Critical Chain: A hands-on project application

Ernst Meijer

Introduction

How often do we hear that projects are late, over budget or fail to meet all technical objectives? One of the first lessons new project planners learn when beginning their career is that projects inevitably seem to slip past their due dates. This can become really frustrating for the novice since it seems to undermine the very professional reason for existence. Thus, a concise and detailed planned project turns into a well-documented failure.

The Critical Chain approach is a project management application of the Theory of Constraints (TOC) of Eliyahu M. Goldratt. According to this approach the mechanism that causes a project's failure to deliver is project management's inability to recognize a project as a network of dependent events with statistical fluctuations and the resulting failure to account for uncertainty. As we will see, if projects safeguard for uncertainty at all, it is usually by adding time to individual activities. Even if we reserve some contingency funds in the budget to account for unexpected expenses during project execution, time, or better, the lack of time, is the prime driver for cost increases and quality deficiencies. Overtime and cutting corners in order to make timing is not an exception, but a rule. Examples are plentiful. Engineering will deliver incomplete CAD data after long work weekends just to make due dates. Expensive materials for a new design are used because of the lack of time available to evaluate more cost effective alternatives, and an ad-hoc decision to use costly rapid prototype tools buys more design time. After the project is finally finished a separate cost reduction team has to come in to reevaluate the design and its manufacturability on cost and quality and to make incremental improvements. The additional millions of dollars invested in this effort to reduce product cost by a percentage point may be justifiable in the eyes of an accountant over the life cycle of the product but, it takes expensive engineering and manufacturing resources away from the primary business of the company: to make quality products within budget and on time.

The paper will discuss how traditional project schedules protect for contingencies and why it is insufficient, introduce the main concepts of the Critical Chain approach and conclude with a brief report on an application of this approach in the Big Three automotive environment.

Project Contingencies

Contingency Time Added

To get a better grasp on how projects inevitably fall into a timing crunch the section below analyzes how schedules usually deal with contingency time.

The Nature of Tasks

Once all the activities are identified to design, build, test and produce the product, the activities are scheduled as discrete tasks with Start and Finish dates and combined in a complex network of predecessors, successors and resources. The durations of scheduled tasks are estimated durations pulled out of an experience database. The project planner's software allows for the capturing of estimated task durations up to decimal points of precision, but what does it actually mean? A typical task duration distribution curve is displayed in Exhibit 1.

The horizontal axis of the figure is time and the height of the bold curve represents the likelihood of finishing a job during any time period. The curve climbs steadily at first and peaks out at the point where the job is most frequently finished (the "mode" of the curve), and concludes with a long tail to the right. The normal time variation in duration of the task can be found on the time axis in the "Common Causes" section. Infrequently, certain events can push the completion date out considerably, as shown on the right, or finish the job in homerun time as shown on the left, as highlighted in the sections titled "Special Causes". So what happens when a task owner has to estimate the required time to finish a job? One can take the median of the curve but, that is, per definition, wrong half of the time. Since, in the corporate world, workers are measured by how well they follow schedules, the estimator usually shoots for a safe 75% or 90% confidence level, thereby doubling or tripling the time estimate. The same reward system applies to the supervisor of the team who will likely add 20%-25% of elapsed time as protection, adding a safety margin to cover the unplanned but predictable late product changes and administrative overhead activities that will come up.

Indeed, since every scheduled task in a project plan has a fair amount of safety built in it, the project lead-time balloons to a level where it will trigger the attention of management. Under the pressure of a competitive environment or of product cycle time considerations, it is upper management's objective to reduce the overall project time in order to meet or beat financial targets. It becomes the program manager's task to renegotiate the timing of the project and ultimately cut lead-time.

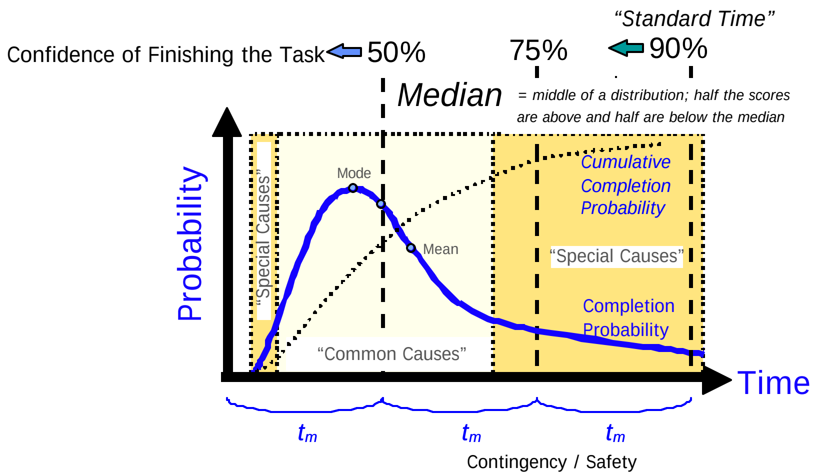

Exhibit 1: Task Completion Distribution Curve

This cat and mouse game between workers and management is a recurring ritual in program management and another motivation for workers to determine their time estimates on the high end, since it is inevitable that they will end up with less time for their jobs than they asked for. This is not a description of a conspiracy but of reality where everybody does his or her job. Is it unrealistic to expect a job owner to submit a worst-case time estimate in order to be sure the deadline is met, or for lower management to cover their resources for anticipated activities outside of the original statement of work, while upper management acts out of concern for the bottom line?

Contingency Time Wasted

If projects naturally have all this contingency time baked into their work plans, why are so many still running late? There must be something wrong with how uncertainty is managed and how safety time simply disappears. The following sections will elaborate on this phenomenon.

Multi-tasking

Project resources usually have to work on more than one task concurrently. Project managers and staff may be dedicated, but designers and engineers, for instance, often have to juggle their time between several projects and satisfy all internal customers. If the worker has three jobs that each take one workweek and divides the time evenly over the jobs to satisfy multiple needs, it will take three weeks before all three jobs are finished. This is multi-tasking at its worst. The throughput per task has lowered one third and the elapsed time to finish each job has tripled. If, on the contrary, one would complete the jobs sequentially (first Job A, then Job B, then Job C), each job would have a throughput and elapsed time of one week; this is an example of multi-tasking at its best which is rarely seen because little to no effort is usually made to coordinate shared resource schedules on concurring projects.

Student Syndrome

A job may begin on its official Start Date but most people do not gear up their work effort until the deadline is in sight. In the beginning, there are still so many other things to finish, or there is finally some time to get better organized and catch up with new program directions. The job owner knows that the scheduled work time has some slack in it and they tend to take advantage of it. This work pattern defeats the purpose of contingency time since it consumes it before the work is really started. If during the actual job performance a real problem pops up, the chance to meet the deadline diminishes while the only remedy is to go into overtime. Student Syndrome behavior is a major contributor to "fire fighting" crisis situations.

Schedule Syndrome

"*Work expands so as to fill the time available for its completion*"(Newbold 1988, 51). There is no incentive for a worker to finish work early. First, nobody wants to sit idle and look superfluous. Second, if one finishes a job early there is a good chance that the next activity in the chain is not ready to start anyway. The Schedule Syndrome makes planned Start and Finish Dates a self-fulfilling prophecy. Third, there is a good chance that the supervisor will reward a worker by assigning some intermediate task to help out another activity that is currently in a crunch mode and, chances are, the worker will be late to start the next scheduled job. Our corporate culture usually penalizes people who fail to meet a deadline but it does not reward people that finish a job early. However, those that complete their work on the scheduled Finish Date are praised.

Path Merging Effect

If Purchasing manages to cut the price of one of two parts of an end item, the total assembly will cost less than budgeted. However, if in a network, one of two parallel tasks finishes early (let's say -5 days), the common succeeding activity cannot be started until the other predecessor is finished. If that task finishes late (+5 days), the succeeding activity will have to start late as well. We cannot use the intuitive rule of averaging delays out with early finishes. In other words, merging delays (negative variances) accumulate while gains (positive variances), if they happen at all, are mostly lost as wait time.

Critical Chain Scheduling

As previously stated, in a standard project plan each task has a considerable amount of safety time built in as padding for unexpected delays; however, this safety time is either lost or consumed during project execution and is not available when Murphy really strikes. The resulting time crunch is then responsible for the failure to meet deadlines and has an adverse affect on cost and delivering the full technical scope or meeting quality requirements of the project. How then can we meet the lead-time of a project while effectively managing uncertainty? The Theory of Constraints turns this argument around. By focusing on managing uncertainty, a project can meet or beat its original lead-time. Let's take a look at the Critical Chain.

Project Buffer

The Critical Chain of a project is defined as the set of tasks that determine the overall lead-time of the project, taking into account both path precedence (like the critical path) and resource dependencies (unlike the critical path). Once resources are leveled and a path is determined in the network, focus must be placed on the tasks on the Critical Chain and protected for contingencies by creating a buffer (Project Buffer) at the end of the chain.

In order to create the Project Buffer, safety time has to be taken out of each and every task on the Critical Chain and aggregated at the end of the chain to make safety time available for those tasks that need it. Let's say, for sake of argument, that we take a conservative approach and decide that all dedicated task durations can only be reduced to their average (mean) value, which is always greater than the median value since the task completion distribution curve has a positive skew. The resulting new chain becomes 50% shorter but very risky since it uses average task durations. If it is possible to eliminate the Schedule Syndrome (never start early and always finish as late as possible) and the Student Syndrome (starting late) and manage resource conflicts (bad multi-tasking), then one can capitalize on all cases that have an early finish, i.e. earlier than the average finish time, because each early finish will immediately be followed by an early start of the successor task. In other words, if one can capitalize on positive variances from the average and not just have to add the negative variances to the chain, the length of the Project Buffer will be less than the 50% taken out. As a rule of thumb, the initial Project Buffer size should be 50% of the unpadded critical chain duration, creating a total project time of 50% + 25% = 75% of the original lead-time estimate. The available Project Buffer will change continuously during project execution; when a task on the Critical Chain finishes early the buffer will increase because all succeeding start dates are pulled ahead while, if a task takes longer than average it will consume a part of the buffer by pushing the start dates of all successors downstream. This movement can be visualized by imagining the Critical Chain as an elastic spring rather than a solid chain. Through these statistical fluctuations the exact project end date is thus unknown but will reside on the Project Buffer. Typically a Critical Chain project finishes at 60% of the original Critical Path lead-time estimate.

Feeding and Resource Buffers

Besides the tasks that are on the Critical Chain there are other activities in our project network that need to be started and ultimately will feed into its path. Feeding Buffers are placed wherever non-critical paths of activities merge with the Critical Chain. Since these Feeding Buffers, like all buffers, are unlikely to be fully consumed, they allow succeeding tasks on the elastic Critical Chain to start early if everything goes well. Placing buffers strategically where paths merge avoids that upstream schedule advances are getting lost. Sizing of Feeding Buffers uses the same rule of thumb: 50% of the length of the preceding unpadded chain of tasks. (See Exhibit 2)

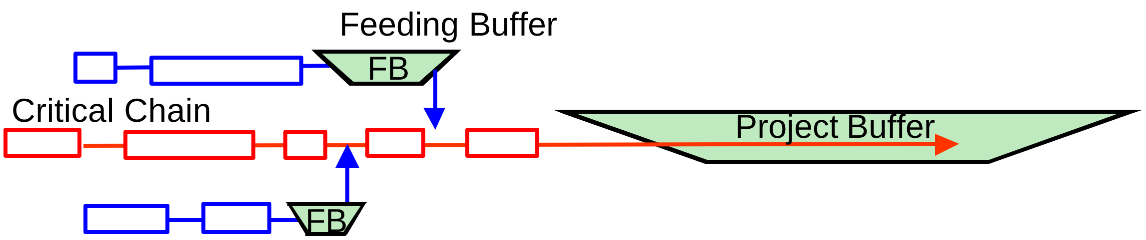

Exhibit 2: Single Project Critical Chain

Critical chain tasks always get priority over non-critical tasks. Resource Buffers make sure that the resources are ready to do their work on Critical Chain tasks. The Resource Buffer is the time period in which the resource is alerted to get ready to start work. These wake-up calls are needed because the task start dates on the Critical Chain are in a constant flux. Once the initial work plan is scheduled, the daily updated plan does not show durations or task Start/Finish Dates -except for tasks without predecessors. Task owners are not asked to report their work progress in percentage of completion since this assumes fixed task durations and focuses on the amount of work done. What we really want to know is the amount of work that is left to do. In the Critical Chain method the question is how many days are still needed to finish the job, and this estimate will fluctuate during task execution depending on the severity of obstacles that are encountered. This kind of dynamic scheduling and progress measurement stimulates task owners to finish their work as soon as possible and discourages the complacency and self-fulfilling prophecy of fixed dates.

In a multi-project environment there are two additional resource buffers. With multiple projects it is impossible to synchronize all resources across all projects. Instead, focus on the critical few: shared capacity constraint Drum Resources (department, facilities, equipment). If increasing the capacity of such shared resources cannot be done, then it is necessary to protect and monitor workloads. Project management synchronizes the projects using the Drum Resource only, and leaves handling the other resource demand fluctuations to the project buffers. A Capacity Buffer is placed between the projects to link the use of the Drum Resource, while Drum Buffers are placed in each project before the Drum Resource to ensure that the resource is not starved.

Buffer Management

How can project management protect the project from the variability of the processes that impact it? In this dynamic scheduling process, how can one identify the areas that are running out of control and safeguard the project Finish Date? This control mechanism is called Buffer Management. Project and Feeding Buffers offer project management a clear focus point: manage a handful of buffers instead of trying to manage the hundreds of individual tasks. A buffer can be divided into three time zones. Each time zone requires its own set of managerial actions. No interference is needed as long as the buffer consumption is restricted to the first ("Green") zone, the farthest from the deadline. In the second zone ("Yellow"), management analyzes the problem and induces recovery plans from the task owners and resource manager. In the third zone ("Red"), management is directly involved in implementing corrective actions. Buffer Management is the process of managing variability or uncertainty to improve throughput.

Reward Systems

"*Tell me how you measure me and I'll tell you how I'll behave*" (Goldratt 1990, 26). An essential component for the Critical Chain method to work effectively is that corporations need to rethink their reward systems. Workers need incentives to start and finish their job as quickly as possible and with quality. Workers should not be penalized if this results in some idle time before a new job comes up. The idea that efficiency should mean that everybody is constantly working is, as discussed earlier, a self-fulfilling policy that increases, not decreases, project lead-time. Instead of thinking in terms of workload we should think of workflow, of the Critical Chain as a relay race where you handover the baton as soon as possible instead of after your prescribed running time.

One possible alternative reward system is to use Throughput Dollar Days (TDD) to measure work performance. TDD is defined as a task's buffer usage in days times the selling price or budgeted cost of the entire project. TDD forces everybody in the team to take a CEO's perspective by focusing on the global project impact of his or her task performance. For example, if slow completion of a task on the Critical Chain causes 3 days of additional consumption of the buffer, the worker(s) responsible for the lateness will get assigned 3 days times $100 million (let's say that is the project's price), which equals 300 million dollar days debit. On the same token, if a worker adds 2 days to the buffer by finishing early the worker earns a credit of 200 million dollar days. Assigning responsibility for the task performance is, of course, the hard part and has to be done very carefully. If a late finish is due to a late change in product specifications, it can very well be attributed to upper management and not to the task owners. Nobody in the team is excluded from this throughput measurement.

Critical Chain Project Application

Project Description

For analyses of a real life project application of the Critical Chain approach we move to the automotive industry. In one of the programs of the Company, management decides to make a late change for Job1 to improve product quality. It is a complex change that involves 6 parts and 3 suppliers. The deadline of the Project is determined by the timing of a dedicated pre-production build that is locked in by the Job1 date. This means that the workday before the start date of the build becomes the Project finish date.

The Project is in its sixth month from kickoff and has three more months before the deadline. The Project has encountered many delays and is currently in a compressed mode. Prototype tools for the parts have been kicked off as well as production tools. Management considers project completion a high risk and requests Planning to become part of the team.

Buffers are not found but have to be created

The first step is to review the current workplan, determine the Critical Chain and identify buffer points. The current Project Workplan shows the three separate supplier workplans stacked together. Each supplier workplan ends with the component PPAP (Production Part Approval Process) finish date and two suppliers schedule this event on the Project's end date. Planning decides to focus on these two suppliers and to create a project buffer at the end of each path, after the PPAP finish date. The Project Workplan, therefore, will have two parallel Critical Chains. For Resource Buffers (wake-up calls) a teleconference is scheduled three times a week. Resources are not shared so there is no need for a Drum Buffer.

The second step is to create the buffers. This is the most difficult part of the process because of the cultural change that is involved. The Planning's strategy is to find areas in the supplier workplans that can be reduced further and to set that time aside as Project Buffer. This can only be accomplished if the supplier understands and cooperates with the strategy. Teamwork needs to be forged but also forced before it can become successful: Planning announces that it will call daily meetings with each supplier until the target is reached. An important tactic is to insist on talking with the tool shop owners directly, instead of supplier program managers, since the tool shop owner knows every detail of the tool plan and is the best expert in identifying excessive or just cautious time estimates in the plan.

After ten days of negotiating, management of the two critical suppliers both sign written agreements that outline "super compressed" workplans leading to a target component PPAP date far in advance of the required Project finish date. Each agreement spells out on what tasks time is taken out and where an extended work schedule (6d/w or 7d/w) will be applied, resulting in a pull ahead of 16 days for Supplier A and 17 days for Supplier B. The compelling argument for the suppliers to voluntary commit themselves to this super compression is that the official delivery date remains the original Project finish date; they sign up for a home run schedule with the comfort of having a buffer at the end that can be used if needed.

Progress Tracking

The Project Workplan is rewritten and includes all detail tasks derived from the tool workplans, all DVP and PV tests and shows the various buffers. Once committed to the super compressed schedule both suppliers try to follow it as if there is no buffer at the tail end. In other words, nobody is "slacking off" and quite a few early finishes are reported during the Project. Tight progress monitoring (3x per week phone conference plus weekly meetings) helps to maintain this discipline.

Buffers for Risk Management

At the end of the Project, Supplier B did not use any of their accumulated buffer time; they hit all consecutive home runs. Supplier A, however, consumed about all of their buffer time due to unexpected issues ("Special Causes") that prevented an early PPAP. A time slot for one of these issues (component redesign and production tool change) had not been allocated for in the original workplan and thus was not an artifact of the super compression effort, while the other issue (production tool tuning) took much longer than foreseen. If the original workplan had been followed, the Project would have slipped its deadline. Therefore, the buffer proved to be an ideal tool for risk management. Exhibit 3 shows the consumption of the project buffers over time.

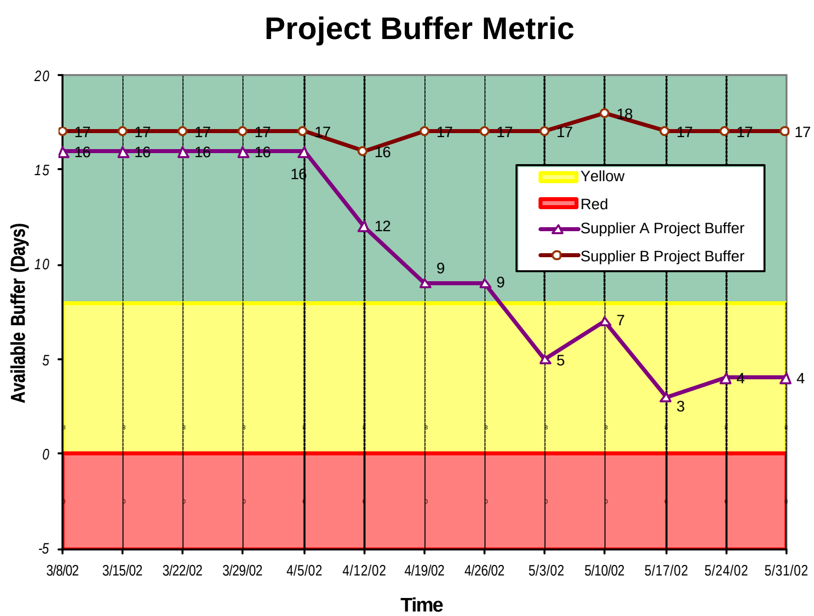

Exhibit 3: Project Buffer Metric

Lessons Learned

Using the experience gained from this project, what are the main enablers for the successful application of the Critical Chain method? First, management has to fully understand, trust and support the new rules of engagement and give the Project Planner a free hand to roll it out and work through the initial bewildering of the team. Second, the communication that buffers are not float or slack time but integral part of the schedule is crucial. In the described project, the task owners (the suppliers) were willing to participate once they felt comfortable that they still owned the time that was taken out of their tasks, only now aggregated in the Project Buffer. The sign-off of written agreements that defined this for all parties involved turned out to be a catalyst to earn their engagement. Third, task owners have to be motivated to do their work as fast as possible without having reservations about adding time to the buffer that will be available to others downstream. Since each critical chain in the project was owned by a supplier this form of self-interest was not a serious problem.

The planning software used in the project was MS Project 2000®. Creating buffers and automating the display of buffer consumption days was possible but required special attention during schedule maintenance. There are Critical Chain dedicated software programs on the market of which ProChain® Project Scheduling is an add-on to MS Project. These dedicated software programs are a requisite for scheduling more complex Critical Chain projects.

Conclusion

In project management, the Theory of Constraints (TOC), laid out in Goldratt's best-known books "The Goal" and "Critical Chain", has many readers but few practitioners. The main reason for this is that TOC literature makes a good read in theory, with intriguing promises like cutting the lead-time of projects with 40%, but at the same time seems to depart so radically from "how business is normally done" that most readers' fascination stops the moment the books get closed. This paper has demonstrated however, that even in the rigid environment of one of the "Big Three" automotive companies, there are niches of opportunities to use this approach successfully. A full implementation of the Critical Chain approach on complete model year programs will first need drastic organizational and cultural corporate changes but, with increasing competitive pressure to shorten cycle time and lower cost while maintaining quality, its time has come.

References

Goldratt, Eliyahu M. 1990. The Haystack Syndrome. North River Press, Inc.
Goldratt, Eliyahu M. 1992. The Goal. North River Press, Inc.
Goldratt, Eliyahu M. 1997. Critical Chain. The North River Press Publishing Corporation.
Goldratt, Eliyahu M. 1990. Essays on the Theory of Constraints. The North River Press Publishing Corporation.
Goldratt, Eliyahu M. 2000. Necessary But Not Sufficient. The North River Press.
Leach, Larry P. 1997. The Critical Chain Project Managers' Fieldbook. Quality Systems
Leach, Larry P. 2001. Critical Chain Project Management. PMI Training & Professional Development Seminars
Lepore, Domenico, Cohen, Oded, 1999. Deming and Goldratt: The Theory of Constraints and the System of Profound Knowledge. The North River Press Publishing Corporation.
Newbold, Robert C. 1988. Project Management in the Fast Lane: Applying the Theory of Constraints. St. Lucie Press/APICS Series on Constraints Management.