Using the Critical Task Method with the Critical Path Method

Bruce M. Nagy, President, BMN International

Introduction

This paper introduces a methodology and metrics that allow development teams to take ownership of the project schedule and improve their estimation reliability by an average of 80% based on earned value analysis. This approach will also allow development teams to translate "actuals versus estimates" into process improvements, adding increased reliability to follow-on task estimation efforts. In addition, this paper describes how this process provides a vehicle to allow development teams to achieve consensus involving solutions to complex task execution reliability and efficiency issues associated with meeting customer milestones.

Although the computer age has improved the appearance, data input ease and analytic response times involved with Program Evaluation and Review Technique (PERT) or Critical Path Method (CPM) networks, the use of these processes have remained basically limited to 1958 business applications (Kerzner 1995). In 1958, management principles such as total quality management (TQM), corporate reengineering, integrated project teams, product development teams, concurrent engineering, capability maturity model, ISO 9000 and continuous process improvement were not readily accepted or known. The scope of this paper will discuss developing and analyzing a Critical Task Method (CTM) network, integrating CPM with continuous process improvement (CPI) business structures to improve task execution reliability.

Background

CTM, having similar origins to PERT, was initially developed in 1992 in response to managing risk critical programs within the Special Projects Office of the United States Navy. In the late 80's, TQM was being implemented throughout DoD. The result of this implementation caused project management to focus on managing the process of task execution. Statements like "Doing it right the first time!" by making business processes more efficient and reliable (Walton 1986) are the current goals of all competitive organizations. Yet, the CPM paradigm primarily enforces the tracking of time and money to determine project health. The paradox being that tracking time and money focused managers on problems without providing solutions to process improvements associated with maintaining budget and schedule. CTM, a derivative of CPM, provides a development team with the means to measure and improve task execution reliability (Goldratt et al 1992) prior to cost overruns and schedule delays.

There are several assumptions associated with using CTM effectively. The first assumption is that project management staff knows how to determine a critical path from a network of tasks. CTM is simply an extension of CPM. CTM normally requires a detailed project network. Each task within the network represents a single team member's activity. There can be group tasks within the network. Yet, the group of tasks must trace to individual tasks. Therefore, using this approach may require advanced modeling tools to decrease the time to accurately develop and represent complex project networks (Nagy 1999). It will be assumed that a project team has the ability to understand, develop and represent detailed project networks.

Project Control

The measurement and tracking of labor-hour consumption, schedule integrity, or other cost and schedule variance metrics are the principal tools used by managers to control a project (Lewis 1993). Do manager's control a project? If someone went to the race track, gambled on a horse and watched the race, would that person be in control of the horse race? Tracking earned value (schedule and budget), although a necessity, provides managers with little control and a lot of observation. Where does the control begin and who does the control effect? No manager will control a development team. And no amount of finger pointing will make a horse run faster!

The key is to provide development teams with the authority to make decisions (Hammer et al 1993) that effect the topology of the project network in order to maintain budget and schedule. What does authority really mean? The best project plan matches the project tasks to the capability of the team (Drucker 1992) to achieve the customer deliverables on time and within budget. Even if the skills of the team are well matched against the project tasks, what insures that each individual on the team is willing to overcome any unforeseen obstacles during development to achieve customer satisfaction? What motivates a team? If a team has the ability to match their own experience and skills against the required tasks and create their own motivational incentives for maintaining budget

Proceedings of the 30th Annual Project Management Institute 1999 Seminars & Symposium
Philadelphia, Pennsylvania, USA: Papers Presented October 10 to 16, 1999

and schedule, then management has provided the development team with the authority to succeed.

CTM is a methodology incorporating a set of experience, skills and motivation metrics that provide a development team with the authority to maintain budget and schedule of a project plan, thereby increasing task execution reliability by over 80% based on earned value analysis.

Network Implementation

To create a CTM network, it is first necessary to create a CPM network of tasks. Again, each task must represent a single team member performing an action (or actions), linked to a series of other tasks, to meet a milestone. A task must include labor hours, start and end dates, predecessor-successor links and any other information that effects a team member's ability to understand the project manager's requirements for completing the task on time and within budget. Once the CPM network is created, the team is now ready to continue the analysis of the schedule using CTM, incorporating experience, skills and motivation metrics into the network. Yet, are there really objective metrics that can measure experience, skills and motivation?

Metrics

Experience refers to a developer's past technical experience associated with a specific task. Has the developer performed a specific task in the network under identical conditions, e.g. task description, labor-hours, "burn rate," task duration, predecessor-successor links, contractor and in-house personnel availability, or hardware/software resources. If an adequate amount of detail is available in describing the task conditions for execution, the easier this question will be to answer. Basic skills refer to the communication, mental and physical requirements of the task. Motivation refers to professional (Peters 1987) and personal incentives (Drucker 1986) associated with the developer's desire to participate in the project.

"Soft" Metrics

The first key to understanding how to measure these "soft" metrics, i.e. experience, skills and motivation, is recognizing that everyone is different (Keirsey et al 1984). Therefore, relating one measurement to another would be like measuring two books using different systems. One book in inches would measure 10 units. Another book in centimeters would measure 10 units. Although the number 10 would be the same, not knowing the system of measurement could cause confusion. The same holds true when using "soft" metrics. Although, it may appear the measurement system is the same, since people are different, the units measured all have different meaning.

Reliability

When a team measures a project network using these "soft" metrics, they are measuring the reliability of the schedule. A reliable schedule describes a realistic schedule. An unrealistic schedule provides the team with a general idea on approach but not a plan of action. Therefore, experience, skills and motivation metrics are referred to as reliability metrics. If a team has past experience and is motivated to accomplish the tasks on time and within budget, the schedule will be reliable. Time and money based metrics are referred to as efficiency metrics. Sometimes, efficient schedules are not reliable. Cutting cost and time without assessing impact can sometimes cause excessive cost overruns and schedule delays. On the other hand, reliable schedules can sometimes mean that the development teams aren't being creative in determining "better, cheaper and faster" methods of meeting customer needs. Therefore, management needs to use both types of metrics to manage the risk of the project.

Methodology

Once the development team has created a CPM network, there's a seven step process in using CTM. Step 1: The basic methodology consists of measuring each task in terms of experience, skills and motivation. Step 2: After the measurements are complete, the critical tasks within the critical path are determined and ranked. Step 3: Once determined, individuals use a series of algorithms to determine the measurable increase in their experience, skill or motivation from their original values. Step 4: If the individuals on the team can not determine how to increase their motivation, then group problem solving sessions are conducted. Step 5: Once conducted, each individual on the development team determines their increase in experience, skills or motivation. Step 6: At this point, it is necessary to create a CPI structure to use the "soft" metrics as reference points in learning how to raise experience, skills and motivation with greater reliability. Step 7: Once a CPI structure is created, implementing a lessons learned database based on the CPI structure is the last step before task execution and the tracking of "actuals versus estimates."

Measurements

There are a variety of ways to measure experience and skills. The easiest way is to ask the question: Is a developer's past experience (type of task, time frames, labor-hours defined, work environment, hardware/software tools

used) identical to the current task execution conditions? Any variation may still indicate an exact match, if the variation is transparent in terms of the developer's efforts to complete the current task being measured. An answer of no or yes, numerical (1) or (2), can be recorded. This question can be asked for each task in the CPM network. More complex rating systems involve a greater range of answers to the question. For example, a (1) to (5) grading system may allow the developer to answer in a variety of ways: (1) I never completed this type of task. (2) I completed a similar task. (3) I completed an identical task. (4) I completed many identical tasks. (5) I completed one or more task under more difficult conditions. (Examples of more difficult work conditions could be shorter time frames, less sophisticated tools, little or no support structures as compared to the task being measured.)

A (1) to (10) rating system has been used successfully to measure experience, where several answers are equal to the same number, e.g. two answers are both a (3). Therefore, even though the rating system is (1) to (10), there may be twenty answers. Each measurement has two answers. Each pair of answers is equal in measurement. This allows for team variations in answering the questions, while maintaining a ten scale system. A ten scale rating system provides the opportunity to define similarity, ease and difficulty with greater clarity and resolution. As a minimum, the following set of answers is recommended: (1) I've never attempted a technically similar task. (2) I've accomplished a technically similar task under (2a) easier, (2b) similar or (2c) harsher conditions. (3) I've accomplished an identical task under (3a) easier, (3b) similar or (3c) harsher conditions. Depending on how these answers are paired, the experience measurement can be a (1) to (7) or (1) to (5) rating system.

Basic skills can also be measured in a variety of ways, from complex to easy. A (1) to (5) simple rating system is described as follows: (1) I do not have the skills to complete this task within the conditions defined. (2) I believe my skills will definitely be challenged to complete this task within the conditions defined. (3) I believe my skills may possibly be challenged to complete this task within the conditions defined. (4) I believe my skills will not be challenged to complete this task within the conditions defined. (5) I have the skills to complete this task within the conditions defined. The trick with measuring skills is that the developer only needs to take this measurement if the experience measurement was low. In other words, if the developer has the experience, then the developer has the skills. Basic skill measurements can be used to determine if the developer feels confident about his skills. More sophisticated basic skill measurements ask a series of questions that rate the confidence of the developer and also describe the types of skills needed to complete the task successfully.

Motivation requires a specific measurement approach and historically does not allow for many deviations. There are five basic category pairs that cause people to be motivated: (1) Acceptance/Recognition; (2) Adventure/Fun; (3) Ambition/Accomplishments; (4) Comfort/Security; and (5) Money/Finances. The reason why there are two motivational elements in each pair is based on the variations in the way people perceive each element. A developer might want to be recognized for his achievements. Another developer may want to have his ideas accepted. Both developers may be talking about the same experience using slightly different language. Measuring motivation involves comparing what a developer considers important in terms of motivational pairs as to what he can expect to gain during the course of completing his tasks, and meeting budget and schedule.

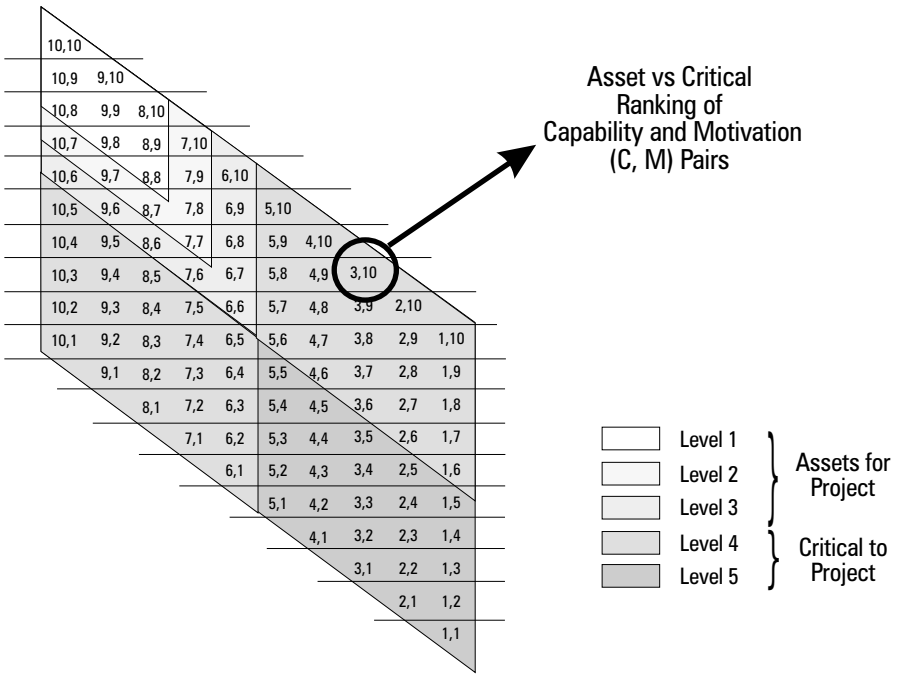
Calculating the motivation of an individual to complete his project tasks reliably requires a weighted average calculation. Simply stated, the weights are created from determining the level of importance that each motivation pair has in terms of the developer's personal and professional goals. Levels of importance can be ranked between (1) and (10). Each motivational pair must be ranked in order of importance. There are formal processes to use to insure the ranking is accurate. The same (1) to (10) rating system is used to measure the motivation of the developer to complete the task. The question asked is: Based on the developer's past experience and instinct, what outcome is more likely to be felt, seen or heard while completing the assigned task? Ratings of (1) through (5) indicate the levels in which a positive experience is likely. Ratings of (6) through (10) indicate the levels in which a negative experience is likely. Again, the above discussion is simplifying the approach, but hopefully provides an understanding of the metric. A time weighted average calculation determines a single motivational value of the developer in completing the task on-time and within budget.

Using the above approach to measure experience, skills and motivation, the development team can measure each task within the project network.

Ranking the Critical Tasks

For ease of this discussion, the term capability will be used to describe both experience and skill measurements. In viewing a CPM network, both the critical tasks and tasks not in the critical path are measured. Those tasks not in the critical path are called asset tasks. Exhibit 1 describes the ranking of task in terms of a (1) to (10) rating system for both capability and motivation measurements. The ranking is based on not having any prior knowledge

Exhibit 1. Capability and Motivation Critical and Asset Rankings



about the team's performance. Without prior knowledge, capability and motivation are weighted equally in the ranking. Asset tasks are task where resources can be consumed to support tasks on the critical path. When listing assets, a development team has to solve problems, asset tasks are a major "asset" for the project. Using Exhibit 1 as an example, tasks within the critical path can be ranked in terms of most to least critical.

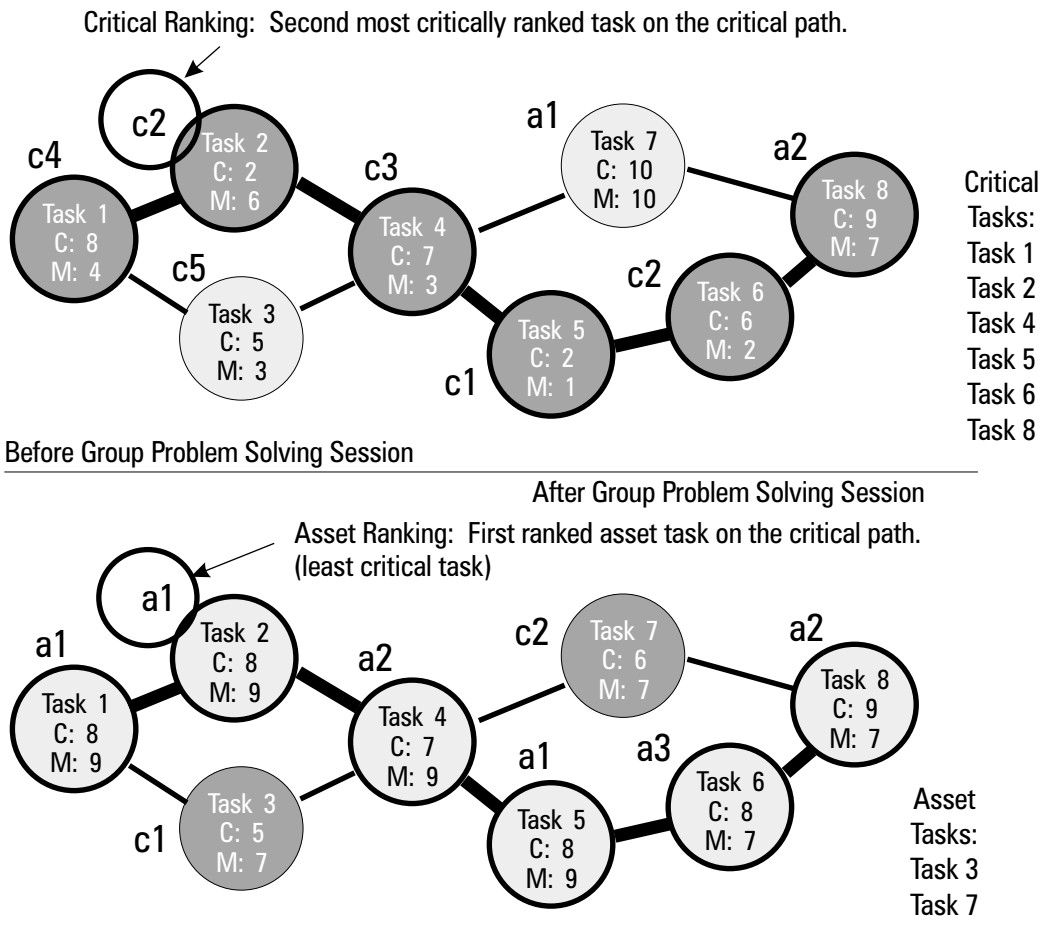
Individual Problem Solving

There are basically five general methods to increase motivation and five general methods to increase capability. Problem solving consists of improving the capability and motivation rating of a task. Again, a team member can improve his capability to complete a task by (1) increasing his confidence, knowledge or training, (2) using a different task approach, (3) modifying the physical resources, (4) modifying the support of human resources, or (5) increasing the incentives to perform the complexities of the task. Likewise, a team member can increase his motivation to complete a task by (1) increasing the incentives to perform the complexities of the task, (2) using a different task approach, (3) modifying the physical resources, (4) modifying the support of human resources, or (5) increasing his confidence, knowledge or training to do the task. A formal methodology has been developed to assist individuals to reliably raise their capability and motivational levels associated with completing the critical tasks on time and within budget. Yet even with a formal methodology, an individual may be within a paradigm, requiring team support (Deming 1986) to think "out of the box."

Group Problem Solving

This is also a formal methodology. If individual problem solving does not solve all the critical task issues, then a team problem solving session is required for "out of the box" thinking. The most critical tasks always have priority in being solved by the team. Each team member having their tasks ranked in critical order allows the group to support a paradigm shift. The sessions begin with the individual describing the task, why the capability or motivation is low, any ideas in increasing these values and what concerns are at issue. The team responds with a single idea to increase the capability or motivation of the fellow developer, followed by concerns. Each concern is addressed with another idea (or modification of the original idea). This approach continues until a solution is reached. This approach allows all ideas and concerns to be addressed. In addressing concerns with ideas, the result becomes a convergence to the solution. Sometimes the solution requires separate "tiger teams" to investigate alternatives. Having a wide variety of technical backgrounds supports the ability of the

Exhibit 2. Improvement in (C,M) to Support Critical Task Reliability

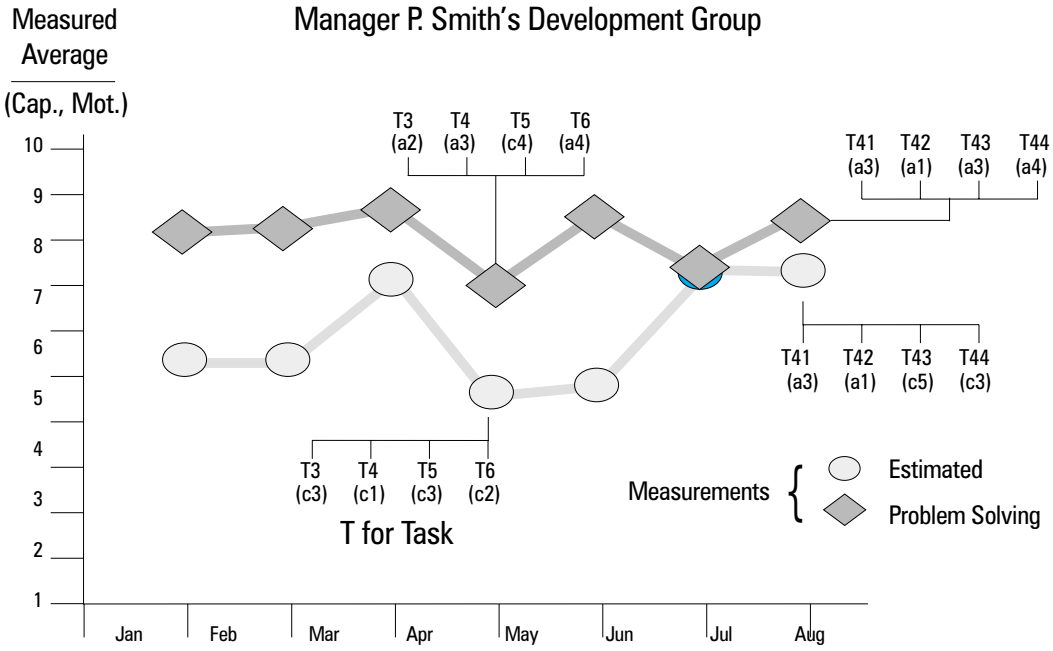


team to "break" paradigm and discover creative and usable ideas. The team recognizes that they have achieved a solution when the individual whose task is being analyzed recognizes that the concerns are addressed to his satisfaction and the ideas provide him with an increase in capability or motivation to perform the task reliably. Many times, as stated earlier, an asset task is used to consume time or resources to support a solution in increasing capability and motivation measurements along the critical path. The result of this shift in resources will cause the asset tasks to have a decrease in "soft" measurement and the critical tasks to have an increase as described in Exhibit 2. This group problem solving approach has been used with great success over the last seven years.

Metric Evaluation

When solutions are achieved using the individual or group problem solving methodology, each team member then measures the solution (new approach) using the same metric definitions for experience, skill and motivation as measured in the previous steps. When a "soft" metric is used, as stated earlier, the units of measurement are unknown. Yet, the value is defined. The question remains, what is the unit as related to other developer's measurements? That answer is that for problem solving solutions it doesn't matter. As an individual measures experience, skills and motivation per task, the unit of measurement becomes absolute. In other words, if a developer measures a 3 on a task, his reference point is a 3. What is important in measuring "soft" metrics is that the problem solving sessions improve reliability, increasing a 3 to a 7, as an example. This is true for each task. If a team member's average was a 5, again, it would be important that the average is increasing after each problem solving session. This is the approach in creating a continuous process improvement environment. If a team member can measure an increase in experience, skills and motivation, then reliability in terms

Exhibit 3. Measurement Increases in (C,M) Based on Problem Solving.



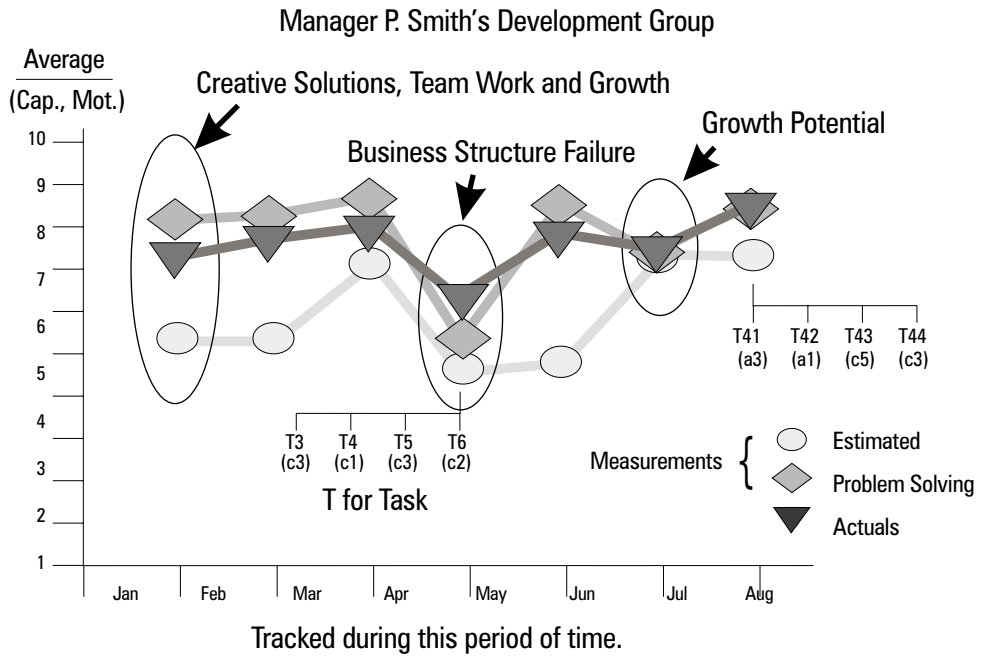
of task execution can increase and risk decreases. The key is to provide the team with an opportunity to problem solve the CPM network, especially the critical path, in terms of increasing their experience, skills and motivation. Again, it is not the relationship between the units that is important, it is that the units increase after each measurement as described in Exhibit 3.

Establish CPI Structure

When creating a CPI structure for a task execution environment, it is important to recognize that it is a belief system that is being constantly monitored and improved. Therefore, the CPI structure requires that the continuous process improvements effect the teams belief that the job can be done on time and within budget, ergo "soft" metrics. Measuring and improving the experience, skills and motivation values effects the teams belief that the project can be completed on-time and within budget. One belief system that the experience metric provides is, "If I have performed a like task in the past, I can perform this task with at least as much reliability." The basic skill belief system states, "If I have the basic skills, I have the ability to meet the schedule." Finally, the motivational belief system suggests, Because of my motivation, I will develop the skills and gain the experience of completing this task on-time and within budget."

Based on project uncertainties, it can be anticipated that the development team will be challenged to meet budget and schedule. Likewise, what will also be challenged is the team's belief system that they can meet budget and schedule. If this belief system is not addressed, no counter project strategy will be efficiently implemented. Therefore, when schedule delays and budget overruns exist, there must be a group problem solving session available for the team to respond by determining ways to increase measurements in experience, skill and motivation based on lessons learned from past measurements related to actual results. In other words, using a simple example, if a developer is late on finishing his tasks and he had scored high in capability and motivation, the developer either did not understand the requirements of the tasks or had a motivational change during task execution. For each "actuals versus estimate" in tracking time and money, there is corresponding "actuals versus estimate" in tracking experience, skills and motivation. Understanding how to continuously increase experience, skills and motivation values associated with the tasks within the project network

Exhibit 4. Tracking of Capability and Motivation "Actuals vs. Estimates"



becomes the purpose for the CPI structure. Finally, since human factors are a main issue in any belief system or problem solving approach (Mobley et al 1989), management needs to develop a business culture that supports the growth of maturity in dealing with project issues. This approach is discussed in (Nagy 1999).

It is recommended that the development team and project management staff meet during periodic intervals to review lessons learned based on increasing experience, skills and motivation of the team.

Establishing a Repository

Establishing a repository consists of creating a lessons learned database associated with the project management software. Therefore, as the team tracks "actuals versus estimates" in time and money, the database will store how these "results" based metrics improved the "process" of learning to better match experience, skills and motivation of the team to improve the estimation reliability of the project network. Exhibit 4 graphically describes "actuals versus estimates" associated with capability and motivation measurements.

Balancing Efficiency and Reliability Metrics

Eventually, when the team's paradigm of belief is established, the project manager can begin to challenge the team in group problem solving alternate task approaches in reducing schedule while maintaining reliability. This can only be accomplished once a belief system is in place stating that the team's ability to estimate task duration and quality is reliable. Using "soft" metrics to make the project plan more efficient while maintaining reliability is a tool available to a project team prepared for group creativity within their group problem solving sessions.

Summary

CPM offers many valuable insights into the risk assessment of the project. Yet, there is a select set of critical tasks within the critical path that are the key factors in assessing risk. Once a team recognizes these critical tasks within the critical path, project managers have increased control in overcoming potential project delays or cost overruns.

Normally, CPM is determined through use of a time metric. CTM is calculated using three additional metrics. These additional metrics are based on an individual's ability to reliably execute a task within the time and budget parameters defined. The first metric measures the individual's past experience as related to assigned tasks needing to be completed during the life-cycle of the project. The second metric measures the individual's basic skills per tasks assigned. The third metric measures the individual's motivational needs. Motivational needs refer to both

professional and personal goals achieved through project participation.

Applications

CTM has been used for a variety of projects and organizations over the last seven years ranging from two hundred to seven hundred tasks within the network. Project types included hardware communication systems and large scale software development programs. It has been consistently noted that when a developer determines that all three areas involving experience, skills, and motivation do not match the execution parameters of the task defined by management, those critical tasks, if left unmanaged within the critical path, will cause the project/program to experience cost overruns and schedule delays.

Value Added

Most project managers have recognized that the essential challenge of managing a project is matching experience, skills and motivational needs of individuals executing tasks against the budget and cost constraints of the project. The reality is that "good" project managers have always intuitively assessed the individuals on their team in terms of their ability to complete the tasks assigned within time and budget. Use of these "soft" metrics has enhanced the project manager's ability to make decisions on risk management and effectively prioritize resources based on accurate data collected by the team members before task execution.

Results

Using CTM has allowed teams to anticipate potential critical path problems and discover solutions before these problems become realized. The use of CTM has increased project execution reliability nominally by 80%. One case study demonstrated 500% estimation reliability improvements (using before and after earned value measurements). This dramatic increase in reliability is principally based on the teams ability to communicate and support other team members in matching experience, skills and motivational needs to the project milestones. When the team is allowed to take ownership and responsibility using CTM, project risk is forecasted with greater accuracy and managed more effectively.

References:

Kerzner, Harold, 1995, Project Management: A Systems Approach to Planning, Scheduling and Controlling, New York, NY: Van Nostrand Reinhold.

Walton, Mary, 1986, The Deming Management Method, New York, NY: Perigee Books.

Goldratt, Eliyahu M. and Cox, Jeff, 1992, The Goal, Croton-on-Hudson, NY: North River Press.

Nagy, Bruce M., 1999, Modeling and Rehearsing Project Management, Philadelphia, PA: PMI '99, Project Management Institute.

Lewis, James P., 1993, The Project Manager's Desk Reference, Chicago, IL: Probus Publishing Company.

Hammer, Michael and Champy, James, 1993, Reengineering the Corporation, A Manifesto for Business Revolution, New York, NY: Harper Business.

Drucker, Peter F., 1992, Managing for the Future, The 1990s and Beyond, New York, NY: Truman Talley Books.

Peters, Tom, 1987, Thriving on Chaos, Handbook for a Management Revolution, New York, NY: Harper Perennial.

Drucker, Peter F., 1986, The Practice of Management, New York, NY: Harper Business.

Keirsey, David and Bater, Marilyn, 1984, Please Understand Me, Del Mar, CA: Prometheus Nemesis Book Company.

Deming, W. Edwards, 1986, Out of the Crisis, Cambridge, MA: Massachusetts Institute of Technology.

Mobley, Lou and McKeown, Kate, 1989, Beyond IBM, Washington D.C.: Enter Publishing.

Nagy, Bruce M., 1999, Playing "Milestone: The Game," Philadelphia, PA: PMI '99, Project Management Institute.