Chapter 8: Project Planning: Estimation of Task Durations, Cost and Schedule Considerations

The Why, What and How of Project Estimating

CHAPTER OUTLINE

Techniques for Estimating Task Duration and Cost – Inputs, Tools & Techniques, and Outputs
Expert Judgment Estimation Pitfalls
Multitasking
Delays Accumulate—Advances Do Not
Safety—A Bad Thing
Using Estimates from the Assigned Human Resources in IT Projects
Goldratt and Critical Chain Project Management Methodology
Cost Estimation – Analogous Estimating, Parametric Modeling, Bottom-up Estimating, and Computerized Tools
Software Cost Estimation Methods
Special Problems in Cost Estimation/Budgeting
Reducing Project Duration
Summary and Conclusion
Exercises
References

LEARNING OBJECTIVES

After reading this chapter, you will be able to:
1) Understand the difficulties associated with estimation of both time and cost
2) Understand the implications for schedule and budget
3) Comprehend how to overcome certain 'behavioral' problems with estimation

Chapter 8

8

Jack, Jody, and John are having a discussion about project estimation. This is what they are saying:

Jack: Well, here we are back to square one and having to estimate how long it's going to take us to do our assigned tasks.

Jody: Let's see, what is the rule? Do your best in calculating how much time is required and then take that number and double it?

John: Something like that!! I'm not forgetting what the project manager did to us the last time we turned in our estimates.

Jody: Cutting our estimates by 50% if we said we were completely confident we could finish our tasks in our estimated times! This time for sure, I'm going to double my estimate in anticipation of his 50% cut.

Jack: He said that our estimates should have about a 50% probability of getting completed on time. If our estimates are 80 or 100% probable of getting completed on time, we have built too much safety into our estimates. Safety, safety—so what's with this safety business??

John: And, do you remember what happened to my project before last? I actually completed my task ahead of time by 20% and turned it in ahead of time. So they gave me a lot more work to do in the little time I had left. Then, they penalized me for not being more accurate in my estimate, and on the next project, my estimate was reduced by 50%. If you finish ahead of time, just wait until your time is up before you turn in your deliverable and declare your task done. Otherwise, the project manager will think there is something wrong with your estimation and will be unhappy with you.

Jack: He does have a point, though… Our projects are never finished on time and that is in spite of the fact that we are all adding a lot of safety time into our estimates.

Jody: I think we should try to understand why this is happening—why we aren't finishing our projects on time in spite of our super conservative estimates. Perhaps there is something we could learn here.

Jack: Last time I purposefully gave myself twice as much time as I thought I needed to do the assigned work. But then I waited until the period for doing the work was half over before getting started. I wound up taking an extra week longer to get the task completed. I guess I should have really believed my conservative estimate.

Jody: That is called student syndrome. You should have learned not to do that when you were in college.

John: Last time I estimated how many hours of actual clock time it would take me to complete the work and then translated that to calendar time. But I forgot to take into consideration the other two projects I was working on. They suddenly became very demanding right during the period in which I was required to do this critical task. I wound up finishing the task late by a week.

Jody: This is called multitasking and you should do no multitasking at all when completing critical task.

TECHNIQUES FOR ESTIMATING TASK DURATION AND COST

Every project is made up of steps that must be completed in order for the project to be finished. Two of the most important steps in any type of project are project scheduling and project budgeting. Project scheduling takes the previously determined project activities (tasks) and puts them into a timetable. Project budgeting takes the allotted funds for a project and decides how and when they will be spent. Each of these steps involves making assumptions in order to determine the most accurate estimate. Both scheduling and budgeting are hinged upon estimations of the durations of the tasks that make up any project. Several techniques can be applied in order to help the project manager make these determinations.

As noted earlier, the project activities will have already been established before the project schedule is developed. The activities can be viewed as one “input” in a collection of inputs that must be present before the project schedule can be developed. Typically, these activities come from a Work Breakdown Structure, as discussed in Chapter 7 or from use of a predetermined project template of activities.

There are six main inputs that are needed in order to form a project schedule: an activity list, constraints, assumptions, resource requirements, resource information, and historical information. Through the different tools and techniques used to estimate task duration, these inputs will be transformed into the project schedule and budget.

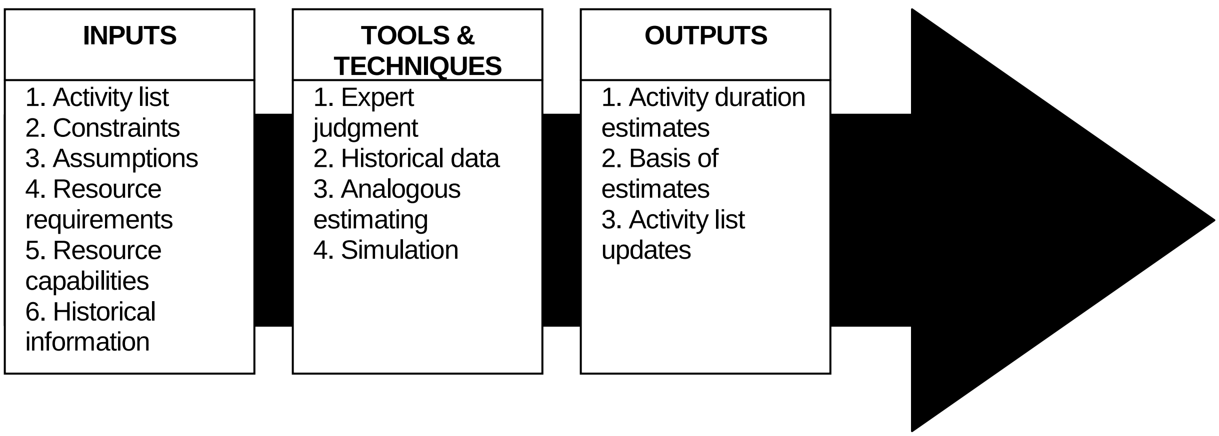


Figure 8.1 Task Duration and Cost Estimating – Inputs, Tools & Techniques, and Outputs

Estimating how long each activity will take, from start to finish, is the first step in developing the project schedule. Estimation is, however, one of the things we don’t do well, without some experience and maturity. One problem is that human estimators tend to have wide variability in their estimates of the effort required to do a piece of work. It has been said that the same task under the same conditions will be estimated differently by ten different estimators or by the same estimator at ten different times. The time for the work to be done plus any associated waiting time (total elapsed time) is required of the task duration estimate.

There are many different techniques used to estimate task duration. Each firm might have its own technique that is used for every project, while other companies may use a combination of techniques to estimate task duration. Some examples of techniques that can be used include expert judgment, historical data, analogous estimating, and simulation.

Expert judgment can be used when there is a group or individual with specialized knowledge or training in the specific task being estimated. However, a downside of an expert judgment analysis is that there may be an existing bias in the form of individual preferences or past experiences. This estimation method contains the highest risk factor of the four methods.

Historical data can be used to improve the accuracy of task duration estimates. The information can be obtained by studying previous projects and talking with team members. There are several factors to consider when looking at historical data. These factors may be technical difficulty, project environment, availability and capability of resources, and project constraints. Previous documentation such as network diagrams and WBS can also be helpful. The goal of using historical data is to look at past performance in order to make the most accurate predictions for the current duration estimates.

Analogous estimating, or top-down estimating, uses the actual duration of a previous activity similar to the current activity in order to estimate the task duration. It is most reliable when the previous activities are similar in fact and when the individuals preparing the estimates have the needed expertise. This estimate is also made without the use of any engineering data.

Simulation is another method of estimating task duration. Simulation is done by calculating multiple durations with different sets of assumptions. The most common form of simulation is Monte Carlo Analysis. This form defines a distribution of probable results for each activity and calculates a distribution of probable results for the total project. The simulation method involves the use of well-engineered research data about the specifics of the program, and has the lowest overall risk factor of the estimation methods.

As successful companies complete projects and gain more accuracy in their estimation efforts, many will attempt to standardize their procedures by creating an *estimating manual*. These manuals often provide much better estimation methods for the company, because they are tailored to the individual organization and are able to highlight specific strengths and weaknesses of the organization's efforts. In addition, other factors can be noted and compared, such as machine specifications and capacities, worker downtime, labor unions, and seasonal fluctuations in supply and demand of inputs.

The outcome of the previous techniques is several activity duration estimates. They are quantitative and should always include an indication of the range of possible results. For example, a task duration estimate might be three weeks plus or minus three days. Task duration estimates should be made aggressively. If it is estimated that a task that should take three days will take ten days, it will probably end up taking the whole ten days. People tend to use the amount of allotted time even if it could be finished earlier.

EXPERT JUDGEMENT ESTIMATION PITFALLS

In spite of its inherent problems, team members are often asked to estimate how long it will take them to do the task(s) they have been assigned. There are several reasons why doing so makes sense. First, probably no one knows better than they do how long a task will take them to complete. Second, the team member usually has "his feet held to the fire" with regard to the estimate; that is, he or she is expected to complete his or her task within the time estimate he or she chose. Third, by asking the team player to estimate the length of time required to complete a task, that team player has some ownership in the estimate and in the overall project plan. This ownership translates to a commitment to complete the task by its due date.

Here is what typically happens with regard to expert judgment estimation. When asked to estimate your task, you think about the task and the effort and decide that you can do the task in, say five days. Then you think a little bit more. There may be something unfamiliar in the task. You worry about the effect of unplanned work interruptions. Finally, you want to make sure that you won't be late on your estimate because you don't want negative attention. Based on all this uncertainty, you announce that you can do the task in, say ten days. Since you're new at the task of estimating and since you don't have a personal history database of actual times required to complete similar tasks, and since your professor said to take your "due-diligence" estimate and double it, you decide that ten is the correct number of days.

What you have done here, is to add five days of safety to your original estimate. You have hidden five days of safety in your ten-day task. We say the safety is hidden because the task is entered into the project planning software and database as a ten-day task. The five days of safety is your private contingency factor. It's important to note that adding safety into your estimate isn't necessarily wrong. It's a reasonable thing to do considering the factors involved and the project management environment in which you work. After all, you don't want to be the one that misses a task due date. **Safety** is protection time placed in an estimate to ensure completion on time. Some project players will more than double their estimates and this makes the total project take twice as long as required and cost twice as much.

Now let's consider what happens when the task is actually performed. In his business novel, *Critical Chain* (1997), Goldratt tells the story of what happens when a professor gives a class assignment that is due in two weeks. The students complain that the assignment is tough and will require more time. The professor agrees and gives them additional time. Later, when the students look back on how they actually performed the assignment with this additional time, they note that they all had plenty of time, with safety, to do the assignment so they put off starting until the last minute anyway. Let's look at how this **student syndrome** can affect your task and the whole project.

Given the student syndrome, you put off really getting to work until the fifth day of the task. This start should be OK because you have adequate safety in your estimate. Unfortunately, four days later, you encounter an unexpected problem with your task. Suddenly, you realize that your safety is gone and that you will overrun your estimate no matter how hard you work. You spend the next five days working as fast as you can, with an overrun of 30% of your original estimate.

This simple task scenario is not unusual. It happens over and over again in the completion of a project. We are all human, and when we establish a task schedule with a hidden safety margin, most of us naturally fall into the student syndrome.

According to **Parkinson's Law**, work expands to fit the allotted time. Most of us have heard about Parkinson's Law and seen it in action on projects. If a task is estimated to take 10 days, it usually doesn't take less. This adjustment of effort to fill the allotted time can come in a number of ways. Software projects often exhibit a tendency towards creeping elegance when the developers sense that they have more time than actually necessary on a task. In other cases, people will simply adjust the level of effort to keep busy for the entire task schedule. As discussed in the opening scenario, traditional project environments stress not being late, but they do not promote being early. In fact, an early completion may be dis-rewarded as discussed in the opening scenario. The traditional project culture actually encourages Parkinson's Law effects.

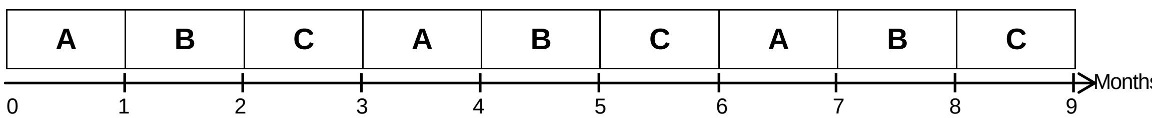
MULTITASKING

Multitasking is doing several "things"—tasks, projects—at once, concurrently. It means starting several "things" without first finishing any of them. Suppose you have three tasks, A, B and C, each taking three months to complete. You could work on A for a month, then B for a month and then C for a month until they are all complete. This would be called multitasking because you started three tasks before completing any of them. Or you could work on A until it is finished, then B until it is finished and finally C. This would be single-tasking because you worked on only one task at a time until it was complete.

Most of us work in a multi-project environment. We all have experiences of having to stop working on one task so that progress can be accomplished on another task in another project. Often, we wonder if all this jumping around makes sense because it comes with the penalties of reduced focus and loss of efficiency. However, there is a reason for this multi-tasking environment as discussed next.

Project managers are responsible to a customer for successful completion of a project. These customers can be internal or external to an organization. Customers have a tendency to be demanding. They think that their project is the highest priority and they want to see frequent progress on their project. Resources tend to migrate between projects in response to the latest, loudest customer demand in an attempt to keep as many customers satisfied as possible. This focus on showing progress on as many active projects as possible is the major cause of multitasking. As we will see, this focus is to the detriment of the overall project throughput of the organization.

Let's consider the bad effects of multitasking in a simple multi-project example. Assume we have three projects, A, B, and C, each of which is estimated to take three months to complete. Our project environment is one of organized chaos. Resources migrate from one project to the next to show as much simultaneous progress as possible to the project customers. To keep this example simple, let's assume resources work one month on each project and then migrate to the next project. In this environment, the projects are accomplished in intermittent spurts as shown in the illustration. The completion date of each project is noted with a red milestone. Note that this example assumes zero efficiency loss due to changing tasks so it minimizes the real-world bad effects of multitasking.



Now, let's assume we get organized with the simple goal of doing work based upon which projects are most important. This is an important change; we are moving from organized chaos based upon sub-optimized micro-level decisions to an optimized situation based upon macro-level decisions. For our example, let's assume the project priority, from highest to lowest is A, B and C. By eliminating multitasking and executing our projects by priority, we get the results illustrated in Figure 8.2.

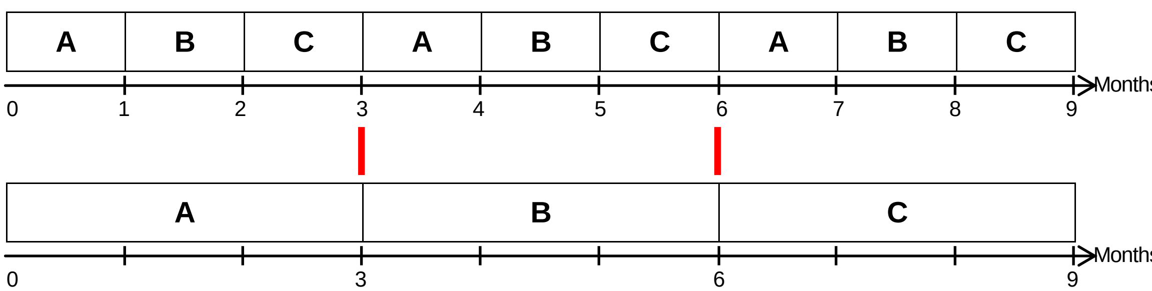


Figure 8.2 The Multitasking Scenario

Note how the lowest priority project, C, is still accomplished on the same date as the multi-tasking example. However, the highest priority project A is done four months sooner—a 225% improvement. Project B also is done in much less time than in the multitasking environment. The message is clear, if you eliminate multitasking and make resource allocation decisions based upon project priority, you get better performance on your projects.

Clearly, by working on A and nothing else, A is finished at the end of month 3. If, on the other hand, A is multitasked with B and C, then A is not finished until the end of month 7. Similarly, B is finished at the end of month 6 when single-tasked in sequence after A. But B is not finished until the end of month 8 if it is multitasked with A and C.

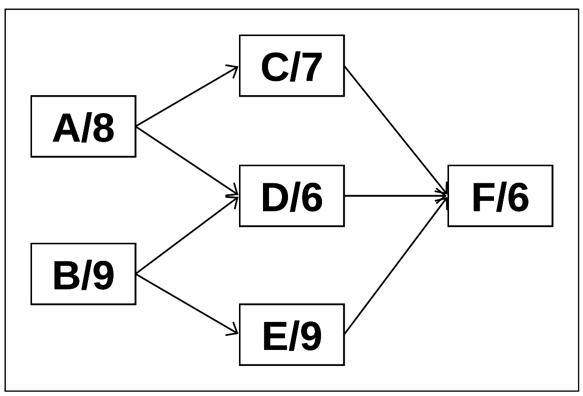
In reality, the situation is much worse than depicted in the figure above because every time you stop something and start doing something else, you undergo a setup—a period of time in which you are unproductive while you're figuring out what exactly to do. Now, instead of multitasking once a month you changed tasks three or four times a day, which is quite characteristic of reality. Clearly, your productivity is severely eroded because of the setups entailed.

The elimination of multitasking also applies within a single project. The demanding customers can be work package managers who demand progress from limited resources. If the resources are allocated to silence the squeaky wheels, the project can suffer unnecessary delays as tasks are performed in an un-optimum sequence. Later, we shall see how the Goldratt's Critical Chain method gives us a simple method for eliminating this intra-project multitasking with clear, concise rules for which work should be done first.

DELAYS ACCUMULATE—ADVANCES DO NOT

Goldratt (1997) is quick to point out that delays in the completion of tasks accumulate, whereas advances do not. To see this, consider the following network. There are four paths in this network leading up to the final task, task F. The following table presents an interesting situation:

PATH	Estimated time	Actual time
A-C	15 months	10 months
A-D	14 months	10 months
B-D	15 months	10 months
B-E	18 months	20 months



In the above network, all of the paths leading to the last task, task F, finish early except the tasks on the critical path. Originally, task F was scheduled to start on month 19. However, according to the table above, all of the paths except path B-E would allow for task F to start on month 11. So when is task F actually permitted to start? On month 21, of course, as a result in the delay that occurred in path B-E, 2 months late!

So what happened here exactly, the advances (early finishes) on the part of paths A-C, A-D, B-D did not get passed on, but the delay (late finish) in path B-E did get passed on, resulting in a delay to the entire project. There can be exceptions, certainly. Consider the following scenario.

PATH	Estimated time	Actual time
A-C	15 months	17 months
A-D	14 months	17 months
B-D	15 months	17 months
B-E	18 months	17 months

In this scenario, you have late completions on all non-critical paths, but an early completion on the critical path. In this case, task F gets to start one month early.

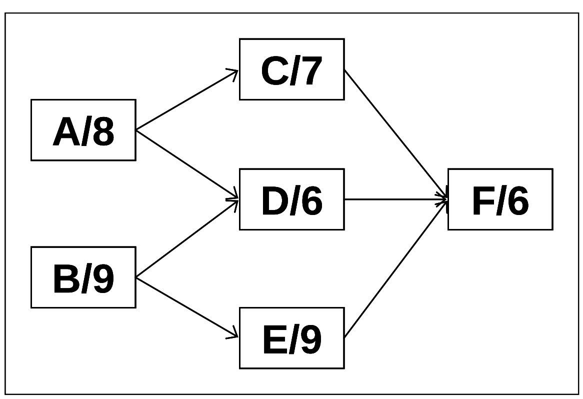
SAFETY—A BAD THING

As mentioned, it is commonplace for project players, especially if they are not programmer types, to add safety to their estimates. What happens after this as these estimates get passed up the project hierarchy?

The scenario gets worse. Often a team leader will add still more time to the estimates of his team members before passing times onto a project leader. The project leader will, to ensure completion on time, add still more safety time to the estimates to ensure on-time completion. Consequently, most of the time that has been placed into the tasks is safety time, according to Goldratt. The exception, of course, is for naïve optimistic programmer/developers who under estimate how long it takes them to do stuff. Generally, the programming community has a notorious reputation for underestimation of the time it will take to get tasks done. Worthy of note, however, is the fact that developer productivities may vary by as much as seven or eight to one. What takes one developer one week, might take another seven weeks or longer.

There are several things that can be done to detect whether safety has been added to the estimate. Ask the estimator how confident he or she is in the estimate that has been put forth. If they tell you they are 100% confident they will complete the task within the allotted time, most certainly they have added a substantial amount of safety to the estimate. Actually, a zero-safety estimate is one in which there is only a .5 probability of finishing on time. It is recommended that tasks be broken up to align and compare with a standard set of catalogued tasks whose time durations are well known. Then that duration needs to be adjusted depending upon the "natural productivity" of the developer. This number can then be compared with the developer's estimate and the estimate adjusted appropriately.
According to Goldratt, safety should be removed. The reasons for why it makes sense to do so is because safety consumes time and cost but adds no value. Safety is usually lost because: 1) of student syndrome (procrastination), 2) of multitasking, and 3) early finishes do not translate into early starts for subsequent tasks, generally, whereas late finishes do get passed down to subsequent tasks as late starts. According to Parkinson's Law, the time to do a task fills up the span of time allocated to it. There are several reasons for why this is so. In many organizations, an early finish is dis-rewarded. For example, suppose that you finish your task early. How will you boss respond? In some organizations, next time around when you are asked to estimate how long it's going to take you to complete a task, your boss may assume you over estimated and take some time away from you.

Goldratt suggests safety that is removed from tasks on the critical path should be placed in a time buffer at the end of the project. This actually gives the project some likelihood that it will finish early. Safety takes from tasks that are off the critical path should be placed in a time buffer at the point where the path intersects the critical path. This will prevent the non-critical path from ever becoming critical. An illustration of how this happens follows:



The critical path here is B-E-F and is 24 months long. Suppose the project manager decides that all tasks are twice too long because half of the estimated time is safety. Clearly the critical path would be reduced to a length of just 12 months. The non-critical paths would also be reduced to half. The project would, in all likelihood, be finished 12 months early.

Selling and Negotiating Your Estimates

Steve McConnell (1996) points out that project managers must not only be good at estimating, they must be good at selling and negotiating their estimates. All too often well-conceived estimates are revised downward by project stakeholders who want to see the project completed sooner and at less cost. What this does is create too much stress and pressure, resulting in low-quality work that has to be redone.

USING ESTIMATES FROM THE ASSIGNED HUMAN RESOURCES IN IT PROJECTS

It is commonplace to allow the person or persons assigned to complete a particular work package to estimate how long it will take them to complete that work. This is a good practice because it is usually understood that they (the person(s) doing the estimating) will be required to perform to their estimate, in other words, to complete the work package by their estimated duration. In effect the estimate generates a commitment from that person to complete the work within the estimated duration. It is possible to intentionally overestimate the length of time required to complete a task. When the practice is employed, the estimator is putting **safety** into his or her estimate.

When it comes to using the assigned project professional to estimate how long it will take him or her to do the work, there are several concerns. First, that individual should be an expert and experienced in doing the work being estimated. If not, the estimate could be suspect, and probably underestimates the actual time required to complete the work. The reason for this latter state of affairs is because of the reputation that IT professionals have, namely that of underestimating how long the task or work package will take him to complete it. Especially inexperienced, project professionals have this tendency.

Seasoned project professionals have learned just the opposite. In fact, according to Goldratt, this group tends to overestimate the time required to complete the work package. The seasoned professional adds what Goldratt calls safety to the estimate, to make sure he has a near certain probability of completing the task in the time he estimated. Goldratt recommends cutting this persons' estimate in half, taking the additional time estimated and placing it along with all safety at the end of the project.

The pressure of getting product out in a timely manner must also be taken into consideration. While new products must reach their markets rapidly, shortened task durations, especially for tasks that require lots of creativity, will almost certainly result in an inferior product being brought to market. The advantage of being first to market has even been called into question in the face of a substantially better product that reaches the market later. In a book entitled simply **Slack**, the authors make a case for providing sufficient time to enable creativity to have its way.

Perhaps an expert system would be helpful in assisting the project manager in deciding how to manage all of the time estimates coming from project players. A rule-based system might consist of the following rules, among others:

IF ESTIMATER IS SEASONED AND IF THE WORK PACKAGE REQUIRES CREATIVITY ON THE PART OF THE ESTIMATOR, THEN LEAVE ESTIMATE AS IS.

IF ESTIMATER IS NOT SEASONED AND ESTIMATE APPEARS TO BE OPTIMISTIC, THEN INCREASE ESTIMATE BY 50%.

IF ESTIMATOR IS SEASONED AND ESTIMATOR ASSERTS 90% OR ABOVE CONFIDENCE HE WILL COMPLETE WORK WITHIN HIS ESTIMATE AND IF WORK PACKAGE DOES NOT REQUIRE SIGNIFICANT CREATIVITY, REDUCE ESTIMATE BY 50%.

Placing team members under too much schedule pressure is not necessarily a good thing. McConnell suggests that increased schedule pressure leads to skimpy development, which leads to major mistakes which lead to slower than expected deadline completions which lead to still more schedule pressure.

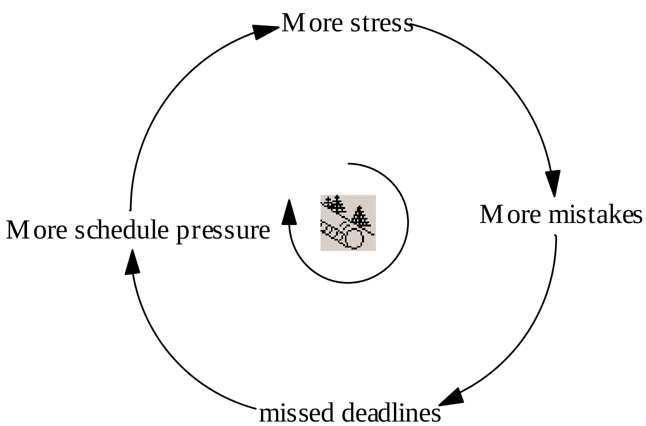


Figure 8.3 The Vicious Cycle Caused by Too Much Schedule Pressure

As Figure 8.3 suggests, putting team players under too much schedule pressure can lead to worsened project due-date performance relative to milestones. According to McConnell, we need to take time out to learn how to do our jobs better. Most managers, customers and end-users want to force as much productivity as they can for the expenditures being put forth, so they push to set optimistic milestones for project deliverables. Most software project schedules are, in the words of McConnell [], overly ambitious.

Furthermore, there is little awareness of the software estimation story or the real effects of overly optimistic scheduling. Software can't be reliably estimated in its early stages. It's logically impossible. Yet we let people force us into unrealistic estimates. Another contributor to the problem is the simple fact that developers are poor at negotiating their estimates. They may be pretty good estimators, but they are not good at selling those estimates to upper management and the customer. The end result is reduced estimates that may be overly optimistic.

GOLDRATT AND CRITICAL CHAIN PROJECT MANAGEMENT METHODOLOGY

In 1997, Goldratt published his book *Critical Chain* in which he delineated a new perspective on project management that has come to be known as Critical Chain Project Management (CCPM). Since then, much has been written about the construct that he created using the theory of constraints and statistical fluctuations. CCPM is presented as an alternative to the classical methods for project planning and control. It assumes that most estimations of task duration contain safety that is lost and wasted during execution. CCPM seeks to change project team behavior by encouraging 1) estimates that are only 50% probable of on-time completion, 2) un-penalized/un-rewarded reporting of late/early completions of activities, 3) the use of time buffers to compensate for absence of safety[1], and 4) elimination of student syndrome and multitasking. Companies such as Texas Instruments, Lucent Technologies, Honeywell,

[1] and slack, as late starts are used for all tasks not on the critical path.

and Harris Semiconductor complete projects in one half or less the time of previous or concurrent similar projects, using CCPM (Leach, 1999). However, not all recent discussions of CCPM have been positive.

CCPM does not prevent an estimate from containing hidden safety in it. As conventionally rendered, CCPM provides no formal method or technique for discerning when an estimate is or is not padded with safety. Leach (1999) suggests a technique for soliciting "low-risk" estimates first and then later, 50% probable estimates after teaching the estimators what a 50% probable estimate is and making sure that they understand they will not be penalized for going over or under their estimate.[2] However, there is still no guarantee that some estimators may choose to "social loaf" by padding their 50% probable estimates. Once some players/participants understand the system, they will find ways to defeat it.

Goldratt (1997) originally suggested cutting all "low-risk" estimates by 30% to 50%. By cutting all estimates by, say 50%, some durations may have less than a .5 probability of on-time completion while others may have a much greater probability of on-time completion. Training people to provide 50% probable estimates is no guarantee that they will, in fact, do that. Some estimators may have been through a 50% cut before of their estimates and therefore pad accordingly. Other estimators may provide a "low-risk" estimate that is four-times their 50% probable estimate so that they can then provide a "50% probable estimate" that is twice their actual 50% probable estimate. One advocate of CCPM suggests "In risky situations, and in subcontracts, it may be appropriate to include financial incentives …. such as paying for early delivery, penalties for late delivery, or paying for standby time." Clearly, such inducements will create incentives for persons to find ways to pad their estimates.

As previously mentioned, it is commonplace to have project players estimate durations for the tasks they are assigned. Given that we know a project player's estimate for an assigned task and his or her confidence in completing the work by the estimate, we can use the Z statistic and the standard normal table to calculate the time it should take the player to complete the task with a confidence/probability of .5. In this way, we can remove the safety the player has included in his estimate.

This work endeavors to improve upon some details contained within the Critical Chain Project Management (CCPM) developed by Goldratt and advanced by Leach (1999), Newbold (1998) and others. CCPM suffers from the following shortcomings. First, it is still possible for a project estimator to include safety in his estimate, in spite of the admonition, encouragement and training not to do so. Second, CCPM is totally unconcerned about project costs. Its only concern is that the project, feeding and resource buffers do not get completely consumed. If you have a project buffer that is 50% of the size of the project itself (which is commonplace), it could allow for a budget overrun of 50% over the critical chain budget. This might be unacceptable depending on whether stakeholders are focused on a critical chain budget and schedule or not. Acceptability is dependent on where the stakeholders expectations are set.

The use of the estimation questionnaire exhibited in Table 8.1 along with the following methodological detail will help to eliminate safety from estimates. The traditional CCPM estimation procedure works as follows. First, the estimator is asked to come up with a "low risk" estimate. Then, the estimator is asked, after some training, to come up with a 50% probable estimate. The estimator would be expected to complete his or her task within the 50% probable estimate with the understanding that there will be no penalties for early or late finishes (Leach, 1999). It is understood by the estimator that a portion of the difference between the low risk and the 50% probable estimates would be placed in a time buffer somewhere within the project network, perhaps at the end of the project in what has been appalled the "project buffer."

[2] Players would not be asked to do more work if they finished early as this would be viewed as a "penalty" for early completion.

Rather than just a simple request for a 50% probable estimate as suggested in the estimation methodology described above, the estimator might respond instead to the estimation questionnaire in Table 8.1.

Estimation Questionnaire

1. What is it that you've been asked to do, exactly, in 25 or less words?
2. How long will it most likely take you in hours of actual effort to do this?
3. Have you done anything like this before?
4. How many times? Have you taken this into consideration in your estimates?
5. How confident are you that you will finish your task in the time you've estimated in question 2 above? State your confidence as a %.
6. Will you be multitasking (doing several tasks at once) during this period?
7. If yes to question 6 above, how many other tasks will you do in addition to the current one for which you've been asked to estimate duration?
8. If yes to question 6 above, what percentage of your time will you be spending on the task you described in question 1 above?
9. In terms of elapsed time how long will it take you to complete this task?
10. What is your optimistic completion time (in hours of actual effort)?
11. What is your pessimistic completion time (in hours of actual effort)?

Table 8.1 An Estimation Questionnaire

Beginning with the responses supplied to the questionnaire above, it would be possible to calculate an updated, consistent estimate for each task time. From question 2, we are able to derive a most likely completion time for each assigned task. From questions 10 and 11, we are able to obtain optimistic and pessimistic completion times. We can then calculate a mean completion time using: mean = $(a + 4m + b)/6$ where a = optimistic completion time; m = most likely completion time; b = pessimistic completion time. A standard deviation can be calculated using: $\sigma = (b - a)/6$.

Using the response given in question 5 of the questionnaire, we can calculate an updated task time t that is 50% probable: $P\{(\text{mean} - t)/\sigma\}$ = player's response to question 5.

Suppose that a project player's responses to questions 2, 10, 11 and 5 are: seven weeks, four weeks, twelve weeks and 85% confident. Then the mean is calculated to be 7.33 weeks and the standard deviation 1.33. Using a standard normal table, the $P\{(7.33 - t)/1.33\} = .85$ can be solved for t. The Z-value corresponding to a normal probability of .85 is 1.03 (.5 must be subtracted from .85 to obtain the "lookup" number of .35 to which 1.03 corresponds). Thus, $(7.33 - t)/1.33 = 1.03$ leads to a value for t of 5.96. The project player might then be told that he has six weeks to complete the task. If a player reports that he is 100% confident, then $P\{(7.33 - t)/1.33\} = 1$. The Z-value corresponding to a normal probability of 1 is 3.05 (remembering to subtract .5 from 1). The value for t is then calculated to be 3.27 and the player might then be told he must finish the task in 3.3 weeks. In this way, all time estimates can be "normalized" to a completion probability of .5. Of course, the project manager's final estimate for the player would be tempered by the players' answers to the other questions in the questionnaire as well and by the project manager's subjective assessment of the time required by the particular project player to complete the task.

We must concede that in performing this calculation, we have assumed a normal approximation to what was first asserted to be a beta distribution. Because we do not have standard beta tables from which Z values can be determined, such an approximation was necessary. The approximation may not be valid in cases where the beta distribution is very skewed.

Pursuant to standard CCPM conventions, resource flags will be used to inform resources that their tasks will begin shortly. If the project player is performing a task that is on the critical path, then pursuant to CCPM discipline, the player would be required to "drop everything and do nothing but work on that task until it is complete," once the time has arrived for the player to do the critical task. This would mean no multitasking, no student syndrome, nothing but total 100% focus on that critical task until it is complete.

Detail Steps to Use of Questionnaire and Reverse Calculation of a Safety-free Estimate

Steps to use of the questionnaire and reverse calculation of a safety-free estimate using a standard normal table follow. At all times the focus is on the critical path, as Goldratt suggested, and exploiting that.

1. Extract a low-risk "comfortable" estimate from each player for each task.
2. Utilize the low-risk estimates in a predefined network chart & Gantt chart to determine total project duration and cost; publish and distribute total duration and cost.[3]
3. Train players in the basic concepts of CCPM and PERT
4. Submit the questionnaire in Table 8.1 to each player and use the data so obtained to determine "updated" estimates for each task.
5. Negotiate updated estimates with each player under the following "umbrella" of understanding:
    a) players will be trained in CCPM concepts;
    b) a major portion of the players' lost time will be re-allocated to a "time buffer;"
    c) estimates are roughly 50% probable of on-time completion;
    d) players will not be penalized for late or early completion of their assigned tasks;
    e) all players are losing the safety they placed in their estimates;
    f) safety is being removed and placed at the end because, embedded within projects, it gets wasted (illustrations of why this is so would be provided in the training session, step 3.).
6. Use the updated estimates to determine new Gantt & network charts.
7. Determine duration of all time buffers using standard conventions within CCPM.
8. Set all non-critical paths for late start but with time buffers included at the point where the non-critical paths intersect the critical path.
9. Begin execution with the understanding that there will be total focus (no student syndrome or multitasking) especially for tasks on the critical path.
10. Utilize resource flags (notification events) to notify players when their deliverable will "arrive" so they can be ready to "drop everything" and do nothing but their task until their contribution is finished; update resource flags as necessary[4].
11. Utilize a modified EVM/EVA tracking methodology that is sensitive to progress on the critical path, as discussed in the early sections of this paper. Once the project is at least 40% complete, after each milestone, calculate EAC and ETAC as well as the PERT-based probability of completion by a specified date and budget amount.

Leadership on the part of the project manager will be required to get the project players to buy-in to the reduced duration estimates and why these reduced durations will benefit the entire project and team. Clearly, a computerized support system would facilitate the implementation of this methodology.

COST ESTIMATING

A cost estimate of the project needs to be developed because project stakeholders, particularly the project sponsor, must know how much the project will cost in anticipation of a go/no go decision. The cost estimate should be an approximation of the costs of the resources needed to complete project activities. The customer may want an overall cost estimate for the project, or they may request a detailed breakdown of various costs. An example of this breakdown might be to estimate costs for labor, materials, subcontractors and consultants, equipment and facilities rental, and travel.

[3] The steps up to this point are identical to conventional project management; the point of what has transpired so far has been to establish benchmarks for total duration and cost against which the CCPM can be compared.
[4] Resource flags should notify successor resources five business days before the starting time arrives.

After the project cost estimate is determined, it is then allocated to the various tasks. The inputs for this process include the work breakdown structure, resource requirements, resource usage rates, activity duration estimates, and historical information. The work breakdown structure will be used as an organization tool. The cost estimates will be organized to make sure that all identified work has been estimated. The resource requirements describe what types of resources are required and the quantities needed of each resource. The resource rates are the unit rates for each resource. Historical information can be gathered from project files, commercial cost estimating databases, and project team knowledge.

There are several techniques for cost estimating. Four of the main techniques include analogous estimating, parametric modeling, bottom-up estimating, and computerized tools.

Analogous estimating, or top-down estimating, uses the actual cost of a previous project with similar circumstances and characteristics as a basis for estimating the cost of the current project. This technique is also used in estimating task duration. This technique is generally less costly than other techniques; however, it is also less accurate because projects are not exactly the same.

Parametric modeling uses project characteristics in a mathematical model to predict project costs. Parametric modeling is most reliable when accurate historical information is used and when the parameters used in the model are readily quantifiable. It is also most reliable when the model is scalable and can be used for large or small projects.

Bottom-up estimating takes the costs of the individual work packages and "rolls them up" (adds them all together) to get a project total. Since smaller work items can increase both cost and accuracy, the project team needs to weigh the advantages and disadvantages of this method very thoroughly.

Computerized tools can also be very helpful in assisting with cost estimating. Project management software and spreadsheets are widely used for this purpose. This method can reduce time as well as simplify the process of cost estimation.

Cost Realism

The term "cost realism" is widely used today in the information technology project management realm. No one expects a cost estimate to exactly predict what a hardware or software product or a service will cost. We see that cost realism does not have to do with finding a precise cost estimate for these things. Cost realism is about the system of logic, the assumptions about the future, and the reasonableness of the historical basis of the estimate. The realities behind cost realism are the things that make up the foundation of an estimate. A realistic cost estimate should not underestimate the actual cost; rather, the estimate should be 75% to 10% over the actual cost—the more preliminary and less-studied the estimate, the more it should over-state the actual cost.

IT project managers seek to discover the mystery of cost realism. They understand that if they don't continually work towards the goal of cost realism, they will soon be replaced by project managers who grasp this phenomenon. The closer a project manager comes to accurately estimating project costs, the closer he or she comes to delivering a successful project. There is no question that cost estimating is a difficult task and one that each IT project manager is faced with. It is also known that the cost estimating process will continue to confront future unknowns. These unknowns are what make cost estimating one of the most difficult tasks. But sound assumptions, high quality historical data, and unbiased analysts and estimators will improve the process for all, and will allow corporate and program management to make the best selections of projects for their organizations to accept. Project managers from the top consulting firms continually search for the best methodology to use for estimating software project costs.

There are many methodologies that are used to achieve cost realism when estimating IT project costs. Each method is unique on its own accord. Each method is only as good as its conductor. A software developer can have the greatest and most innovative programming skills, but he or she would still fail as a cost estimator if the estimation method employed were not correctly carried out.

Accurately estimating the resources and time needed for a software development project is essential for the survival of the project. In many cases, the resources and time actually used are much more than the initial planning estimates. An approach for estimating the resources and schedule needed for software development is the use of a software cost and schedule model that calculates the resources and time needed as a function of some other software parameters (such as the size of the program to be developed).

SOFTWARE COST ESTIMATION METHODS

There are many methods that IT project managers use to come up with a realistic software cost estimate. Two types of parametric cost estimating methods are the COCOMO method and the PUTNAM method. The COCOMO (Constructive Cost Model) is developed by Barry Boehm; it uses a regression formula to estimate effort, cost, and schedule for software projects. The PUTNAM method, created by Lawrence Putnam, describes the time and effort required to finish a specified size software project. Many of the larger consulting companies such as, PriceWaterhouseCoopers, and Accenture are refraining from using these models. Companies are not heavily using these models because they are based on knowing, or being able to accurately estimate the number of lines of code that are needed to develop a software package. These companies state that there is often great uncertainty in the software size of the projects they are developing.

Another software cost estimating approach is called Function Point Analysis (FPA). FPA is used to estimate the effort required for the software development and the size of the software to be developed. In FPA, you determine the number and complexity of inputs, outputs, user queries, files, and external interfaces of the software to be developed. The sum of these numbers, weighted according to the complexity of each, is the number of function points in the system. Using data from past projects, it is possible to estimate the size of the software needed to implement these function points (typically about 100 source language statements are needed for each function point) and the labor needed to develop the software (typically about 1 to 5 function points per person month). Consulting companies tend to prefer this method to the previous methods that were mentioned.

Dick Lefkon believes in yet another IT project cost estimating method. He believes that "there is no better tool for predicting project costs than experience." Lefkon believes that until you actually roll up your sleeves and get started on a project, all estimating tools suffer from the same lack of basis in the facts of your specific application. Lefkon's methodology is similar to the Function Point Analysis method in some aspects. His eight-step process is listed below.

1. Divide the software project into as many individual steps/tasks/modules as possible.
2. Predict the level of effort required to complete each task and multiply that prediction by 2.0.
3. Add up the numbers and multiply by 2.0 again to account for testing and debugging.
4. Take the total and multiply by 1.25 to account for meetings, administration, and paperwork.
5. Multiply this level of effort by your company's "magic number" for labor costs.
6. Present this to management as a range. Take the cost as predicted above and present the range as –10 percent and +25 percent.
7. Stand your ground and remind management that you did not arbitrarily come up with these numbers and they cannot be adjusted arbitrarily. You may have to suggest reducing scope and cost if management does not agree with your estimate.
8. Revise your project budget as you undertake and complete the project.

Lefkon states, "It's a really good idea to get management used to the idea early in the project that projects are dynamic and budgets have to be flexible – both up and down." He believes that the worst mistake that a project manager can make with regard to a software project that has been prepared in optimistic ignorance, is for that project estimate to remain unchanged in the face of actual experience.

Another approach to software project cost estimating is in disagreement with Lefkon's methodology in some aspects. McNeil states that a project manager should deal with cost estimating before any actual steps toward production are taken. He explains that a well thought out upfront estimate would allow a project manager to achieve an accurate project cost estimate. Like Lefkon, McNeil believes that a cost estimate should be based on all known and measurable factors as well as on past performances. McNeil speaks of the importance of the learning curve and a cost estimator. He feels that an estimator can more accurately project future costs from past performances by being further up on the learning curve.

Representatives from some of the top consulting firms would agree that an evaluation of past performance is necessary in estimating software project costs. It seems that it is also necessary to estimate the costs of a software project before the project work begins. Representatives from one system integration firm agree with Lefkon's method in respect to taking all things into account before providing a cost estimate that is too optimistic. The system integration firm gains or loses clients based on the accuracy of their project cost estimates.

It is smart not to depend solely on one cost estimating method or model. There are many models and methodologies that have proved to be effective over the years. IT project managers from the top consulting firms definitely have their preferences for software project cost estimating, but in most cases they use a couple of different software estimating models and compare the results from each model. They feel that this enables them to provide more realistic software cost estimates and brings them closer to the goal of cost realism.

The art of achieving cost realism in estimating software project budgets is illusive. Estimated budgets for software projects continue to be underestimated and thousands of software projects over-run their budgets every year. We hope that project experience will change this trend, but new technologies emerge yearly making it increasingly difficult to accurately estimate the cost of a software project. Regardless of these unfortunate happenings, any good project manager sets his or her sites on the goal of cost realism and the hope that he or she will deliver a successful project. We acknowledge that one supreme, widely-accepted practice for cost estimating does not exist. It is impossible to estimate with 100% accuracy, but the top consulting firms continue to work at solving the mystery and focusing on accurate and precise estimating.

SPECIAL PROBLEMS IN COST ESTIMATION/BUDGETING

We have learned that the approach to creating an accurate cost estimate for a project is multi-faceted. If a company is to win contracts from an outside customer, they must be able to guarantee three things:

1) On-time delivery of a quality product or service that meets or exceeds the customer's expectations.
2) Availability to perform follow-up work on the product or work on the next phase in the project (if project consists of more than one phase).
3) Flexibility to adapt to changing customer needs.

In the extremely competitive business world of today, upper-level and corporate managers are always trying to cut costs on the projects their company is involved with. This practice often leads to reduced product quality or schedule slippage. How organizations improve their bottom line without sacrificing quality or delaying the product is one of the challenging questions for today's project managers. Later in this chapter, method for reducing project duration without compromise quality will be presented.

The 10 Percent Solution

In his book, Harold Kerzner writes, "For the project manager, the worst situation is when senior management arbitrarily employs 'the 10 percent solution,' which is a budgetary reduction of 10 percent for each and every project, especially those that have already begun. The 10 percent solution is used to 'create' funds for additional activities for which budgets are nonexistent. The 10 percent solution very rarely succeeds." Too often, upper-level executives believe that cost and quality are linearly related: if the project budget is slashed by 10 percent, the quality of the product will decrease by 10 percent. However, this is simply untrue. When you factor in time to the equation, the original 10 percent reduction in cost results in the 10 percent reduction in quality **and** exponentially delays the product.

So what are some alternatives to this 10 percent problem? Generally, there are two. The first solution is for the 10 percent solution to be used, but **only** after an impact study has been performed, which would allow executives to see the impact on time, cost, and performance constraints. The second solution, which is the preferred option, is for the executive committee to cancel or descope selected projects. As it is impossible to reduce a project's budget without reducing its scope, canceling or delaying a project is a viable option. All projects should not have to suffer due to budget reductions on one.

Risk Involvement

Every project contains risk. Examples include: suppliers missing delivery deadlines, customers requesting changes at the last possible minute, or workers going on strike. Risks tend to increase the chances that the project goals of time, cost, and performance will not be met.

However, with the aid of tools such as decision support systems, expected value measures, and trend analysis/projections, risk can be minimized. Independent reviews or audits performed can also be particularly useful. The best strategy of dealing with risk relies on six steps:

1. Identify the risk
2. Quantify the risk
3. Prioritize the risk
4. Create a strategy to manage the risk
5. Review by the project sponsor or executive committee
6. Take action

REDUCING PROJECT DURATION

One of the challenges of managing projects is estimating costs and schedules. Earlier in this chapter and previous chapter, we have described the processes and techniques for estimating task duration and cost to create a realistic schedule and assign resources to that schedule. A project manager might consider applying some techniques discussed in this section to reduce the total project duration. Shortening a project schedule, however, may entail an additional cost that a project manager has to take into account as a cost-time tradeoff problem—whether it is worth expediting the project. Some of the reasons to accelerate a project are as follows:

1) A project sponsor or customer might simply ask if a project team can get the project done earlier than scheduled. Project sponsors and customers are highly influential persons who can make a project succeed or fail. When appropriate, therefore, a project manager and project team should consider their requests.
2) In a world of intense competition, time to market is also important. A company can gain competitive advantage of getting products and services to the market sooner. When the company is first to market, there is a high chance to capture more market share which, in turn, yields more profits.
3) When a project runs behind schedule, a project manager may take action to get a project back on schedule. As mentioned before, reducing the duration of a project may incur additional costs. A

project manager might have to take a look at a project priority to see if the time is a first priority and compare the incremental costs of getting the project back on schedule with the results of being late.

4) At times, contractual agreements set forth penalties for delivering a project late or rewarding for completing a project earlier than planned. In a first case, when a project has slipped behind schedule, a project manager might have to compare the penalty costs of being late with the additional costs of getting the project back on schedule. In a latter case, rewards can provide incentives for accelerating project completion that usually benefit for both the project contractor and owner.
5) If a project can finish early, a project team and equipment can be released from a project and be reassigned to another project. Again, a project manager has to compare the cost of shortening the project with the costs of not freeing up people and equipment.

According to a PMI-defined process of schedule development, *schedule compression* deals with reducing the duration of the project without changing the project scope or objective. A project manager can use two different methods, crashing or fast-tracking, to compress a project schedule. These two methods, discussed next, can be applied independently or together to selected activities or the entire project.

Crashing

Crashing is an approach for deciding cost-time tradeoffs that provide the greatest amount of schedule compression for the least additional costs. By adding resources to an activity, it can be completed sooner, albeit at additional cost. The purpose is to reduce project duration. Shortening activities on the critical path can make a project finish sooner while reducing non-critical activities does not.

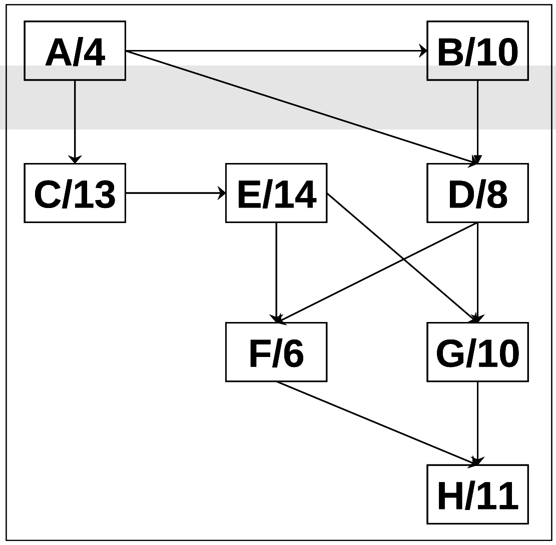
While it is obvious that the advantage of crashing is accelerating the time it takes to complete a project, it often adds total project costs. However, in some circumstances, crashing is not applicable. There are some activities that have fixed duration and will not complete faster when adding more resources. For example, the task time "letting the paint dry" cannot be reduced by adding extra resources. A heuristic for crashing follows.

A Heuristic for Crashing

1) Enumerate all of the paths in a table showing duration of each
2) Pick an activity on the CP (Critical Path) with the lowest crash cost per day and crash it to its minimum duration if possible, but not so far back that the critical path is no longer critical and not so far back that you exceed the budget
3) Show the effects on each path's duration in the table
4) Reduce the budget by the amount spent on the crashing step
5) Continue with steps two through four until all of the budget is used up.

NEVER CRASH THE CRITICAL PATH BEYOND THE POINT WHERE IT IS NO LONGER CRITICAL (THAT IS, SOME OTHER PATH HAS BECOME CRITICAL).

The best time to do crashing is toward the end of the second stage—Planning and Budgeting, in the finalization of the project plan and budget. An example of crashing follows.



For the network diagram above, find the best crashing strategy considering that you have $1,300 in crashing budget and the following:

Task	Crash cost/day	Minimum duration
A	$100	2
B	$200	6
C	$300	8
D	$400	6
E	$400	9
F	$200	4
G	$100	8
H	$500	10

PATHS	LENGTH	Crash A 2 days	Crash G 2 days	Crash C 3 days
A—B—D—G—H	43	41	39	39
A—D—G—H	33	31	29	29
A—D—F—H	29	27	27	27
A—C—E—G—H	52	50	48	45
A—C—E—F—H	48	46	46	43
A—B—D—F—H	39	37	37	3
BUDGET LEFT	$1,300	$1,100	$900	$0

The entire $1,300 budget is used up. A—C—E—G—H remains as the critical path. Seven days are reduced from the total project duration, which was 52 days and now is 45 days.

It always makes sense to crash tasks so long as the crash costs per day are less than the indirect costs per day. When such is the case, crashing not only saves time, it also saves money.

The methods for adding more resources to a project so as to finish it sooner include:

1) Assigning additional labor or equipment to activities. This is our definition of 'crashing.' Obviously, adding more people or other facility to the project increases the amount of work that can be finished in parallel, at the same time. This is true when the project requires intensive labor and minimal communication between workers. Consider, for example, a ditch-digging project requiring minimal training and communication overhead. The ditch can be dug with 1 person in 100 months, by 10 people in ten months or by 100 people in one month. However, most projects require interaction among team members. Thus, adding more people onto a project increases the cost of communication and coordination among the team members. It might also increase cost of training new people and additional time and process to make team members get to know each other so that they can work well together. This is especially true of software projects as Brooks' law states that 'adding manpower to a late software project makes it later.' And, we might add, a whole lot more expensive. In software projects the interchange of men and months is not possible as it was in the ditch-digging project. In the ditch-digging project there was not training, communication or interaction overhead, so the interchange between men and months was linear. In a software project, such is not the case. As additional manpower is added, the total man-months required to complete the project goes up because of the additional training, communication and interaction that is required.
2) Scheduling current employees to work overtime instead of adding more people to a project. This option can avoid the cost of training, communication and coordination that occurs when new people are assigned to the team. Moreover, the current workers already know about the project and know what they have to do in order to get the work done. However, sustained overtime work might cause project players to burnout which, in turn, reduces their productivity.
3) Reducing workload for workers if they have been assigned to work on multiple projects so that they can devote more time to the desired project. Some workers may be taken off one or more projects, so they can concentrate on the project of greatest interest. This is another way to add manpower to the project. A project manager should prioritize projects so that the best people can be placed into the most important project.
4) Outsourcing project work or subcontract activities to outside expertise. This option is practical when there are not sufficient resources to complete the project as scheduled or when a project team does not possess specialized skills to complete the project or other contractors might have expertise in the work that can get the work done quicker and cheaper.

Fast-tracking

Fast-tracking is one of the two methods described in PMBOK on how to compress schedule. It involves rearranging the relationship of activities so that critical activities are done in parallel or with some overlap rather than in sequence. Assuming that there are sufficient resources for more than one activity to start simultaneously, fast-tracking can be implemented by changing the relation of activities from a finish-to-start (FS) relationship to a start-to-start (SS) relationship. For example, instead of waiting for the final design to be approved, the programmers can start coding the system as soon as major specification requirements have been determined.

Like crashing, fast-tracking can shorten the time it takes to finish a project. While fast-tracking can speed things up, it can result in rework and increase risk because this approach requires work to be done without completed detailed information. Fast-tracking relies on overlap tasks that have ordinarily been done in sequence. The overlap increases risks to the project. However, if there is well coordination among a project team, this method can reduce project duration significantly.

In addition to crashing and fast-tracking, there are several good ways to reduce the project duration such as reuse, parallel or concurrent activity, doing it right the first time, do it twice—fast and correctly, avoid changes in project scope and requirements, removing safety, elimination of multitasking, establish a core project team, re-estimate the project, and reduce project scope. A brief discussion of these options is presented here.

Reuse

A reuse concept can reduce the amount of time for a project team to create a new deliverable if it is the thing that has already done before and can be applied to the project. When a project team can make use of some material from a previous project, a project might require significantly less labor, time and effort to finish the project. Therefore, reuse not only shortens project duration but may also reduce the size of a project team, which translates into cost savings. In a context of information technology project, for example, a project team might be able to use part or all of object code, source code, documentation, and specifications that were written for the previous project with the current project instead of starting to create everything from scratch.

Parallel or Concurrent Activity

Another method of accelerating a project is looking for an opportunity to perform tasks in parallel or concurrently rather than developing a long series of sequential tasks, as briefly introduced in Chapter 1. In order to shorten project duration, a project manager identifies tasks on the critical path that a project team can start to work on concurrently. For example, task B can start when task A is 50 percent complete instead of waiting until task A is 100 percent complete. By implementing this method, a critical path might be altered requiring a project manager to recalculate the critical path and the overall network.

Do It Right the First Time

Philip Crosby, one of the guru's of the quality revolution, promoted the phrases 'zero defects' and 'right first time' in his book *Quality is Free*. His principle of zero defects does not mean mistakes never happen. Rather, there is little tolerance for errors/defects built into a product and workers endeavor to "do it right first time." There is no defect cost adding to the product when doing it right the first time. On the other hand, costs such as rework, test, warranty, inspection incur when doing things wrong. Moreover, additional time is required to fix the errors/defects and the project may be delayed. A project manager can apply this concept when planning and executing a project. Once the project is executing, the costs to fix defects are more expensive and consume more time. The further down the lifecycle that the defects are discovered the more expensive and time-consuming they are to fix. Preventing errors or unpleasant events from happening to the project can help the project stay on track.

Lean Concepts

The philosophy of 'lean' was originally known as the Toyota production system and Toyota was the originator of it. The lean philosophy is an approach to create the most value while using the fewest resources. Value is defined by a customer; therefore, in order to implement a successful project, a project manager and project team have to know their customers, try to understand what they want, and look at things from the customers' perspectives. This concept can be used when defining and conceptualizing a project plan and requirements. A project requirement or statement of work should contain only work that adds value to the customers. By eliminating or minimizing things that do not add value to a project, a project can finish sooner and cheaper.

Do It Twice—Fast and Correctly

This method seems to contradict the concept of doing it right the first time as stated earlier. Instead, this alternative advises that if you are in a hurry, try building a quick and dirty short-term solution or prototype. Learn from that prototype and then go back and build it the right way based on the learning. Using this method, a project can finish in a short-term period and a customer gets a quality product fast. Prototyping tools are used to get a solution up and running quickly. It is recommended that this method should be used when delays of the project are not acceptable or when the time to market for a product is vital. Another advantage of this alternative is that a project team can get feedback from the first product so that the second product can be improved.

Avoid Changes in Project Scope or Requirements

A project can quickly and easily fall behind schedule because there are so many changes in project scope or requirements after a project has started execution. Changes in scope or requirements most likely come from customers who might not know exactly what they want at the beginning or they might come up with a new idea to improve the product when the time goes by. Changes can delay the project and cost more to the project. If possible, a project manager should discuss with customers what specific changes they really want to the stated requirements and what the additional cost and duration of each proposed change will be. Often a Change Control Board is used to assist with making these changes. Then, a project manager should make an agreement with customers to freeze requirements during the execution and control stage.

Another source of changes might come from a poor planning project at the beginning. While executing, a scope or requirement of the project has to be adjusted to be consistent with the actual requirement. In some cases undiscovered work is found after the planning and budgeting stage. In this case, changes might not be avoided. Good project definition and planning should be implemented before a project is executed. Although limiting changes in project scope or requirements might not guarantee that the project will finish on time and on budget or will not reduce the planned project duration directly, it helps keep the project running as originally scheduled.

Removing Safety

Often, when a project manager asks project team members to estimate the time to finish their work, project team members might pad their estimates of the time to complete their activities in order to protect themselves from delivering the project late. This additional time is called 'safety.' Embedded safety adds additional time and cost to a project but does not reduce its risk of late, over-budget completion. According to Goldratt, this safety should be removed from all of the estimates by using the method suggested earlier.

Elimination of Multitasking

As discussed earlier, multitasking occurs when people have been assigned to more than one project. Project team members have to allocate their time to work on multiple projects instead of concentrating only one task at a time. Switching from one activity in one project to another task in a different project incurs interrupt time and switching cost. For instance, as a student you enroll in many courses in one semester and every class has an assignment. It is more efficient to finish one assignment entirely for one subject before moving to work on an assignment for another course. Every time you switch to a different assignment, you have to recall where you left off and figure out what to do next. Reducing interrupt time to minimum can increase your productive time of working on the project. Thus, the elimination of multitasking is another way to shorten the project duration.

Establish a Core Project Team that is focused

A core project team can be established when a project manager and project team are assigned to work full-time on a specific project. Projects tend to get done sooner when workers devote all of their time on a project and share vision, goal, and responsibility. A project team stays focus only on the project and avoids costs of multitasking as stated earlier when a core project team has been established. Ideally, a dedicated core project team can help speed up the completion of a project through staying focused. Goldratt is an advocate of this idea. He recommends that project players completing tasks on the critical path be informed two-five days before that their task is coming. This allows the project player to 'get ready.' When the deliverable that player is to work on arrives, he or she is expected to 'drop everything' and do nothing but finish that task as quickly as possible, informing the successor two-five days in advance, when the deliverable will be ready.

Reducing Project Scope

When resources are constrained and the project schedule is not able to meet a deadline, reducing project scope is another way to accelerate a project. While this alternative can lead to savings in both time and budget, it can reduce the value of the project. Scaling-down project scope means reducing the features and functionality of the project. In this case, a project manager has to discuss with a customer whether this option is acceptable, whether time and/or budget takes priority over scope.

Tools and techniques described in this section can be used to accelerate project completion without compromise quality. However, reducing the time of a critical activity in a project may result in a higher cost. A project manager has to weigh whether time or money has higher priority. It is accepted that as long as the daily costs of schedule compression are less than the indirect daily costs, the firm is saving money and time by compressing the schedule.

SUMMARY AND CONCLUSION

In this chapter, the different methods of estimating schedules, tasks, and costs associated with a project were discussed. The task estimation process is best approached as a system, with separate inputs, tools, and outputs. When the task estimation period is complete, the project team will have a project schedule, which will consist of all of the tasks related to completion of the project.

The next phase involves costing out the activities. The four main techniques used in cost estimation are analogous estimating, parametric modeling, bottom-up estimating, and use of computerized tools. It is very important that the cost estimate for the project is as accurate as possible, as this can potentially determine whether or not a firm wins a contract, and whether they can deliver the final product on budget. We also discussed some of the common software cost estimation methods (COCOMO, PUTNAM, FPA, etc.) and some common problems in estimation, such as the 10 percent solution and risk involvement.

EXERCISES

1. Define what is meant by:
   - Analogous Estimation
   - Estimating Manual
   - Expert Judgment Estimation
   - Function Point Analysis
   - Historical Data Estimation
   - Monte Carlo Analysis
   - Safety
   - Simulation Estimation
   - Student Syndrome
   - Multitasking

2. During initial pricing activities, one of the functional managers discovers that the work breakdown structure requires costing data at a level that is not normally made, and will undoubtedly incur additional costs. How should you, as a program manager, respond to this situation? What are your alternatives?

3. How does a project manager price out a job in which the specifications are not prepared until the job is half over?

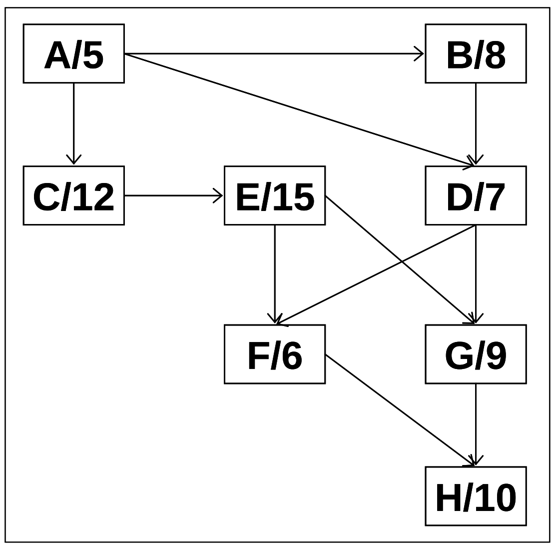
4. Crash the network diagram in the table below assuming you have $1,200 to spend and that activities cannot be shorter than their minimum durations. Note that activity A can only be crashed 3 days. Try to achieve the greatest reduction in project duration for the least amount spent.

ACTIVITY	DURATION	MINIMUM	PRECEDENT	CRASH

	(DAYS)	DURATION	ACTIVITY	COST/DAY
A	11	3	-	$200
B	9	6	A	$300
C	16	8	A	$200
D	13	6	B, C	$100
E	12	9	B, C	$200
F	7	4	D, E	$300
G	10	8	D, E, F	$400

5. For the network diagram below, find the best crashing strategy considering that you have $1,200 in crashing budget and the following:

TASK	CRASH COST/DAY	MINIMUM DURATION
A	$100	3
B	$200	6
C	$300	8
D	$400	6
E	$300	9
F	$200	4
G	$100	8
H	$500	10

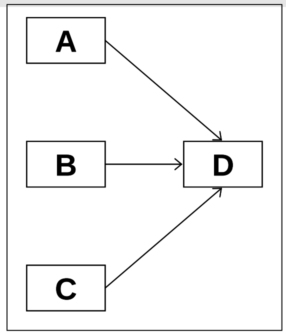


6. Draw the network diagram given the information in the table below.

ACTIVITY	DURATION (DAYS)	MINIMUM DURATION	PRECEDENT ACTIVITY	CRASH COST/DAY
A	9	3	-	$400
B	8	6	A	$300
C	12	8	A	$200
D	12	6	B, C	$100
E	11	9	B, C	$200
F	6	4	D, E	$300
G	9	8	D, E, F	$400

7. Crash the network diagram in Problem 6 above assuming you have $1,100 to spend and that activities cannot be shorter than their minimum durations. Note that activity A can only be crashed 3 days. Try to achieve the greatest reduction in project duration for the least amount spent.

8. In the project network below, suppose activities A, B, C, and D can be crashed at the following additional costs:



ACTIVITY	COST/DAY	CRASH COST/DAY	MINIMUM DURATION	CRASH DURATION
A	$800	$1,600/day	60	45 days
B	$1,000	$2,000/day	90	60 days
C	$500	$1,000/day	120	70 days
D	$1,000	$2,000/day	75	50 days

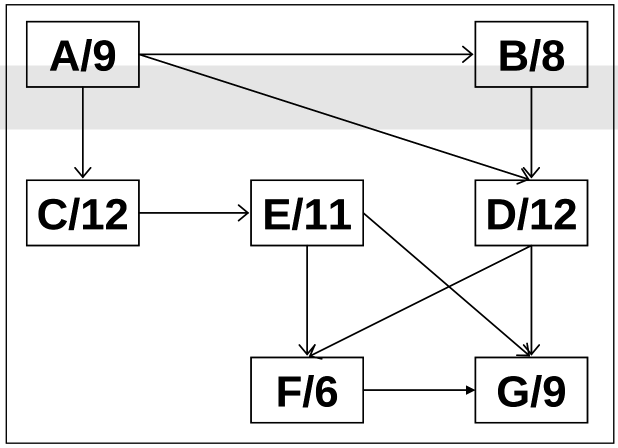
Suppose that you have a crash budget of $100,000. You can only crash entire activities. It is not possible to crash portions of activities; that is, activity A if it is crashed at all, must be crashed to its minimum of 45 days at a cost of 15 days*$1,600/day or $24,000. Your decision has to be made before the project begins, so progress data cannot be used.

9. Crash the network described by the table below, assuming you have $1,100 to spend and that activities cannot be shorter than their minimum durations. Note that activity A can only be crashed 3 days, from 5 down to 2 days. Try to achieve the greatest reduction in project duration for the least amount spent.

ACTIVITY	DURATION (DAYS)	MINIMUM DURATION	PRECEDENT ACTIVITY	CRASH COST/DAY
A	5	2	-	$400
B	6	3	-	$200
C	8	5	A, B	$200
D	7	4	A, B	$300
E	9	7	B, C	$100
F	8	6	C, D	$300
G	10	8	E, F	$100

10. For the network diagram below, find the best crashing strategy considering that you have $1,100 in crashing budget and the following:

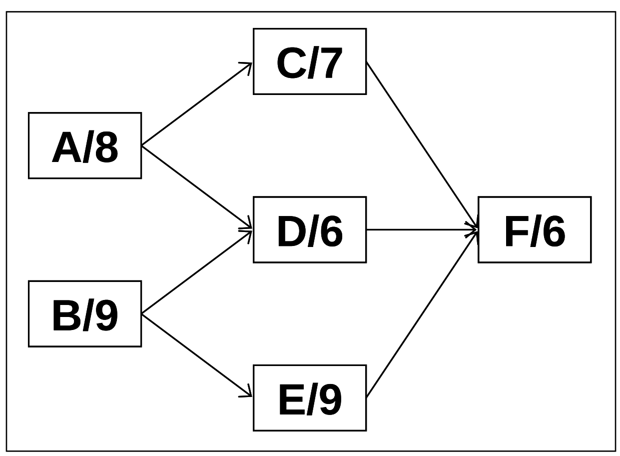
TASK	CRASH COST/DAY	MINIMUM DURATION
A	$100	3
B	$200	6
C	$300	8
D	$400	6
E	$300	9
F	$200	4
G	$100	8



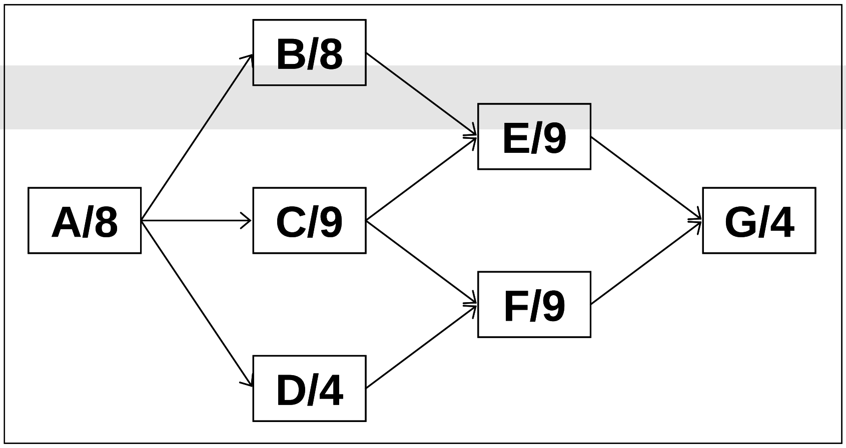
11. Crash the network described in the table below assuming you have $1,000 to spend and that activities cannot be shorter than their minimum durations. Note that activity A can only be crashed 4 days, from 6 down to 2 days. Try to achieve the greatest reduction in project duration for the least amount spent.

ACTIVITY	DURATION (DAYS)	MINIMUM DURATION	PRECEDENT ACTIVITY	CRASH COST/DAY
A	6	2	-	$400
B	4	3	-	$200
C	8	5	A, B	$200
D	9	4	A, B	$100
E	3	2	B, C	$100
F	5	3	C, D	$300
G	6	4	E, F	$400

12. For the following network, assume that half of the time in each estimate is safety time. Remove the safety from each node, and place it in appropriate buffers at the end of each non-critical path or at the end of the entire project itself, as suggested by Goldratt.



13. For the following network, assume that half of the time in each estimate is safety time. Remove the safety from each node, and place half of it in appropriate buffers at the end of each non-critical path or at the end of the entire project itself, as suggested by Goldratt.



14. In problems 12 and 13 above, discuss whether an early or late start is most appropriate for tasks off the critical path, once an intervening time buffer has been created for them.

REFERENCES

Anbari, Frank. T., (2002). *Quantitative Methods for Project Management*, Second Edition, New York, NY: International Institute for Learning.

Anbari, Frank T., (2003) "Earned Value Project Management Method and Extensions," Project Management Journal, Vol. 34, No. 4, pp. 12-23.

Austin, Larry and James R. Burns, (1985) *Management Science: An Aid for Managerial Decision Making*, New York: Macmillan.

Brooks, F. P. Jr. (1995). *The Mythical Man Month*. Reading, MA: Addison-Wesley.

Chang, Yih-Long, (2001) *PERT/CPM – A Module within WINQSB*, distributed by Wiley.

Crosby, P. B. (1979). *Quality is Free*. New York, NY: McGraw-Hill.

Goldratt, E. (1997) *Critical Chain*, Boston, MA: North River Press.

Gray, C. F., & Larson, E. W. (2008). *Project Management: The Managerial Process*, 4th Edition. New York, NY: McGraw-Hill/Irwin.

Kerzner, Harold, (1984) *Project Management: A Systems Approach to Planning, Scheduling and Controlling,* New York: Van Nostrand Reinhold.

Kerzner, Harold, (2003) *Project Management: A Systems Approach to Planning, Scheduling and Controlling* (8^{th} Ed.). New York, NY: John Wiley, pp. 835-867.

Leach, Larry P., (1999) "Critical Chain Project Management Improves Project Performance," Project Management Journal, Vol. 30, No. 2, pp. 37-51.

Lawrence, John A. and Barry A. Pasternack, (2002) *Applied Management Science: Modeling, Spreadsheet Analysis, and Communications for Decision Making*, New York: John Willey & Sons.

Newbold, Robert C., (1998) *Project Management in the Fast Lane: Applying the Theory of Constraints*, St. Lucie Press, 2000 Corporate Blvd., N.W. Boca Raton, FL 33431-9868.

Pinto, J. K. (2007). *Project Management: Achieving Competitive Advantage*. Upper Saddle River, NJ: Pearson Prentice Hall.

Raz, Tzvi., Robert Barnes and Dov Dvir, (2003) "A Critical Look at Critical Chain Project Management," Project Management Journal, Vol. 34, No. 4, pp. 24-32.

Schwalbe, K. (2007). *Information Technology Project Management*, 5th Edition. Boston, MA: Thomson Course Technology.

Verzuh, E. (2005). *The Fast Forward MBA in Project Management*, 2nd Edition. Haboken, NJ: John Wiley & Sons.

Zwikael, Ofer, Shlomo Globerson, & Tzvi Raz, (2000) "Evaluation of Models for Forecasting the Final Cost of a Project," Project Management Journal, Vol. 31, No. 1, pp. 53-57.

Project Management Institute (2004). *A Guide to the Project Management Body of Knowledge: PMBOK Guide*, 3rd Edition. Newtown Square, PA: Project Management Institute.

1. Internet- http://www.jsc.nasa.gov
2. Lefkon, Dick, *Year 2000: Best Practices for Y2K Millennium Computing*, N.J., Prentice Hall, 1998.
3. McNeill, Thomas, *Cost Estimating and Contract Pricing*, New York, American Elsivier, 1966.
4. Internet- http://www.andersen.com
5. Kerzner, Harold, *Project Management: A Systems Approach to Planning, Scheduling and Controlling*, Sixth Edition, New York: Wiley, 1998.
6. Rakos, John, *Software Project Management for Small to Medium-Sized Projects*, Englewood Cliffs, N.J., Prentice Hall, 1990.
7. Clements, James P., and Jack Gido, *Successful Project Management*, Cincinnati, OH.:South-Western College Publishing, 1999.
8. Duncan, William R., *A Guide to the Project Management Body of Knowledge*, White Plains, MA.:Automated Graphics Systems,1996.
9. http://www.criticaltools.com/pkswbs.htm
10. http://www.criticaltools.com/WBSMAIN.htm
11. http://www.naplesfl.net/~jta/Planning/Activity_Schedule/Activity_Duration_Estimating/compare_current_project_to_past_experience_doc.htm
12. Goldratt, Elihayo, Critical Chain, Three Rivers Press, 1997.