

*Courtesy of:
Scitor Corporation
256 Gibraltar Drive • Sunnyvale, CA • 94089
800/533-9876*

Real-time Status vs. Period Data

The freshest data is not always the best data

In the beginning ...

In the earliest days of modern project management, handling project data was a batch process -- by necessity. Progress data was collected and transferred to coding sheets, which, in turn, was transferred to punched cards and submitted to the computer operator. Then we would wait for the results, review the output, correct errors, resubmit, and eventually publish the output. This process could often take up to three weeks, by which time much of the data was aged and the ability to react was impaired.

So the industry rejoiced when we matured to real-time statusing and reporting. As we near the end of this century and enter the fifth decade of automated project planning and control, we have progressed to a point where almost anyone, at any point of time, can get to any data, from anywhere. That data may be so fresh so as to have been entered in the system within minutes of retrieval. I once had a department manager describe to me the system that he was looking for. He wanted the ability to access an information database that would let him know what projects were in progress and in the queue, their status, and what everyone in his department was working on. And the data was never to be more than 24 hours old.

Today's Typical Process

Well, today, we can certainly give the gentleman what he asked for. But is that really what he wants? Let's look at a few potential scenarios and unearth a few flaws in such an approach.

First we'll start by defining a typical process:

1. The data system is structured, with common project, task and resource coding, calendars, and preferences.
2. Projects are defined; adding tasks, linking tasks, assigning resources, estimating effort and duration, calculating schedules, etc.
3. A baseline is established
4. Progress on tasks is entered.
5. Time spent on tasks is reported.
6. Actual expenses are processed.

Now, at any particular point in time, we can have partially defined projects, and helter-skelter progressing. For instance:

1. Harry, project manager for Alpha Project, enters task status on 4/15.
2. Thomas, project manager for Omega Project, is at an all day meeting with the project sponsor and can't get his task status in until 4/16.

Courtesy of:
Scitor Corporation
256 Gibraltar Drive • Sunnyvale, CA • 94089
800/533-9876

3. Most of the resources report time spent weekly, with electronic time sheets. These are due on Monday morning, 4/20.
4. Jill, however, will be out that week and enters her data on 4/16, using estimates for Thursday and Friday.
5. Jack is out sick on 4/20 so his time does not get in until 4/22.
6. Janice, in Accounting, downloads expense and invoice data from the corporate finance system and allocates expenses to projects and tasks. This is done every Thursday, based on accounting records as of the previous Friday.

Problems With This Process

If we were to take advantage of our real-time capability, the data would never be synchronized. If we used our executive browser to check on things as of the afternoon of Wednesday, 4/15, we would see task status on Alpha Project as of 4/15, Omega Project as of the previous week, actual hours as of 4/13, and actual expenses as of 4/6. In this example, reported hours and expenses on Alpha Project would be lagging the reported task progress, making performance look better than it really is. Omega Project would also be out of synch.

But this is just the top of the iceberg. On 4/14, Thomas reviews the hours charged to his project for the previous week. He questions charges entered by Mike, who was not assigned to Tom's project. He posts a query to Fran, Mike's manager. Fran, in reviewing Mike's time sheet has other questions. By 4/16, these are resolved, and they are posted to the database on 4/17. However, this means that the information viewed on 4/15 was incorrect, and has changed.

But, the potential problems can get worse. Harry had asked his project team to modify the plan for Alpha Project to reflect design changes. They are going to use a different frabistat, containing four type B gizmo assemblies, rather than two type C gizmos. Tony, the assembly team leader entered the new task plan and deleted the now obsolete tasks. Just as this was taking place (the new tasks were added - but the old were not yet deleted) Fran is checking the multi-project database to review the project loads on her resources. The system shows a severe overload during the frabistat assembly period. Fran puts in a panic call to Harry, while simultaneously looking into borrowing or outsourcing resources. Harry is perplexed by the unexpected call, not being aware of any overload problems. The problem is quickly resolved, but not before getting several people involved in dealing with a nonexistent problem.

However, while Tony is finalizing the modification to the frabistat plans, he erroneously enters 55 days for a five day task, unintentionally adding 10 weeks to this critical path work package. Just as that data was added, Charles, the Exec. V.P., viewed the project summary plan via his browser. Seeing the 10 week slip in this key project, he puts in an urgent call to Harry. "Hey", he says, "you told me that this project was on schedule and wouldn't slip". A flustered Harry doesn't know what to say. He hasn't even seen the information that his boss is reacting to. Now he has to waste valuable time putting out a

*Courtesy of:
Scitor Corporation
256 Gibraltar Drive • Sunnyvale, CA • 94089
800/533-9876*

fire and has lost some credibility with his boss. Yet, Harry hasn't done anything wrong, and, in fact, the so-called problem doesn't even exist.

Nebulous Benefits

So what do we have here as "benefits" of real-time project processing and immediate access?

1. We have the problem of synchronizing the collection of task progress data, hours charged and expenses recorded.
2. As a result of this, we diminish the validity of project performance data, often obtaining false indications of schedule and cost variance.
3. We may experience quality problems, as there is no time to analyze and evaluate the data before it is available for publication or viewing.
4. We may cause undue stress and wasted effort when such erroneous data causes other parties to react with shock and alarm.
5. We may, furthermore, precipitate unnecessary responses to problems, which will have to be reversed when the error is discovered.
6. There is a high likelihood of inconsistent information, as various people view the data at different times.
7. Thoughtful and thorough analysis and evaluation of project and resource status is difficult when the data is in an ever changing state.

A Solution

Perhaps what would be best is a combination of the old batch methods and today's real-time access. Easy, fast access for inputting data from various sources, in diverse locations, if kept under control, can be advantageous. However, there should be a structured method for processing these data before it is available for general viewing and distribution.

In the earlier days of project statusing, we had an "as-of" date. All data was normalized to this "close of data" date. This is still appropriate and essential. In our structured system, all project status is reported as of the close of data date. If Harry inputs on 4/15, and Tom on 4/16, it's OK, as long as the inputs reflect the status as of the data date (let's use 4/10). Time entry must also be as of 4/10, as well as imported expense data.

This might not satisfy that department manager, who wanted the data to never be more than 24 hours old. But let's face it. Would you rather have current (but potentially flawed) data, or good data? You are better off, in most cases, to have good data that is a week old, than to have fresh data that lacks accuracy and consistency, and is therefore unreliable.

Next, in our structured system, there should be a series of quality checks, before the data is published. I have a series of exception queries that I routinely make that helps me to ferret out any unusual data. For instance, if I am doing a biweekly update, I list all of the

*Courtesy of:
Scitor Corporation
256 Gibraltar Drive • Sunnyvale, CA • 94089
800/533-9876*

tasks that have slipped more than two weeks since the last update. (I can do this because I capture a temporary baseline of my last published set before further changes are made.) Then, I can run a comparison of the 4/10 data to the 3/27 data and list any items with a spread of over two weeks. Such occurrences may be legitimate, but some may be the result of erroneous inputs.

Now that we have a reliable set of data, what shall we do with it? I, for one, feel that data, by it self, can almost be worse than no data at all. The data should tell a story. The publisher of the data may be fully aware of its meaning. But most of the target audience will need guidance. If there are variances, where are they and what do they mean. What is the impact and what are the recommended corrective actions?

We need to freeze the data at some point in time and stop to analyze it. It is not good enough to just pass the data around, or publish it to a website. The data should be used to generate responses to move the projects ever forward toward their objectives. Reports need to be published for project managers, resource managers, CFO's, other executives, sponsors, clients, etc. Stories need to be created and told. Meetings and communication need to be initiated around the published data.

We need to be proactive in this endeavor. We can't wait for people to react to raw (sometimes inaccurate) data. If we do that we will end up wasting our time responding to irritating and nonproductive calls, rather than guiding the organization toward project success.

If we do not freeze the data periodically, and pause to analyze and publish clear and informative information, the entire system will drown in data and chaos.

Be Careful What You Ask For

There is an old saying: "Be careful what you ask for. You may get it." Recently, many people have been asking for real-time access to project status data. And, today, we can give them what they are asking for. But do they really want it? Do they really want their boss to see the data before they do? Do they really want the client to see the problems before they have had a chance to develop a corrective action plan? Do they want to have all of the data out on the street before it has been checked for quality and impact? I would think not.

The answer is to capture and freeze the data periodically, so that it can be checked, analyzed, and reported, complete with informative stories and action plans.

*Courtesy of:
Scitor Corporation
256 Gibraltar Drive • Sunnyvale, CA • 94089
800/533-9876*

Harvey A. Levine, with 38 years of service to the project management industry, is founder of **The Project Knowledge Group**, a consulting firm specializing in PM training, PM software selection, evaluation & implementation, and PM using microcomputers.

He has implemented or enhanced the project management capabilities of numerous firms, often combined with the selection or implementation of computerized project management tools. Mr. Levine is considered the leading consultant to the project management software industry and is recognized as the leading expert in tools for project management.

He has been an Adjunct Professor of Project Management at Rensselaer Polytechnic Institute and Boston University. And has conducted numerous project management public seminars for ASCE, AMA, IBM, and PMI.

Mr. Levine is the author of the book "Project Management using Microcomputers", and has been published extensively in other books, periodicals and videos.

Mr. Levine is a past president of the Project Management Institute and the recipient of *PMI's 1989 Distinguished Contribution to Project Management* award. Recently, he was recently elected as a *Fellow of PMI*.

Mr. Levine has offices in Saratoga Springs, NY and San Diego, CA and can be contacted via e-mail at: LevineHarv@cs.com