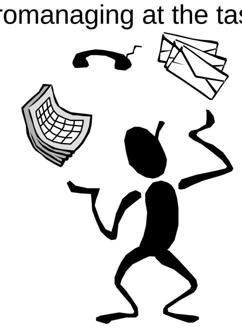
Delivering Successful Projects

Part 3 "So Many Projects, So Little Time"

In part 3 of our series, we want to explore how to reduce the impacts of multi-tasking on our project time management and we want to explore the synchronization of managing multiple projects in the same environment. We also want to expand on how to utilize buffers to make critical judgements in managing our projects.

Recapping from part 2 of our series, we introduced a better way to plan, schedule, and monitor a project. We identified that the goal of any project is to be reliably completed in the shortest time, at the lowest cost, and fulfilling all promised deliverables. We concluded that in general, project cost and deliverable quality was directly related to the project time. This is true even in a project where the deadline is a predetermined fixed date, for example an advertised event, like the Super Bowl. In that situation project time is finite and we must increase cost and decrease deliverables to insure success. In projects where the completion date is not so rigidly predetermined, we allow the project time to expand to protect the quality of the deliverables, unfortunately project cost will increase along with the time. In reality we usually start reducing scope, shaving deliverables, as the time and cost of the project grow beyond desired expectation. Therefore, how we manage our project time, whether finite or not, is critical to achieving the real goal of our project. Beside task interrelationships and dependencies we also identified that project resource availability could also significantly impact project completion times. We determined that the primary constraint to achieving our project goal was the longest chain of dependent tasks and resources, which we named the "Critical Chain". We determined that in order to protect our project completion time promise, we must reposition our task safety time into a time buffer located at the end of the critical chain, which we designated as the "project completion" buffer. We further subordinated the non-critical chain project paths by relocating their task safety into "feeder" buffers located at the end of each non-critical chain path just prior to its merging into the "critical chain". We eliminated task due dates and changed our management approach to remove the perception of expectation or punitive action based on task completion deadlines. And we instituted a method of "critical chain" resource alerts to insure that task handoffs are executed as fast as possible without wasting precious "critical chain" time. Finally, we determined that by using the "project completion" buffer and "feeder" buffers as our points of reference we could monitor and make management decisions without micromanaging at the task level.



Multi-Tasking

Multi-Tasking

In a project environment, multi-tasking occurs when a project resource, an individual or a group, attempts to divide their time and attention across more than one task by successively jumping back and forth between tasks. Multi-tasking could be between tasks on separate projects, or between tasks on a single project and tasks associated with non-project work assignments, or between tasks on two different paths within the same project network. In all cases the effect of multi-tasking is to extend the time needed to complete a task due to the starting and stopping nature of multi-tasking and its associated set-up time. Any project task associated with a multi-tasking effort will take longer than that task would have taken if it had been performed to completion without multi-tasking. We can ask some pertinent questions, "what are the reasons that cause multi-tasking?" and "how can we reduce or eliminate their effects?"

Perhaps the most common cause of multi-tasking is the perception that management expects resources to multi-task. This expectation is expressed through multiple signals they receive. Unresolved scheduling contentions are a signal that management wants multi-tasking. The resource sees that they are being expected to perform two or more different steps at the same time. Of course they know this is impossible, so they perceive that there is a need to show progress on each and every task. People get anxious and feel that showing some progress on each task reduces the risk that they will be reprimanded for not working on the "right" task. Doing one task until it is completed demands setting priorities. Another signal that management wants multi-tasking is the constant insertion of interrupt driven requests. "Just stop what your doing, and take care of this HOT priority first." Or, "we need you to stop and expedite this item." Of course there are causes of multi-tasking that aren't directly encouraged by management. For example, sometimes the work on a task can't be done continuously. The resource may need to wait for information or an approval or some process in which they are not directly involved to complete. This delay causes them to seek other work to fill the void so that they don't appear to be idle. In general this type of task delay is an indication that the task was not properly defined and could have been separated into multiple tasks. The resource would then only be assigned to those tasks that they could complete in one continuous step. Finally, some multi-tasking is caused by human nature. People are easily distracted and may become victims of the temptation to naturally jump between activities. Sometimes it's a burst of creativity that distracts them, sometimes the boredom of dealing with a single task. In some efforts this human desire to multi-task may produce better results. Sometimes the distraction of the mind on one activity promotes a breakthrough idea needed in an unrelated area. But in general this form of multi-tasking leads to poor time management, which is undesirable in most project work.

Single Project Resource Prioritization

Now that we have identified the primary causes of multi-tasking, let's determine the best ways to reduce this time consuming practice. In general, multi-tasking is the result of resource contention. Resource contention means that the same resource is supposed to do two or more different things, steps, at the same time. Removing resource contention between multiple steps necessitates postponing some of those steps. An order of priority must be established. How would we deal with this setting of resource priorities in a single project environment? When we are planning a single project we must first layout a project network which identifies all tasks and their interrelationships and interdependencies. The tasks of a typical project are arranged into a network structure, which consists of multiple parallel paths each consisting of chains of tasks that must be performed in a dependent series.

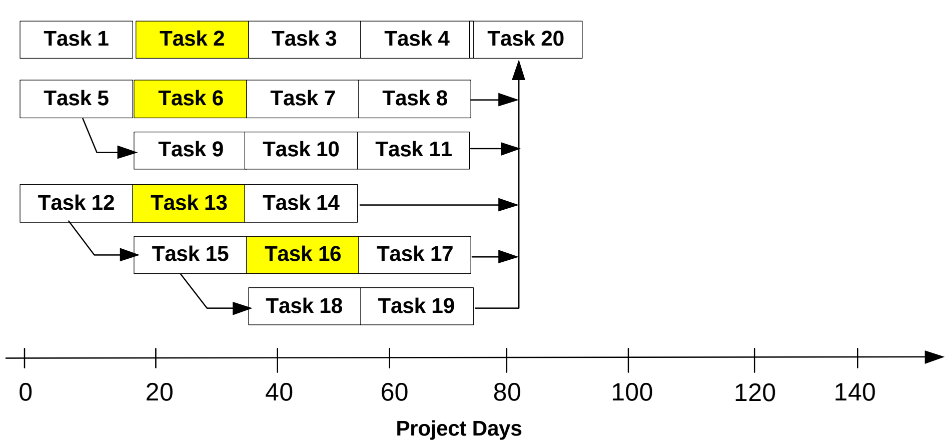


Fig. 2 Project Network With Dependencies Identified But Not Adjusted For Resource Contentions. Shaded Tasks Require The Same Resource.

Once the basic project network has been constructed we assign resources to tasks. Now we begin to look for resource contentions between parallel project paths. The same resource can't work on tasks on two or more separate project paths at the same time. We have to stagger the project path work for a given resource so that they complete all of a task on one path before they can start on a task on a different project path. This staggering based on resource usage will lead us to identifying the "Critical Chain", the longest chain of dependent tasks and resources. In many cases we may want to add additional resources to eliminate contentions. In other cases we may want to reexamine the actual work processes themselves and to utilize methods or procedures that allow the achievement of the deliverables with different resources in a shorten timeframe.

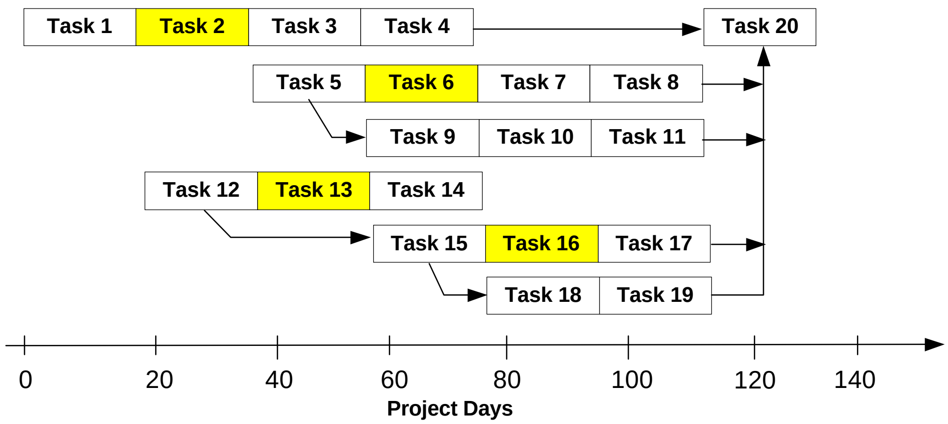


Fig. 3 Project Network Shifted And Adjusted For Resource Contentions

Our objective is to create the shortest possible "critical chain". As the project progresses we may encounter resource contentions created by delays in task completions. Now, in order to determine resource priorities, we evaluate which task has the greatest potential for penetration (time consumption) into the "project completion" buffer. Whichever one has the greatest potential to consume "project completion" buffer time, has the highest priority and therefore gets the constrained resource first. Remember that any consumption in the "project completion" buffer is worse than total consumption of a "feeder" buffer. A non-critical chain task must consume 100% of a "feeder" buffer before it consumes anything from the "project completion" buffer.

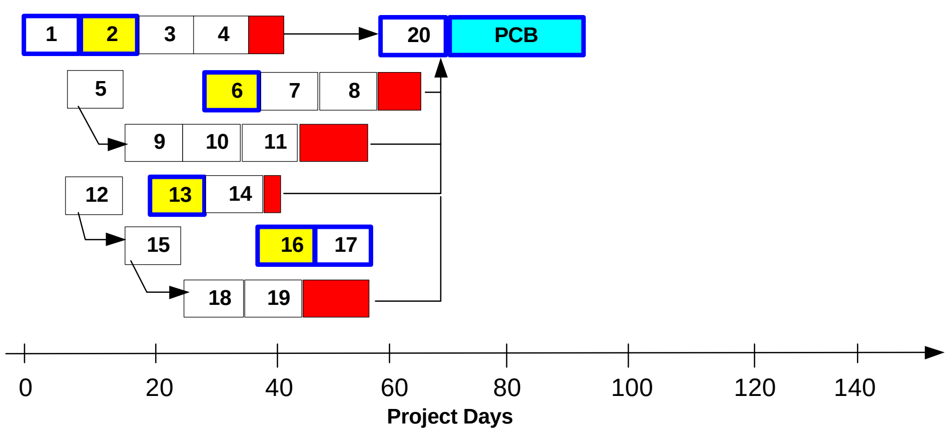


Fig. 4 Project Network With Critical Chain Identified and Completion and Feeder Buffers

Multi-Project Resource Prioritization

Why is resource prioritization not exactly the same for a multi-project environment? There are a number of additional complications that make resource prioritization different in a multi-project environment. We have multiple independent project networks with multiple independent "project completion" buffers to reconcile. We also usually have multiple project managers involved, each with their own set of priorities. Resource contentions are hard enough to identify on a single project network. They are a nightmare to spot when the number of projects is more than a few. But even if you could start off a group of projects with all resource contentions in balance, one late task completion in a single project can create a domino effect of contentions. And remember our task estimates are only approximations at best and can easily be off by 50% plus or minus just based on the 50% completion confidence from which the estimate was derived. We can't ignore resource contentions in a single project, and we certainly can't ignore them in a multi-project environment. So what can we do? To start off, we know that a single resource can only do one task at a time. We could have many resources in contention between multiple projects, but we can never successfully prioritize multiple resources across multiple projects. We can however prioritize a single resource across multiple projects. On which resource should we choose to focus our attention? Constraint management teaches us that in any system there is usually only one constraint that is responsible for most of our limitations. For our multi-project environment we will designate this resource constraint as the "company capacity constraint" resource. It could be a person, a department, or even a policy. Our first step is to identify the "company capacity constraint" resource. Start off by asking project managers who have previous project experience in the company, where did they spend most of their time trying to expedite their projects? Which resource is the resource that project managers are fighting the most to get more of than any other resource? Project managers do have strong intuition whom or what are delaying them the most. The competition on certain resources may go all the way up to the top management of the company. When it does, you know there is a serious lack of capacity available.

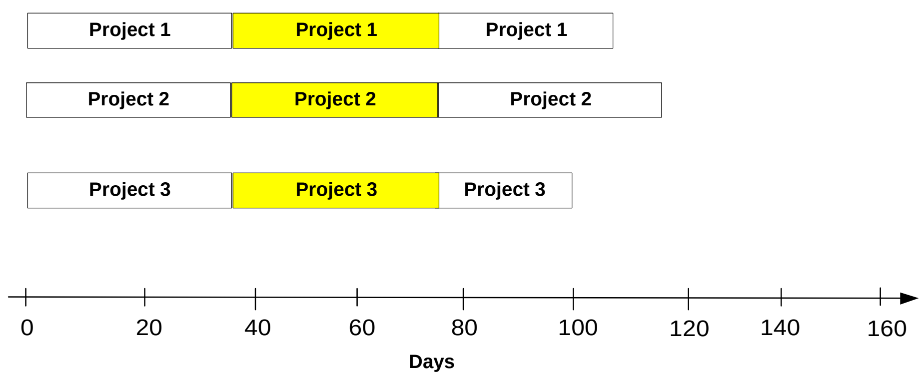


Fig. 5 Multi-Project Environment With A Resource Capacity Constraint

Once we have identified the "company capacity constraint" resource, we need to estimate its capacity. We will then exploit this constrained resource by carefully scheduling the work that it does. We want to squeeze every ounce of productivity out of this constrained resource. Each project in our multi-project environment should initially be planned and scheduled as if it were a single project. Then we will stagger the projects using the "company capacity constraint" resource as the "drum" or source of project synchronization. The staggering of projects is similar to the way we stagger common resource task starts between parallel paths in a single project. The general rule for how projects are ordered in the series, the precedence of the staggering of the projects through the "company capacity constraint" resource, is based on their target project completion date. The project with the earliest target completion date goes first, etc. We will add "capacity" buffers between projects to insure that the "company capacity constraint" resource is available for each subsequent project.

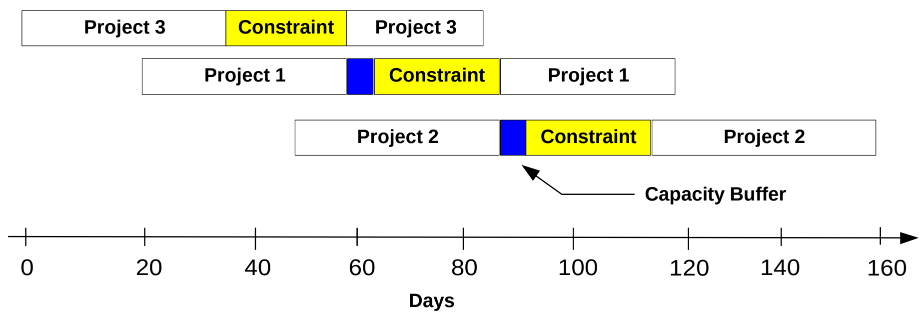
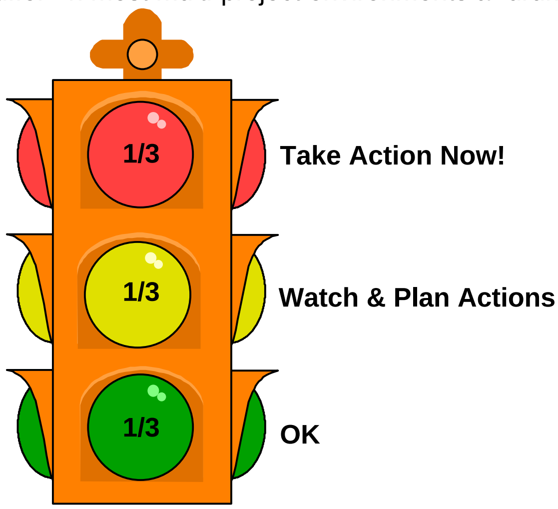


Fig. 6 Projects Shifted Around Capacity Constraint Sequenced On Project Completion Date

For example if project “1” has a task that is scheduled for the “company capacity constraint” resource and project “2” has a task for that same resource that is scheduled to follow the completion of project “1’s” task, then we will place a “capacity” time buffer between the two tasks. If project “1’s” task completes early, project “2” can take advantage of the timesaving. If project “1’s” task completes late it will penetrate the “capacity” time buffer, but based on proper buffer sizing, the “company capacity constraint” resource will most likely be available at the needed time by project “2”. We delay the initial start of project “2” to coincide with the timing of when the appropriate task in project “2” will have availability of the “company capacity constraint” resource. This may seem counter intuitive to your normal impulse to start all projects as soon as you can. But in reality, if the task requiring the “company capacity constraint” resource can’t start until, at the earliest, the beginning of the “capacity” time buffer, then the entire project start should be shifted accordingly. In most multi-project environments even though we staggered the project starts based on the availability of the “company capacity constraint” resource. We actually see the majority, if not all, of the projects finish significantly sooner than had we started them all at the same time and suffered through the multi-tasking delays of the “company capacity constraint” resource. In fact, the staggered starts usually help to eliminate other resource contentions beside the “company capacity constraint” resource. Particularly if the projects are very similar in terms of their individual project networks. That is they have the same types of tasks needing the same types of resources in the same task sequences. This condition is not unusual in most multi-project environments. The projects have different deliverables but are in fact similar types of projects. One other important area to watch in a multi-project environment is the non-critical chain “feeder buffers” for a project. If we see that a resource contention starts to exhaust them one after another, then we may have an additional bottleneck resource to manage.

Buffer Management

In part 2 of this series we touched on buffer management. Before we summarize what we have learned, we should expand our understanding of how to apply the time buffers to help us make difficult project decisions. We have identified four types of time buffers that are utilized in Critical Chain project management. The “project completion” buffer placed at the end of the project’s “critical chain” is an aggregation of task safety time taken from tasks that comprise the “critical chain” of the project. Its purpose is to protect the project completion date promise. The “feeder” buffers are located at the end of each non-critical chain project path just before the path merges with the “critical chain”. Each “feeder” buffer is an aggregation of task safety time taken from tasks that comprise that non-critical chain path. Their purpose is to subordinate the non-critical chain tasks to the support of the “critical chain”. They are providing protection of the “critical chain” from non-critical chain task completion delays. Only when 100% of a “feeder” buffer has been depleted does the associated non-critical chain tasks have an effect on the “critical chain” and its “project completion” buffer. The “resource” buffers are time buffers that we use as part of our “critical chain” resource alert system. They are the notification time buffers that are used to alert a resource that a “critical chain” task is about to be handed off and are used to insure that the pending resource is ready to stop their interruptible work and begin the up coming task. In multi-project environments we have added “capacity” buffers between projects to insure that a “company capacity constrained” resource is available for each subsequent project. A fifth type of time buffer may be used to protect the "company capacity constrained" resource itself. This is referred to as a "drum" buffer. In most multi-project environments a "drum" buffer is not required.



Buffer Management Signals

The project team should monitor the "project completion" buffer and the "feeder" buffers at appropriate time intervals for the project. This is usually a weekly task. The resource buffers need to be monitored more frequently, in some cases daily depending on typical task execution times. The "project completion" buffer and the "feeder" buffers must be monitored at least as frequently as the time interval equal to 1/3 of the total time of the "project completion" buffer. Project managers update the buffers as often as they need by simply asking each task manager to estimate how many days they have left to the completion of their current task work. Remember that the monitoring and resource alert processes are based on each task manager reporting the amount of time they expect to be remaining to complete their unfinished task work. Not how much work they have done so far, but how much time they still need. If the penetration into the buffers is negative, meaning that the latest task on that chain or path finished early, or the penetration into the buffer is less than 1/3 of the total buffer you need not take any action. If a task is expected to have an extended duration or actually finishes late and the associated buffer penetration is between 1/3 and 2/3. Then the project team should plan actions for that chain or path to accelerate the current or future tasks and recover the buffer. If task activity performance penetrates more than 2/3 of the buffer, the project team must take immediate action. Buffer management provides a clear anticipatory tool for project actions and decisions. Buffer penetration should be tracked against a regular time interval to give the project team a perspective of the rate of buffer consumption. An accelerating consumption rate is a strong signal that planning and action are going to be required and to start looking for problems. In all cases decisions are predicated on the net impact on the "project completion" buffer. The highest priority of action or attention should always go to the task that has the greatest potential to penetrate the "project completion" buffer.

Summary

A company's ability to deliver successful projects consistently is greatly enhanced by adoption of "Critical Chain" Project Management. Time management is critical to delivering successful projects. The goal of any project is to be reliably completed in the shortest time, at the lowest cost, and fulfilling all promised deliverables. It is very important to refocus everyone to protect the performance of the project as a whole and not just each task. The use of properly positioned time buffers provides the project team with an excellent tool for monitoring project performance and determining difficult project decisions. Resource contentions must be understood and accounted for in the planning of projects. The concept of "critical chain" includes both aspects of structural project task dependencies and resource dependencies. In single projects, the magnitude of buffer penetration is utilized during project execution to determine resource contention issues. In a multi-project environment a “company capacity constraint” resource must be identified. Its capacity estimated and it must be carefully scheduled to insure that its maximum productivity is maintained. The “company capacity constraint” resource is used as a “drum” to synchronize all projects. Wherever possible we want to reduce the number of tasks dependent on a single resource. We stagger project starts based on their target completion date around the availability of the “company capacity constraint” resource. The “company capacity constraint” resource must always finish a task before we allow it to start on the next task. We never let a resource jump back and forth between tasks or projects.

Finally we have created three important new 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. The last measurement to really know the situation 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).

We hope that this series of articles has provided you with some fresh insights into how to deliver successful projects. We have outlined the foundation for a better way to plan, monitor, and manage projects. There are many related issues, which we have not addressed here, that are of critical importance to consider. Some examples are: organization change management and resistance to change, rationalizing time estimates, proper levels of task definition granularity, investments in increasing resource capacity, managing subcontractors, and project feasibility decisions just to name a few. But any organization that depends on successful and reliable projects can utilize the foundation given here as a major starting point for ongoing improvement of their project delivery process.

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 Press1998
"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