

Delivering Successful Projects

Part 2 “A Better Way”

In part 1 of our series on “Delivering Successful Projects” we determined that most projects have significant safety time included in their task estimates to deal with project uncertainty. Yet even with all this safety included, most projects are late and project lead-time is getting longer. Even when a project is planned and delivered “on time” the actual time that was taken was probably significantly longer than desired. We saw how the “student syndrome” and Parkinson’s Law helped to waste our valuable project safety time. We saw how the very nature of task “integration” in a project network swallowed up the early task completion safety of parallel tasks. And we saw the impacts of “bad” multitasking on extending task completion times. Project managers are working harder than ever to plan and manage their projects and yet the successes that they expect to achieve are as elusive as ever. What can we conclude? Logically speaking, we must conclude, that after applying a set of solutions aimed at delivering successful projects and still not seeing the problems reliably resolved that the solutions must be inadequate. We need to find a better way.

Let’s start with the underlying assumptions on which current project management methodologies are built.

“The way to improve performance of the organization is by inducing local improvements everywhere.”

“The way to have the project finish on time is by putting pressure on each task to finish on time.”

What is the real goal of every project? We want to complete the project reliably as promised. That includes the time of completion, the cost of completion and the results of completion. Does it really matter if a task completes “on time”? All that really matters is the final completion of the project. As project managers, we need to focus on the global objective of project completion time not the local objectives of task completion times. We know from the project network diagram that all tasks are interrelated and have interdependencies and yet modern project management methodologies are primarily focused on task performance in isolation. What if we changed this focus? What if we redirected the focus on to the thing that we know is most critical, reliable project completion time?

In order to shift our project focus from local to global, we will need to make some important changes in our approach to project management. We must determine a method of utilizing our safety time and not wasting it. We must address the issues of the “student syndrome” and Parkinson’s Law. We must also deal with multitasking and multiple projects in our environment. We must change the way we schedule projects and we must change the way we monitor project performance.

Critical Chain Scheduling

We begin by applying the basic focusing steps of the Theory of Constraints. Step 1 is to identify the system’s constraint. In a project we know intuitively that time is a major constraint. (Ultimately project cost and project deliverables are directly effected by the project time). In our current project management methodology we look at the project network, which is made up of multiple paths from the start to the completion of the project, and we identify the longest path. This is referred to as the Critical Path. The Critical Path Method of project management is based on the determination that the project’s completion time is controlled by the length in time it will take to complete the critical path. If we focus on the critical path we are managing the controlling factor in project completion time. One of the major problems with this approach comes from the fact that if a delay occurs on a non-critical path, that path may in fact become the longest path and therefore

become the critical path. The critical path may move many times during the life of a project. We find our focus moving continuously. Another problem with using the Critical Path Method is that the network paths were constructed based on task relationships and interdependencies. But in the real world we not only have task interdependencies but also resource interdependencies. Both of these need to be accounted for. We could have a resource problem that will impact the duration of the project that is not directly on the critical path and yet the project completion is also dictated by this resource constraint. Our system constraint for projects is both time and resources. So we will define a new path through the project that is the longest chain of dependent tasks and resources. We will call this the “Critical Chain”.

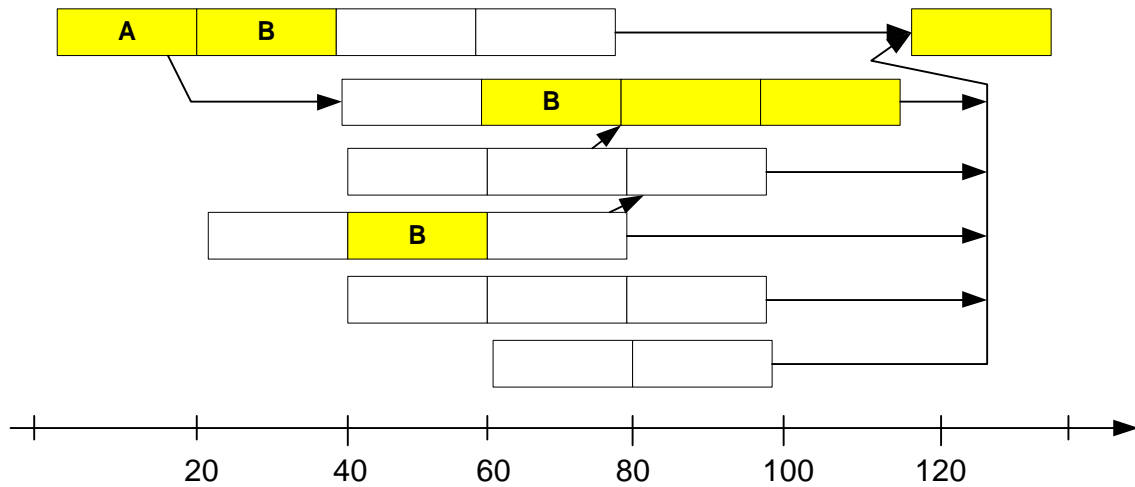


Fig. 1 The “Critical Chain”: The Longest Chain of Dependent Tasks

Next we will apply the second focusing step, which is to decide how to exploit the system’s constraint. Exploit means to squeeze performance as much as possible out of our constraint. Our objective is to protect and manage the “Critical Chain” so as to insure the reliability of our project completion time promise. Let’s start off by examining why we have safety in our project task estimates. Safety time is used to deal with the uncertainty involved in project work itself. Safety time is used to deal with the impacts of distractions and interruptions that are a normal part of our business environment. And, safety is used to deal with the effects of our needing to work on more than one project. But is safety meant to protect our task completion time promise or our project completion time promise? As we stated earlier, we really don’t care about task completion time promises we care about the project completion time promise. That’s our real commitment. We want to make sure that we utilize the safety time to protect our real commitment.

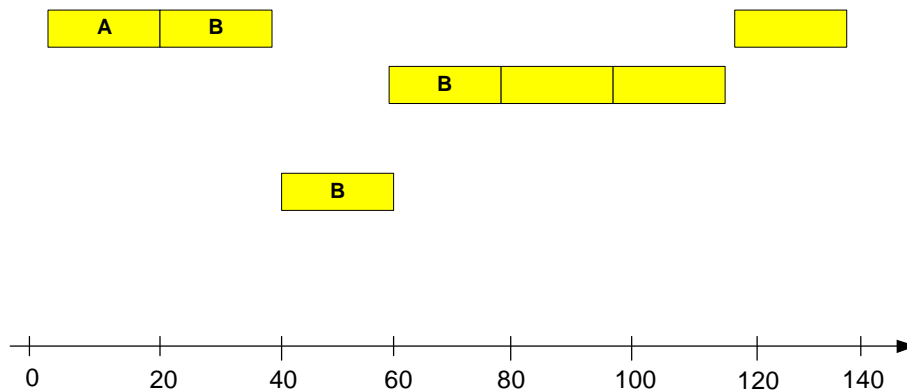


Fig. 2 The “Critical Chain” Before Aggregating The Safety In To A Project Completion Buffer

Let's take all of the safety time beyond a 50% confidence in completion level out of each of the "Critical Chain" tasks. The 50% "completion confidence" level is the amount of time we expect the task work to take if we are allowed to focus a full sustainable level of effort on the it and if there are no significant task related problems. Remember from part 1 of our series that two to three times the focused task work time in the estimate is safety. Most of this safety time is above and beyond the 50% confidence level. But wait, we need most of that extra safety time to compensate for project uncertainty, delays and other distractions. We're taking it out of the specific task estimates but we're going to move it into a buffer at the end of the "Critical Chain", which we will designate as the "project completion" buffer. Now our safety time is positioned to protect the project completion time promise, not the individual task completions. We really don't need all of that estimated safety time. We can cut the safety time in half and still be well protected. A good rule of thumb is that the "project completion" buffer should at least be equal to one-third of the total time in the "Critical Chain".

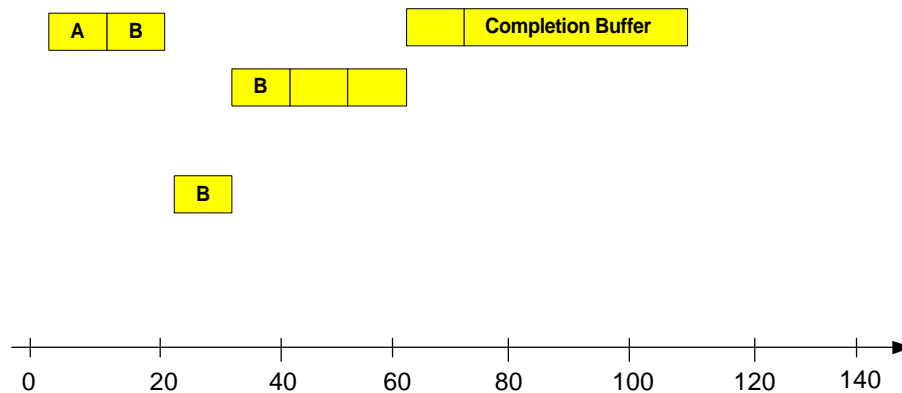


Fig. 3 The "Critical Chain" After Aggregating The Safety Into A Project Completion Buffer

Focusing step 3 tells us to subordinate everything else to the exploitation of the constraint. Subordinate means to support the exploitation but don't do any more than that. So, we will do the same task time adjustments for all non-critical chain project paths. We will remove all the task's safety above a 50% "completion confidence" level and relocate it to a buffer at the end of each non-critical chain path right before this path converges back into the "Critical Chain". We refer to these non-critical chain buffers as "feeder" buffers. Again we can remove one half of the excess safety from the feeder buffer, but still making sure that the feeder buffer is at least equal to one-third the total time of the non-critical chain path. As we will see as we continue our exploration of a better way to manage projects, with this arrangement we can now focus our attention on buffer management to tell us the state of the project and to assist us in making decisions. But more on that later.

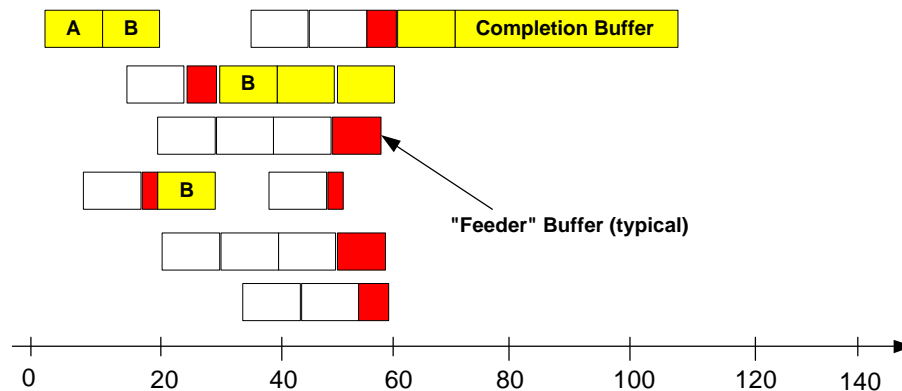


Fig. 4 The "Critical Chain" Schedule Including All Paths With "Feeder" Buffers

Let's see how this "Critical Chain" scheduling will effect human nature and the wasting of our safety time. The "student syndrome" results from people having the perception that they have "plenty of time". We need to alter this perception. Also, Parkinson's Law says, "the work expands to fit the available time". We want to eliminate this as well. Suppose we eliminate task completion due dates and instead instruct the task managers that they are to complete their tasks as quickly as possible. By removing all the safety over the 50% "completion confidence" safety time from each task, there is no longer any perception of "plenty of time". In fact just the opposite, now the order of the day is start your task as soon as possible based on the completion of the prerequisite tasks and work as quickly as you can to finish. Of course, task managers won't accept these new task times unless they feel that they are not being asked to commit to a specified task completion due date and that they will not be "punished" for failing to reach it! One significant change in our management approach, that is essential, is to remove any perception of punitive actions for a task finishing late. We expect tasks to finish early 50% of the time and late 50% of the time. We are no longer concerned with task completion time we are now focused on penetration into the "project completion" buffer and the non-critical "feeder" buffers.

This process puts us into a position where we are no longer tied to a calendar through due dates. We can take advantage of starting future tasks as soon as their predecessor tasks finish early and we avoid the "student syndrome" and Parkinson's Law wasting our valuable safety time. But, if we removed task completion due dates, how do we know when particular resources for follow on tasks need to be ready to start work on their tasks? This is vital for tasks on the "Critical Chain" where we do not want any wasted time. We always want to make sure that "Critical Chain" resources are available when the preceding task is done, without relying on fixed start dates. There are two simple steps required to accomplish this. Step one: Ask the resources how much of an advance warning they need to finish up their other work and shift to interruptible work so that when the proceeding project task is complete, they can drop what they are doing and pick up the critical task. Step two: Require resources to provide regular, periodic updates of their current estimate of the time remaining to complete their current task.

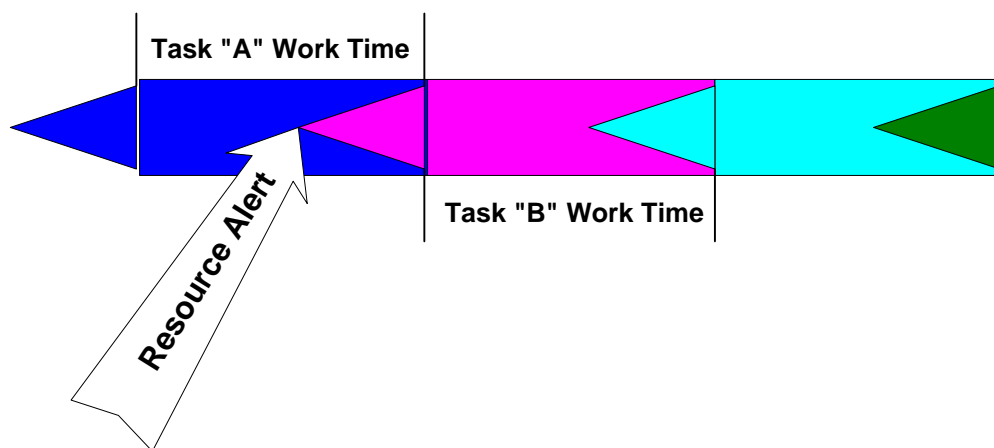


Fig. 5 The Use Of Resource Alerts To Coordinate Task "Hand-Offs" On The "Critical Chain"

When the estimate to complete task "A" matches the advanced warning time needed by the resource who will work on task "B", alert the task "B" resource that the work is on it's way and that they should get ready to pick it up. We now have a system of "Critical Chain" resource alerts. We are also no longer looking at the project from the perspective of "what have we done", but rather from the prospective of "how much time is left" to accomplish unfinished tasks. We don't need to use the "work is coming" resource alerts for non-critical chain tasks, instead we allow the "feeder" buffers to absorb task hand off times. Remember that the feeding non-critical tasks are two buffers ("feeder" buffer and "project completion" buffer) away from impacting the project

completion commitment. For non-critical tasks we don't want to interfere with current task work and create potential "bad" multitasking when it's not critical.

Buffer Management

The use of a "project completion" buffer and "feeder" buffers identifies and protects what is most critical to our delivering successful projects, the reliability of the promised project deliverables completion date. So now our questions are: "how do we manage the execution of the project?" and "how do we know the status of the project once we have started?" The answer is we utilize buffer management. As tasks report in their estimates of the time remaining to complete their unfinished work, we know how much they will either replenish or deplete the associated buffers. We can now focus our attention away from individual tasks and concentrate on monitoring the rates of buffer consumption. Our primary concern at any time is which task is endangering the "project completion" buffer. By endangering we mean consuming time from the buffer. Any consumption in the "project completion" buffer is worse than total consumption of a "feeder" buffer. Except for "Critical Chain" task's buffer consumption, we must consume 100% of a "feeder" buffer before any feeding task consumption occurs from the "project completion" buffer.

Judging The Status Of A Project

- Percentage of "Critical Chain Completion.
- Ratio Between Consumption Of The Completion Buffer And The "Critical Chain" Already Complete.
- Rate Of Consumption Of The Completion Buffer.

We now have a totally new set of project status monitoring and decision measurements. First is the percent of "Critical Chain" completion. What is the percentage of the tasks that make up the "Critical Chain" which is finished. Second, what is the ratio between the consumption of the "project completion" buffer and the "Critical Chain" already completed. For example if we are 60% complete on the "Critical Chain" and have only consumed 10% of the "project completion" buffer we should be feeling pretty good. Of course we need the last measurement to really know the situation, it is the rate of consumption of the "project completion" buffer. How much of the "project completion" buffer did we consume in a measured time period (per week / per month). Any time a task is consuming from the "project completion" buffer we will want to evaluate appropriate actions to compensate for this consumption. We set thresholds of consumption as flags to prompt us to specific levels of action. We now have refocused everyone on the project to watching buffer consumption and to protecting the performance of the project as a whole and not just each individual task. Consequently, we now have a really good chance of delivering a successful project reliably "on time" as promised, and in fact we have a very good chance of delivering the project early. Equally important, we now have a methodology that will also allow us to deliver successful projects with significantly shorter project lead times. We are now using safety time for its original intended purpose, not wasting the safety time, and actually needing less safety time. In part 3 of our series we will address the multi-project environment and how to control or eliminate the effects of "bad" multitasking between many concurrent projects.

Article References:

Critical Chain by Eliyahu M. Goldratt, The North River Press 1997

"Critical Chain Scheduling and Buffer Management" by Francis S. Patrick 1998

Project Management in the Fast Lane by Robert C. Newbold, St. Lucie Press 1998

"The Goldratt Satellite Program" Lecture Series by Eliyahu M. Goldratt 1999

About The Author

Mr. Keslensky has been an executive in the personal computer industry for over 20 years. Today he is the head of Connected Concepts a management consulting firm which specializes in applying Constraint Management and the

Theory of Constraints for Manufacturing, Logistics, Supply Chain Management,
Project Management and Service Industries.

Contact Information:

Phone: 770-481-9992

Jerry.Keslensky@connectedconcepts.net