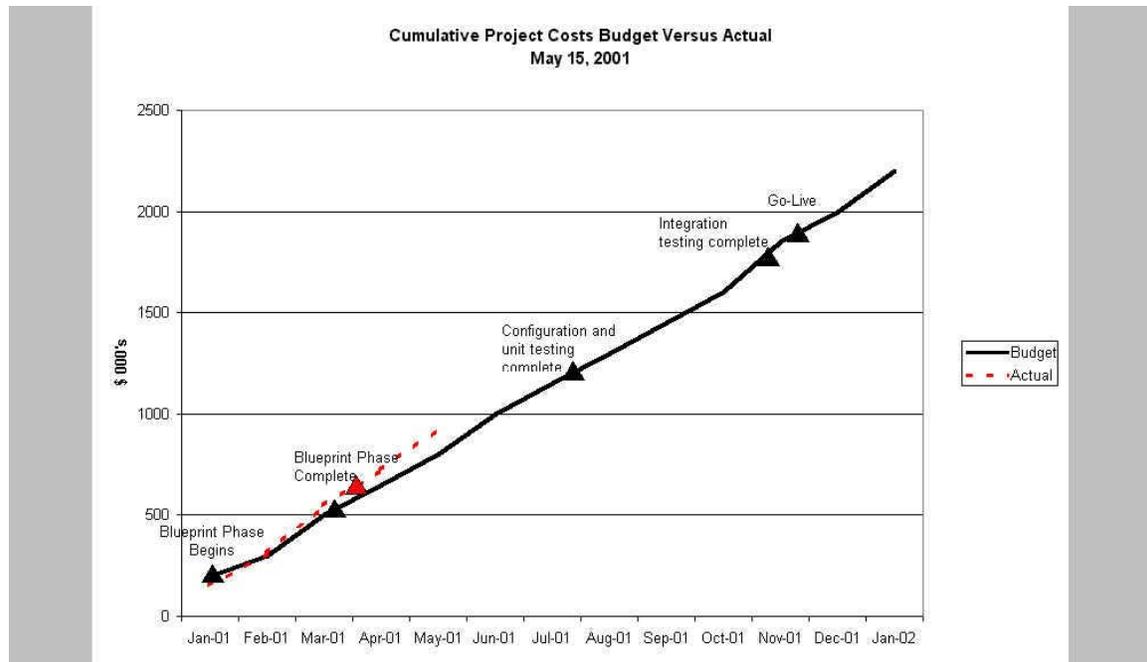
Example 5 shown below illustrates a metric for integration testing. The four dark sawtooth curves represent the planned rundown of integration test steps. The lighter line represents the actual rundown. In this case testing is 2.5 days behind. By the time the project reaches the integration testing phase, the project manager should be updating the metrics on a daily basis.

### Example 5



Example 6 shown below is a tool for project cost management. The graph shows cumulative project costs-actual versus budget. Here again is trend analysis to show variances from plan. Corrective action is triggered when variances from plan are shown.

### Example 6



### Summary

This paper discusses the major considerations related to the creation of an effective project management information system. The examples chosen relate to ERP-SAP implementations, but the principles apply to all complex projects and can be adopted quite readily. The examples shown are by no means intended to be comprehensive but to give the reader the core idea of the concepts involved.

Milestone Planning and Research, Inc. (MPR) specializes in providing corporate clients with high quality project management services, systems and training. For further information see our website at [MilestonePlanning.com](http://MilestonePlanning.com). For inquiries contact John Aaron at:

Office Phone: 708-598-5484  
 E-mail: [John@MilestonePlanning.com](mailto:John@MilestonePlanning.com)  
 FAX: 708-598-5930

### Reference

1 Aaron, J.M., Bratta, C.P., Smith, D.P. ***Achieving Total Project Quality Control Using The Quality Gate Method***, Proceedings from the 1993 Annual Symposium of The Project Management Institute.