

## 2.5 Agent-Based Simulations for Project Management

### Agent-Based Simulations for Project Management

J. Chris White  
ViaSim Solutions  
Rockwall, TX

[jcwhite@viasimsolutions.com](mailto:jcwhite@viasimsolutions.com)

Robert M. Sholtes  
SimBLOX, LLC  
Rockville, MD

[rsholtes@simblox.com](mailto:rsholtes@simblox.com)

**Abstract:** Currently, the most common approach used in project planning tools is the Critical Path Method (CPM). While this method was a great improvement over the basic Gantt chart technique being used at the time, it now suffers from three primary flaws: (1) task duration is an input, (2) productivity impacts are not considered, and (3) management corrective actions are not included. Today, computers have exceptional computational power to handle complex simulations of task execution and project management activities (e.g., dynamically changing the number of resources assigned to a task when it is behind schedule). Through research under a Department of Defense contract, the author and the ViaSim team have developed a project simulation tool that enables more realistic cost and schedule estimates by using a resource-based model that literally turns the current duration-based CPM approach "on its head." The approach represents a fundamental paradigm shift in estimating projects, managing schedules, and reducing risk through innovative predictive techniques.

#### 1.0 BACKGROUND

For both commercial and government organizations, the ability to manage projects effectively is a major contributor to an organization's overall performance. If an organization cannot manage its internal projects effectively, resources, time, and money are wasted. For commercial organizations, this weakens the organization's market position and capacity to generate business, which can ultimately result in closing down entire business units within the organization or the entire organization. For government and military organizations, this reduces the effectiveness of the organization's operations and system development efforts, which can ultimately result in the loss of human lives. Therefore, effective project management (PM) is necessary for strong organizational performance. This is not news to anyone.

However, times are changing for the project manager. Welcome to the 21st century! Projects are more complex than ever, tough economic conditions put enormous pressure on achieving success, and the typical project manager has several concurrent projects and no longer has the luxury of focusing on only one project. In addition,

today's computers are capable of handling more computationally intensive analyses, so why not use this power with an agent-based simulation approach to project management?

This paper will review the current approaches used in today's project management tools and introduce a new, agent-based simulation approach that offers a higher level of realism and, consequently, a higher level of project success.

#### 2.0 CURRENT PROJECT MANAGEMENT (PM) METHODS

Almost all project planning and scheduling tools on the market today use some type of PERT and/or CPM methodology as their primary underlying methodology. PERT (Program Evaluation and Review Technique) was invented by the U.S. Navy in the 1950's to manage the Polaris submarine missile program. CPM (Critical Path Method) was invented about the same time in the private sector. These two approaches are synonymous and are often interchanged or even collectively called PERT/CPM.

The PERT/CPM approach (Figure 1) is considered superior to the previously preferred Gantt approach, a horizontal bar

chart developed as a production control tool in 1917 by Henry L. Gantt. A major problem with the Gantt method was that it did not indicate task dependencies so a PM could not tell how one task falling behind schedule affects other tasks in the project. The PERT/CPM method is designed to do this. (It should be noted that today's Gantt charts actually leverage some of the PERT/CPM information by now showing dependency connections on the horizontal bar charts.)

PERT charts depict task, duration, and dependency information. Each chart starts with an initiation node from which the first task (or tasks) originates. If multiple tasks begin at the same time, they are all started from the node or branch, or fork out from the starting point. Each task is represented by a line which states its name or other identifier, its duration, the number of people assigned to it, and in some cases the initials of the personnel assigned. The other end of the task line is terminated by another node which identifies the start of another task, or the beginning of any slack time or float time (i.e., waiting time between tasks).

Each task is connected to its successor tasks in this manner forming a network of nodes and connecting lines. The chart is complete when all final tasks come together at the completion node. The key difference with the CPM method is that the CPM method highlights the critical path, the sequence of activities for which there is no (or the least amount of) slack or float time. Thus, by definition, the critical path is the pathway of tasks on the network diagram that has no extra time available (or very little extra time). Note that it is possible to have multiple critical paths.

The latest innovation in the PM world is the Critical Chain method developed by Eliyahu M. Goldratt in 1997. The critical "chain" is akin to the critical "path" of CPM, but has a slightly different definition: the path of dependent tasks that define the expected lower limit of a project's possible completion time. The Critical Chain method leans

heavily on Parkinson's Law, which suggests that work will expand to fit the time allowed for it. It is believed that if you "hide" the extra time, people on the project won't know it's there and will work to meet the shorter task times. "Buffer" activities are then inserted into the project plan as buckets of extra time (i.e., collections of slack or extra time from the other activities). The purpose of the buffers is to protect the promised completion date from variation in the critical chain. In essence, the PM makes participants think it's a very short project, all the while keeping some extra time set aside in case any task runs late. It's more a psychological game than a true management method. Thus, it is not much of an innovation. The underlying approach, PERT/CPM, is still the same fundamental approach. To be sure, the Critical Chain method helps a little, but the method is still inhibited by insufficiencies inherent within the underlying PERT/CPM methodology.

### **3.0 LIMITATIONS OF CURRENT PM METHODOLOGIES**

To date, improvements to PM tools have been evolutionary and not revolutionary. All the PM tools are still based on the PERT/CPM approach. The improvements that occur with each round of new PM tools are merely "bells and whistles" (e.g., better tracking of resources, uploading from spreadsheets) or perhaps enabling the tool to be used over the internet. Unfortunately, a 50-year old methodology delivered over the internet is still a 50-year old methodology.

Fundamentally, the PERT/CPM approach suffers from three (3) major flaws:

1. Task duration is an input. However, many factors (e.g., availability and productivity of resources, dependencies among tasks, hours worked by employees) affect the duration of a task. Thus, in the real world, task duration is actually an output.
2. Productivity impacts are not considered. In current PM tools, labor can

be added to or removed from a task with no impact on the productivity of labor applied to the task. Today's tools assume all resources are equal. Yet, we know they are not. New employees or junior-level employees do not get as much work done in the same period of time as experienced, senior-level employees. Also, in current PM tools people can be scheduled for overtime with no impact on their productivity. However, anyone who has worked a significant amount of overtime can validate that productivity decreases due to fatigue or burnout. Working a little overtime on a couple of days usually has a negligible impact, but long durations of working overtime can have significant impacts on labor productivity. Lastly, it is a known fact (especially on software development projects) that throwing more people at a task often makes the task fall further behind schedule due to lower labor productivity as experienced people train the new people and the new people make mistakes that must be corrected.

3. Corrective actions are not captured. The actual management decisions and actions that PM's take during a project are not included. However, these corrective actions can significantly influence progress. Current tools only match resources against task assignments. As a result, current PM tools allow for static planning, but not dynamic reaction and re-planning. In current PM tools, if it looks like a task will run late (e.g., based on the Earned Value schedule performance index, SPI), the project manager must develop several different plans through trial-and-error to see if they will work. The current tools do not help the project manager actually manage the project. The tools only allow the PM to develop multiple, static plans with no insight.

These inherent flaws indicate that our current PM tool set is too simplistic and does not reflect reality. As a result, these tools cause us to make decisions that are detrimental to project success. In fact, it is not uncommon to doom ourselves to failure

(or at least a very long and difficult road) with our first baseline project plan. In other words, right out of the gate we are already off course! This is not a criticism of project managers; it is a criticism of the simplistic approaches found in today's PM tools that give us insufficient and sometimes even incorrect answers. Consequently, we often rely heavily on individual PM's to single-handedly make projects successful. We applaud heroic efforts where PM's work around the "system" to make everything work out right. Why not have a "system" that actually helps the PM succeed?

Essentially, the current PM tools are not capable of handling the complexity of the issues experienced on most projects because they are rooted in a simplistic approach that was developed 50 years ago. When it was developed, the PERT/CPM approach was innovative and useful. Unfortunately, its effectiveness and appropriateness have significantly eroded over time. If we want a better tool for PM's, we need a better approach than PERT/CPM.

#### **4.0 OVERVIEW OF THE DYNAMIC PROGRESS METHOD (DPM)**

The Dynamic Progress Method (DPM) is a new approach to planning, estimating, and managing projects that builds upon the power now found in computers and applies a different type of simulation model than is currently used in most tools. The underlying simulation model in most of today's project management (PM) tools is very simplistic because it was developed at a time when computers had very little computational power. Thus, the model had to be simple enough to use a pencil-and-paper approach as a back-up. Now, computers have much greater power... so let's use it. Furthermore, we've learned a lot more about managing projects, so let's put that knowledge to use, too.

The goal of DPM is to address the three major flaws of the current CPM approach. DPM begins with a fundamental re-

evaluation of how input information is used. Figure 2 provides a good example of this. After defining which tasks are to be done and their dependencies, a project manager (PM) then begins the process of estimating for each task. For a task, the PM usually knows about how much work needs to be done for the task, who might do that work, an idea as to how “good” or “effective” that person(s) is, and an idea about the availability of that person. In the example in Figure 2, it’s easy to see that an 80-hour task with one assigned resource that is available 8 hours/day and is 100% productive (i.e., for each hour the resource is paid, the resource completes an hour of actual project work) gives a task duration of 10 days  $[(80 \text{ hours}) / (8 \text{ hours/day}) = 10 \text{ days}]$ . With CPM-based planning tools, the duration is the input. Conversely, with DPM-based tools, the individual task and resource inputs are used instead. In this example, if the resources are actually available at the level of productivity assumed, then the duration will be the same 10 days.

In the previous section, we discussed how having task duration as an input is a flaw in today’s tools. Let’s use this example to highlight this fact. In this example, notice that the duration input for CPM does not need any of the detailed task/resource information. As a result, a project manager can input a duration without knowing the details. Any duration is just as good as any other duration. Some people may think that this is a good thing because it simplifies the data input process by not burdening the PM with figuring out all those details. However, that fact that an entire plan can be constructed without the resource details is a dangerous endeavor. How can these estimates be justified and verified? They can’t! Yet, a great deal of trust is placed on these plans that are tenuous at best. DPM requires the PM to be explicit about the estimates. If a duration is assumed, why? Surely there must be some information that the PM is basing the estimate on? DPM pushes the project manager to use that

information. It’s another level of detail. It seems at times that we are not willing to add this level of detail at the beginning of a project, yet during a project we are completely fine with re-working and re-planning a project multiple times at great expense (both time and money) as we realize that our estimates were insufficient (because they lacked detail).

What is interesting to note is that today’s tools back-calculate some of these details. For example, if you input a duration of 10 days for a task and then assign a resource that has a 8-hour work day, the tool will back-calculate that the task is an 80-hour task. To test this, try adding another resource. With 2 people, the tool will automatically cut the duration down to 5 days  $[(80 \text{ hours}) / (16 \text{ hours/day}) = 5 \text{ days}]$ . That is because the completion rate is doubled from 8 hours/day (1 person \* 8 hours/day) to 16 hours/day (2 people \* 8 hours/day). Thus, this information is actually used by the current tools, though it is not evident to the user. Or, change the work day for the single resource to 16 hours/day. You will get the same 5 day duration.

The underlying model for DPM is an operational model, which means that it mimics the actual actions and processes of a project. Figure 3 provides a schematic diagram of the task execution portion of the DPM model. (Note that everything in Figure 3 is actually contained in today’s PM tools, just perhaps used a little differently.) DPM begins with a “bucket” of work that will be done for a task (Work To Do). As resources are allocated to the task, work is done (Completion Rate) and work moves from “to do” to “complete” (Completed Work). The rate at which work is completed is based on how many resources are allocated (Number of Resources) and how many hours per day those resources work (Actual Labor Hours). Then,

“Effective” Labor Hours = (Number of Resources) \* (Actual Labor Hours)

Ex: "Effective" Labor Hours = (1 person) \*  
(8 hours/person) = 8 hours

By changing the number of resources allocated to a task and/or the number of hours the resource is available to work on the task, the rate of work completion can be changed.

The graphics in the lower right of Figure 3 indicate that completion of work for a given task may also depend on the progression of work on other preceding tasks. For instance, Task 2 may require that Task 1 is 100% complete before Task 2 can start (i.e., Finish-to-Start dependency).

The term EV/Status Calculations at the top of Figure 3 indicates that at any point in time the status of the task relative to its expected schedule and cost can be determined. This shows whether a task is ahead/behind on schedule and over/under on cost. A growing trend is to use Earned Value metrics (e.g., Schedule Performance Index - SPI, Cost Performance Index - CPI) to represent "schedule pressure" and "cost pressure" on the task. (For a full discussion of Earned Value metrics, please see other sources.)

Where DPM adds value is shown in Figure 4 in green and red. One of the first new elements that DPM incorporates is Resource Productivity (i.e., the effectiveness or efficiency of a particular resource to get work done). Productivity is a measure of how much actual project work gets done for every hour of time the resource is paid. For example, as stated previously, new employees or junior-level employees do not get as much work done in the same period of time as experienced, senior-level employees. As another example, sometimes there is a "learning curve" associated with a project. A person may start the project with a low level of productivity because it is a new team and new work, but over time the person gains proficiency and grows to a higher level of

productivity. DPM adds this productivity component.

Next, in the real world, status information (e.g., EV/Status Calculations) may drive a project manager to apply some sort of corrective action or management decision to get a task back on course. In the specific DPM model described here, there are two actions that a PM can take: (1) Add/remove resources, or (2) Add/remove work hours. For example, when a task is falling behind schedule, a PM may decide to allocate additional resources to the task, or the PM may decide to work everyone overtime, or the PM may decide to do a combination of both. However, in the real world there are consequences for these actions. There is not always a 1-for-1 return. For instance, as stated previously, anyone who has worked a significant amount of overtime can validate that productivity decreases due to fatigue or burnout. DPM incorporates this impact and it is shown by the red line with Fatigue.

Similarly, for a task there may be an "optimum" number of people to work on that task to maximize productivity. Having too few or too many people may make the resources less productive. In some cases, there may actually be physical limits that prevent throwing a lot of people at the job (e.g., limited space in a crawl space for electricians). Or, even though there may not be an optimum number of people to maximize productivity, it is common to experience productivity losses when more and more people are allocated to a job because of difficulties with communication and coordination among all the people. DPM incorporates this impact, also, and it is shown by the red line with Over-manning.

## 5.0 BENEFITS OF USING DPM

DPM has several major benefits over the current CPM approach.

- Proactive Approach: DPM allows PM's and others to review and challenge assumptions and plans before problems

arise, which increases the probability of project success.

- **Better Baseline Plans:** DPM can help PM's launch their projects with more defensible and more achievable baseline plans.
- **Better Corrective Actions:** DPM can guide PM's on which corrective actions are most effective under which set of conditions. For instance, on one task it may be better to work resources overtime to accelerate the project, but on another task it may be better to add more resources.
- **Real-World Consequences:** Everyone knows that productivity can suffer when too many people are thrown at a job or people are working a lot of overtime. DPM incorporates these types of impacts.
- **Better Risk Management:** DPM can highlight the risks inherent in baseline or midstream estimates from changes at the resource level. This helps PM's know where to focus their attention.
- **Project Acceleration:** DPM can show realistic options for "accelerating" a project, along with cost and labor trade-offs for achieving the shorter schedule. DPM can even quantify the shortest possible duration for a project based on its current structure.

## **6.0 CASE STUDY: USING AN AGENT-BASED DPM SIMULATION TOOL**

A project manager is leading a team to develop a prototype sensor system and is using Microsoft Project, which is a CPM-based tool. The baseline project plan (without leveling resources) shows a duration of 39 months at a total cost of \$13.1M (Figure 5 and 6). To remove resource over-allocations, the project plan was level-loaded. After resource leveling week-to-week, Microsoft Project shows a duration of 599 months! Clearly, this is not realistic. When the resources are leveled month-to-month, Microsoft Project shows a duration of 77 months. It is difficult to trust these results when there is such a huge variance from minor changes.

The same three plans were imported from Microsoft Project into an agent-based DPM simulation tool with the assumption of 100% productivity for all resources. The results of all three simulations were identical: total duration 71 months with a total cost \$13.9M. The DPM approach appears to provide a more consistent estimate. Standard PM tools tend to be either overly optimistic (when resources are not level-loaded) or overly pessimistic (when resources are level-loaded) using the simple CPM approach, whereas the DPM approach is realistic.

With the DPM approach, the PM also has the choice of changing the productivity of assigned resources. An additional simulation was run using a resource productivity level of 85%. This is a commonly assumed productivity level for labor resources. The simulation results from the agent-based DPM simulation tool for this scenario are a duration of 84 months and a total cost of \$15.1M. This is the most realistic expectation for this project. The ability to bring simulation and productivity into the estimating process through the DPM approach makes initial project budgets and timelines more realistic than those developed in other CPM-based tools.

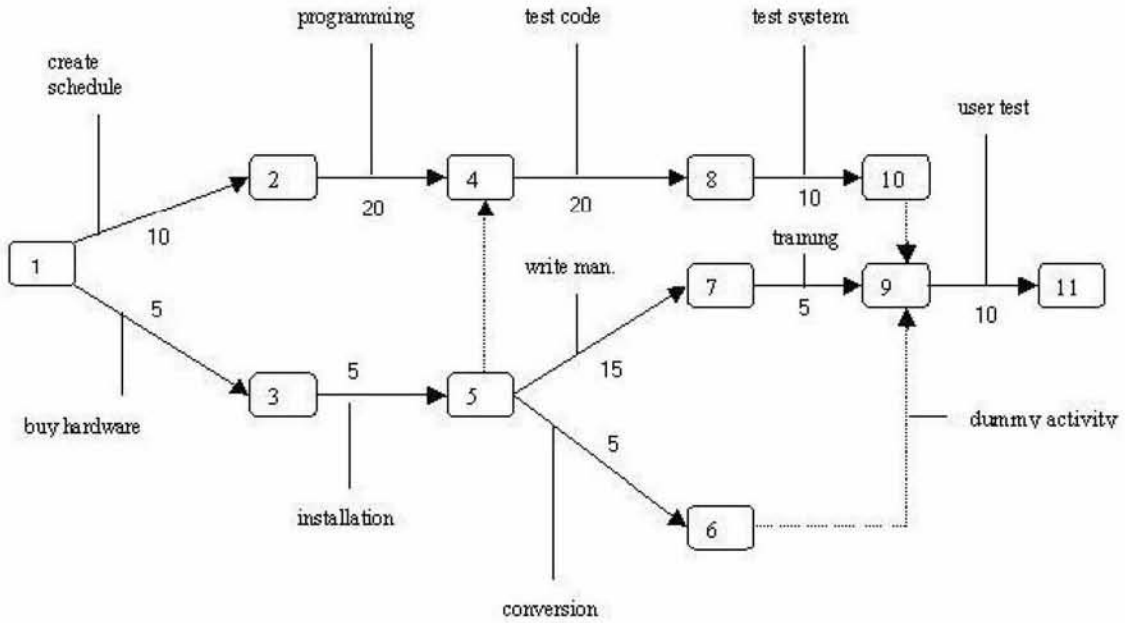

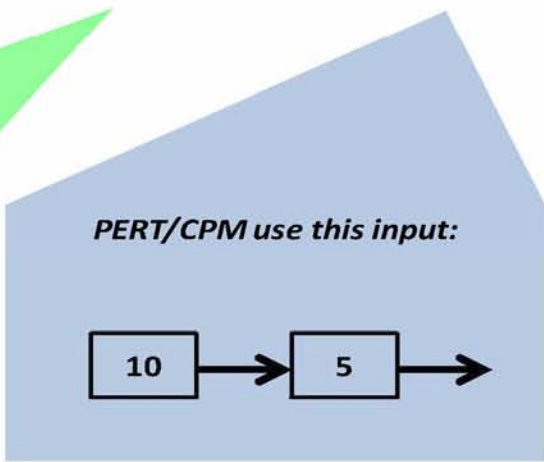
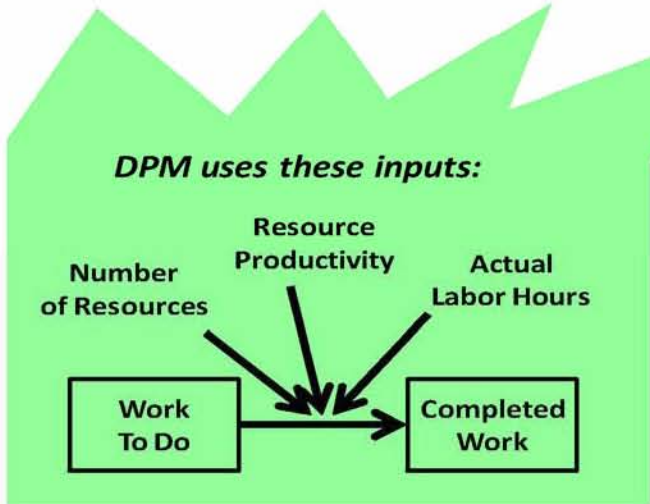


Figure 1: Simple PERT Diagram

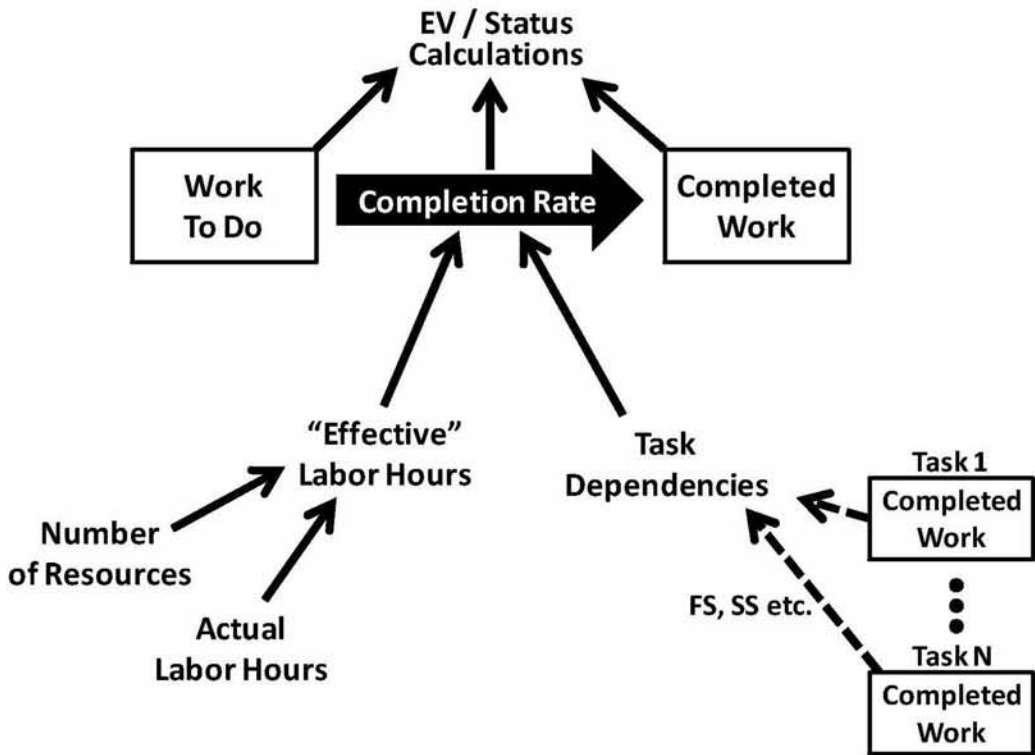
### Project Manager Assumptions:

Expected Amount of Work	+		+	Assumed Productivity Level	+	Assumed Availability (Days, Hrs/Day)	=	Expected Duration
(80 hours)		Joe (1)		(100%)		(8 hrs/day)		(10 days)



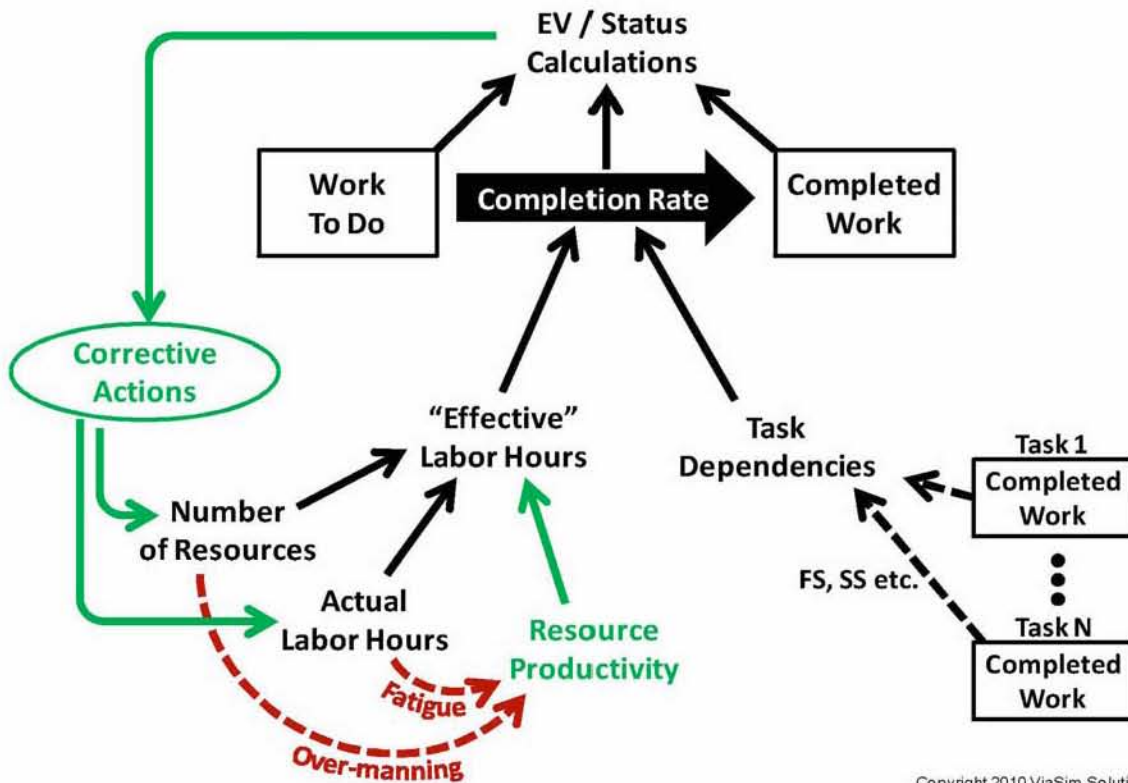
Copyright 2010 ViaSim Solutions

Figure 2: How Information Is Used as Inputs



Copyright 2010 ViaSim Solutions

Figure 3: Overview of Operational Model for Task Execution



Copyright 2010 ViaSim Solutions

Figure 4: Additional Elements Added by the Dynamic Progress Method (DPM)



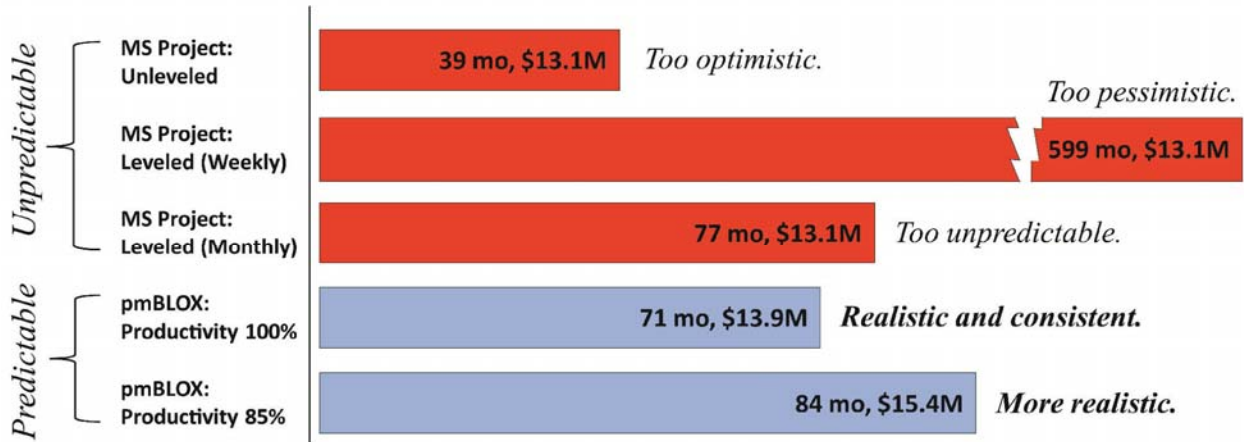


Figure 5: Comparison of Planning Estimates in Bar-Chart Form

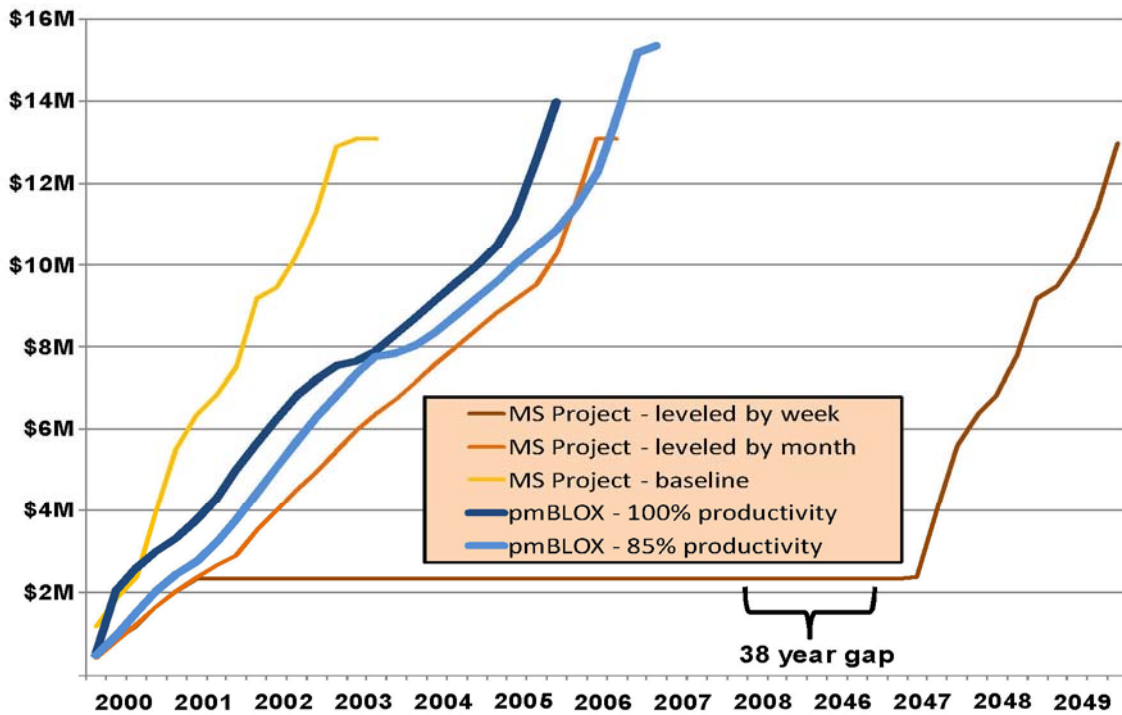


Figure 6: Comparison of Planning Estimates in Line-Chart Form