Critical Chain Project Management: Motivation & Overview

Hilbert Robinson
Afinitus Group, LLC
705 SW Oak Road
Port Orchard, WA 98367
425-231-4637
hrobinson@afinitus.com

Robert Richards
Stottler Henke Associates, Inc.
951 Mariner's Island Blvd, Suite 360
San Mateo, CA 94404
650-931-2700
richards@stottlerhenke.com

Abstract—This paper provides an overview of the aspects of Critical Chain that make it successful, and then provides an introduction to Critical Chain and its application. Project Management involves making and keeping commitments under uncertainty, accompanied by complexity and interdependency. In most project management environments, making binding commitments is expected in three separate dimensions: 1) schedule or time, 2) resource or budget, and 3) scope, quality, or performance objectives. Falling short of a commitment can result in the project being deemed a failure, with attendant negative consequences to stakeholders. Evidence suggests a high rate of Project Management failure exists industry-wide.

Searching for solutions to the problem of high project failure rates yields many valuable contributions. However, a need remains of finding an approach to project management that is effective across diverse domains, and that can be taught to, and successfully applied by, the majority of project managers of average abilities and experience. A technique called Critical Chain has potential in this regard and is backed by over a decade of field testing and refinement.

Critical Chain takes a different approach to handling risk versus most traditional methods of project management such as critical path project management. Traditionally the risk associated with a task is handled by the duration estimate of the individual task. Due to the Student Syndrome and Parkinson's Law this method has not proven to have the desired effect. By utilizing a pooled or aggregated risk methodology, task durations can be shortened to the task's average time to completion, and the variability of the tasks in actuality can be planned for and handled via buffers placed in locations that protect the project as a whole. The buffers utilize the safety time removed from individual tasks. However, it has been found that using more aggressive task durations in conjunction with the buffers results in shorter overall project durations and better on-time performance. [1][2]

TABLE OF CONTENTS

1. INTRODUCTION

Project management deals with planning, organizing, and managing resources to bring about the successful completion of specific project goals and objectives. Project management is about making and keeping commitments under conditions of moderate to extreme uncertainty accompanied by significant levels of complexity and interdependencies. In most project management environments, it is expected that binding commitments will be made in three separate dimensions of a project before the project starts:

1) Schedule or time
2) Resource or budget
3) Scope, quality or performance objectives

Falling short on any of these separate commitments can result in the project being deemed a failure, with attendant negative consequences to stake holders. Although each industry has its share of successfully concluded projects every year, evidence suggests that there is a high rate of Project Management failure across all industries. In an article on its website, The British Computer Society states that, "Project failure is not discriminatory – it pretty much affects all sectors and all countries in equal measure [3]."

The search for solutions to the problem of high rate of project failure has yielded many valuable contributions and, according to The British Computer Society, quoting from the Standish Report, some progress has been observed over the years, at least in the IT sector. However, there is still a need to find an approach to project management that is effective across a wide and diverse domain and that can be

[1] 978-1-4244-3888-4/10/$25.00 ©2010 IEEE.
[2] IEEEAC paper #1436, Version 1, Updated September 23, 2009

taught to and successfully applied by the majority of project managers of average project management abilities and experience. After over ten years of field-testing and refinement, Dr. Goldratt's Critical Chain [1] appears to offer potential in this regard.

2. The Problem

In environments where formal project management practices are used, project commitments are typically derived through a rigorous planning process governed by standard practices. In order to derive a realistic commitment regarding duration and cost of delivering a specified scope of work, developing a model of the logistical process through which the scope will be delivered is necessary. The product of this model-creation exercise is generally known as a schedule or project plan. The model usually reflects the explicit requirements, and the assumptions regarding the conditions under which the project will be executed.

However, in self-identified project management environments, there typically exists a significant degree of *uncertainty* regarding the exact conditions under which the project will be executed. Uncertainty describes the degree to which it is difficult to predict any particular outcome before it happens. Factors that increase uncertainty of prediction in the project environment include, but are not limited to: the number or range of possible outcomes for a given element of the plan, the overall length of the project, the number of different resource types or organizational entities involved in delivering the project, and the level of interdependency that exists between the various activities of the project. In general, as these factors increase, the ability to predict with certainty how long a project will take or how much it will cost decreases significantly.

In order to make firm commitments regarding the outcome of a given project despite the existence of uncertainty, provision is made in the model to accommodate a reasonable amount of the unexpected. This provision is usually in the form of extra time and or budget above and beyond what it is thought would be required were it possible to know all there is to know about the project before its execution. In order to make realistic commitments in the face of uncertainty, project plans must contain some amount of safety or *contingency*. Simply, uncertainty is mitigated by contingency.

Adding unnecessary contingency, however, can significantly increase the expected time and money required for the project. Therefore, pressure exists to minimize the amount of contingency added to a project. As a result of this pressure, in order to ensure that there is enough protection in each project plan to make the related commitments achievable, project management practices have evolved to *disguise* the existence of apparently wasteful protection. Simplifying again, uncertainty requires contingency, but pressure promotes disguise.

The specific project management practices work as follows: the basic building block of the commitment model is the task or activity, which consumes time and resources, and produces an output usable by another task or activity. Explicit and implicit relationships exist between tasks or activities, which together comprise the overall model of project logistics. From this overall model, cycle time/due date and budget commitments can be derived.

In order to disguise the contingency required to make realistic commitments, as a practice enough contingency is embedded in *each* task or activity to ensure that its *chance of completing* on time and on budget are reasonably high. The amount of contingency each activity requires to make its performance highly reliable is *not* trivial. In general, it is safe to say that the more uncertain the environment and the more significant the consequence of missing the commitment, the more significant must be the amount of safety incorporated into the planning of each activity.

At this point the reader might ask; "If failed projects have a significant amount of safety or contingency embedded in them, why are they still failing?" According to Dr. Goldratt, projects fail to execute within the committed time and budget despite the fact that the commitments themselves contain significant contingencies because the existence of such contingencies leads to specific behaviors by members of the project team.

These consequently result in unintended negative consequences with respect to project performance. In a specific example of how presence of contingencies impacts project team decisions, the knowledge that more time and money is available than is strictly needed influences day-to-day decisions about how to get work done.

For instance, the existence of contingencies at the local or activity level provides each individual or organizational entity with a significant degree of flexibility in how it chooses to organize and execute tasks, which motivates certain managerial and resource behavior. This flexibility is the single most important tool a local manager has at his or her disposal to help the manager react to and deal with unplanned developments on a daily basis, and, as such, its very existence is a closely guarded secret. But in order to maintain the secret, local managers and resources have evolved a number of archetypical behaviors common to all project management environments. Because the negative effects of these behaviors only accrue to the project or projects in the form of unintended consequences of otherwise well reasoned local actions, however, one should not expect their presence in an organization or project to be easily detectible.

3. PROCRASTINATION OR STUDENT SYNDROME

One example of an archetypical behavior is procrastination, or Student Syndrome. To be precise, we design a contingency to address the arising of unprecedented factors. Nevertheless, as in the classic case of the student who waits until the semester is almost over to start working on the term paper, we all can remember the choice names reserved for the students who dared to turn in their assignments early at the same time as the rest of the class was asking for extensions. Procrastination is, as a result of this precise contingency, rampant in the workplace. Having more than enough time to do an assignment or task is reason enough to let time pass before investing any serious effort into its completion. When one adds to this the compounding fact that in the workplace there are often several other, more urgent work responsibilities to be addressed, it is then understandable that many tasks or activities are only executed when the level of urgency associated with them is sufficiently high to justify the effort required to accomplish them. Frequently, we discover aspects to the assignment that require more time than we had given ourselves at the end, which is the true damage of procrastination. In the end, when something goes wrong, we have, ironically, squandered the contingency that we had purposely built into the plan to save us from the unpredictable. An activity verses time chart of a bad case of the Student Syndrome is shown in **Figure 1**.

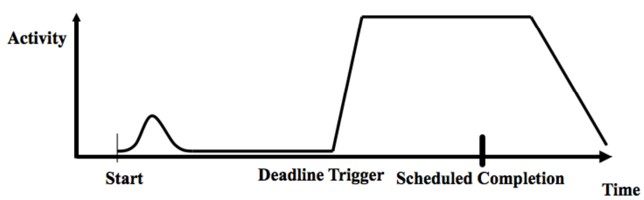


Figure 1. Student Syndrome

4. PARKINSON'S LAW AND FAILURE TO REPORT EARLY COMPLETIONS

Another behavior pertains to Parkinson's Law and the Failure to Report Early Completions, which hinges on the concept of unused contingencies. Specifically, not everyone procrastinates and even those who are famous for it don't always do so; there are those cases where tasks are started as planned. Now, given that most tasks for which there exists some degree of uncertainty are planned with a healthy amount of contingency, it stands to reason that many tasks will be, and are, completed in less time and use less resources than are actually allotted per-plan.

Expressed another way, in order to ensure that all possibilities are covered, tasks are scheduled with enough contingency to cover almost any bad thing that comes up. However, it is rare that all the possible bad things will happen on any specific task. Therefore, for the most part, in working on a task, the resource or organization will frequently find themselves with a completed task and an unused portion of time and resources. In some environments, this situation is seen for the good fortune it is, and efforts are taken to ensure that the project overall reaps the benefits of this good fortune every time it occurs.

However, unused contingency is often viewed negatively as a tell-tale sign of prior "sandbagging", the practice of knowingly asking for substantially more time and or budget than the job requires. Sandbagging is generally an inexcusable offense in the business world and in project management environments in particular. Additionally, it is frequently the case that other tasks will exhaust their allotted budget before they are satisfactorily completed. This represents an opportunity to apply left-over contingency funds, especially when a change order is difficult to justify.

For these reasons, there is usually a strong reluctance to report the existence of unused contingencies. The resulting behavior in project environments afflicted by these conditions is that tasks will appear to take exactly as much time and budget as was allocated, regardless of the degree of uncertainty that exists in the environment naturally.

Under these conditions, resources usually take as much time as is allocated to complete a task or activity, regardless of whether the time was strictly needed. This is usually justified under the guise of improving the output of the task, regardless of whether this improvement adds any material value to the project as a whole.

In some cases, the task is actually completed but is not reported as completed until the due date arrives or the funds have been expended. If the universal measure of task completion is the due date of a task, then there is a very good chance that tasks will never be reported as complete before the due date has passed.

The overall result is the appearance of high predictability in estimates: long and difficult planning cycles, as well as plans that become self-fulfilling prophecies of long cycle times and high project costs.

5. MULTI-TASKING

The shared resource multi-project model for managing projects produces a third problem, called multi-tasking. To be exact, in some project management organizations, resources are not dedicated to only one project in isolation. There is however a good reason for this. Given the nature of project logistics, it is frequently difficult to plan a project in such a way as to efficiently balance the load across all resources, in a manner that ensure all resources are productively engaged throughout its entire duration.

Therefore, for some resources in a project environment, there could be significant periods of planned downtime during which the resources must remain available to be called on if and when needed. But it is not considered a legitimate charge for a project to bear the cost of a resource that is available but not actually being used to work on a project.

Yet, someone has to pay for that availability. Industry uses several solutions. One such solution employed widely is to sacrifice availability in order to minimize cost. Another approach is to share resources across multiple projects to minimize the amount of time when a resource is available but not working on at least one paying project.

The shared resource multi-project model of project management can be found in almost every type of industry and project oriented organization. These organizations are universally driven to maximize the "utilization" of each and every resource, based on the assumption that, if all resources are operating at full or near full utilization, it therefore stands to reason that the organization is running at its full productive potential. [2]

Although the pursuit of high resource utilization as an end in itself may appear on the surface to be a good idea, upon further examination it can be shown to be the cause of a very wasteful practice that often creeps into project management organizations. That practice is called multi-tasking (not to be confused with cross-training resources so that they possess multiple skills). In its simplest form multi-tasking occurs when there is so much demand on a resource's time that the resource is forced to interrupt each task before completion, in order to work on another task. [1]

When the tasks involved are from different projects, the widespread practice of multi-tasking can result in significant delays to each project involved. But the negative effects of multi-tasking are nevertheless difficult to detect in project management organizations. On the contrary, multi-tasking has the effect of ensuring that resources appear to be in constant demand and therefore are satisfying the expectation that they be fully utilized.

To understand why multi-tasking is bad one must consider the effect on the cycle time of a project, each time a task or activity is interrupted. **Figure 2** illustrates the effect on the time to complete each task when the resource switches frequently between or among tasks. In the illustration, a single resource is multi-tasking among four identical tasks belonging to different projects or sub-projects in a multi-project / program environment.

In the non-multi-tasking approach, each task is worked from start to completion without interruption. This means that the task from the fourth project has to wait until after all three preceding tasks are completed before it is even started.

In many organizations, this situation represents an unacceptable state of affairs from the perspective of the project manager and or customer of the delayed project(s). Pressure is, thus, frequently brought to bear such that the resource is forced to show progress on all waiting tasks, even if it means delaying the completion of an earlier started task.

Any organization, for which the term "fire fighting" describes the normal mode of operations, can be assumed to suffer severely from the effects of multi-tasking. When applied to project management environments where early project completion is desirable, the reduction or elimination of multi-tasking can lead to dramatic reduction in cycle times and a corresponding increase in the number of projects completed in a given time period.

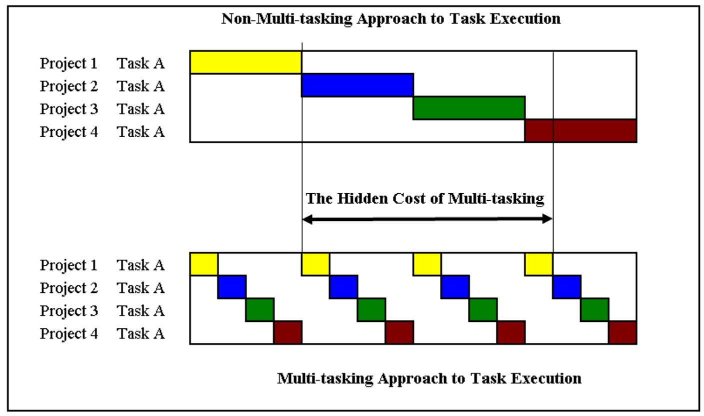


Figure 2. Multi-Tasking and its negative effect

6. THE CRITICAL CHAIN SOLUTION

1. Pipeline Balance or Elimination of Multi-Tasking

Pipeline Balance is one part of the Critical Chain Solution. Specifically, in order to ensure that resources are able to dedicate a full level of effort to each task to which they are assigned, the Critical Chain approach to project management insists on the elimination of multi-tasking to the greatest possible extent.

To eliminate multi-tasking, the Critical Chain Solution initiates numerous approaches.

- Resources are instructed to work each assignment to completion before starting a new one, and all other pending assignments are to be queued up in such a manner that any available resources with the minimum pre-requisite skills can be assigned as soon as they become available.

- A system of work prioritization is strictly enforced.

- To ensure that resources are not overwhelmed with a large backlog of unassigned work, projects are delayed in their start dates until resources are available to take on the new work.

- Excess capacity is maintained in the resource pool to absorb potential delays in work execution.

With these changes, work moves swiftly yet smoothly, resulting in shorter cycle time per individual project and the completion of more projects in a given period of time compared to the case in which multi-tasking is the normal mode of operations.

2. Critical Chain Scheduling or Strategic Aggregation of Contingency

Critical Chain Scheduling or Strategic Aggregation of Contingency is another part of the Critical Chain Solution.

- Project networks are rebuilt in a way that removes the entire disguised contingency from task or activity durations.

- Explicit contingency provisions strategically located at key points within the project network or project model replace the disguised contingency.

The result of this is project models that reflect a shorter overall cycle time while at the same time provide a higher degree of schedule and cost risk protection.

The following steps describe the Critical Chain planning process.

- Create a project network that reflects the logical dependencies between activities in a project.

- Resource requirements are then added (represented in the exhibit by different colors).

- Task durations are estimated according to the new rule of scheduling activities with no contingency embedded.

- The entire network is then scheduled to level the workload of each resource to fit within their stipulated capacity.

- The list of tasks comprising the "Resource Constrained Critical Path" (which is the *Critical Chain*), is then identified.

- All remaining tasks are assigned to one or more feeding paths.[1]

For an example, see the yellow outlined sequence of tasks in **Figure 3** and the yellow filled-in boxes in **Figure 4.** In **Figure 4**, the process is shown in actual software (in this case Aurora-CCPM) to better illustrate what a user will see in actual software. Most software looks similar to that shown in **Figure 4**, so the idea is to show the concepts with an illustration that is easier to understand and then complement this with an illustration of how the concept appears in actual software. At this point the network is a very aggressive representation of the schedule, as it has no provisions for uncertainty. The model at this point is therefore considered to be too high-risk to be used as the basis for committing to a schedule or budget.

The final step of creating the single project Critical Chain schedule is to:

Calculate and insert appropriately sized and located contingency buffers (feeding buffers and a project buffer, see below) designed to render the schedule practically immune to the normal levels of uncertainty one is likely to encounter during execution of the typical project plan.

An example of a fully protected Critical Chain schedule is illustrated in **Figure 5** and **Figure 6.**

A generously sized Project Buffer (PB) of approximately 50% more time (than the length of the Critical Chain) is added to the end of the Critical Chain while similarly sized Feeding Buffers (FB) are inserted at the point where each feeding path joins the Critical Chain. The schedule is always re-leveled post-buffering and cycle time commitments are made based on the entire schedule, including the buffers.

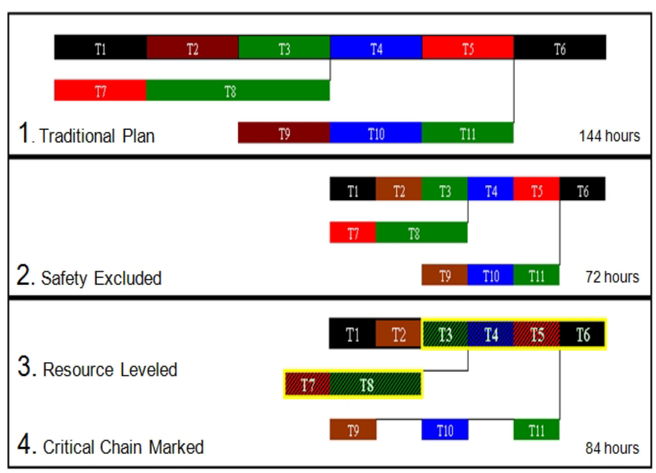


Figure 3. Going from a traditional schedule to a Critical Chain schedule without buffers

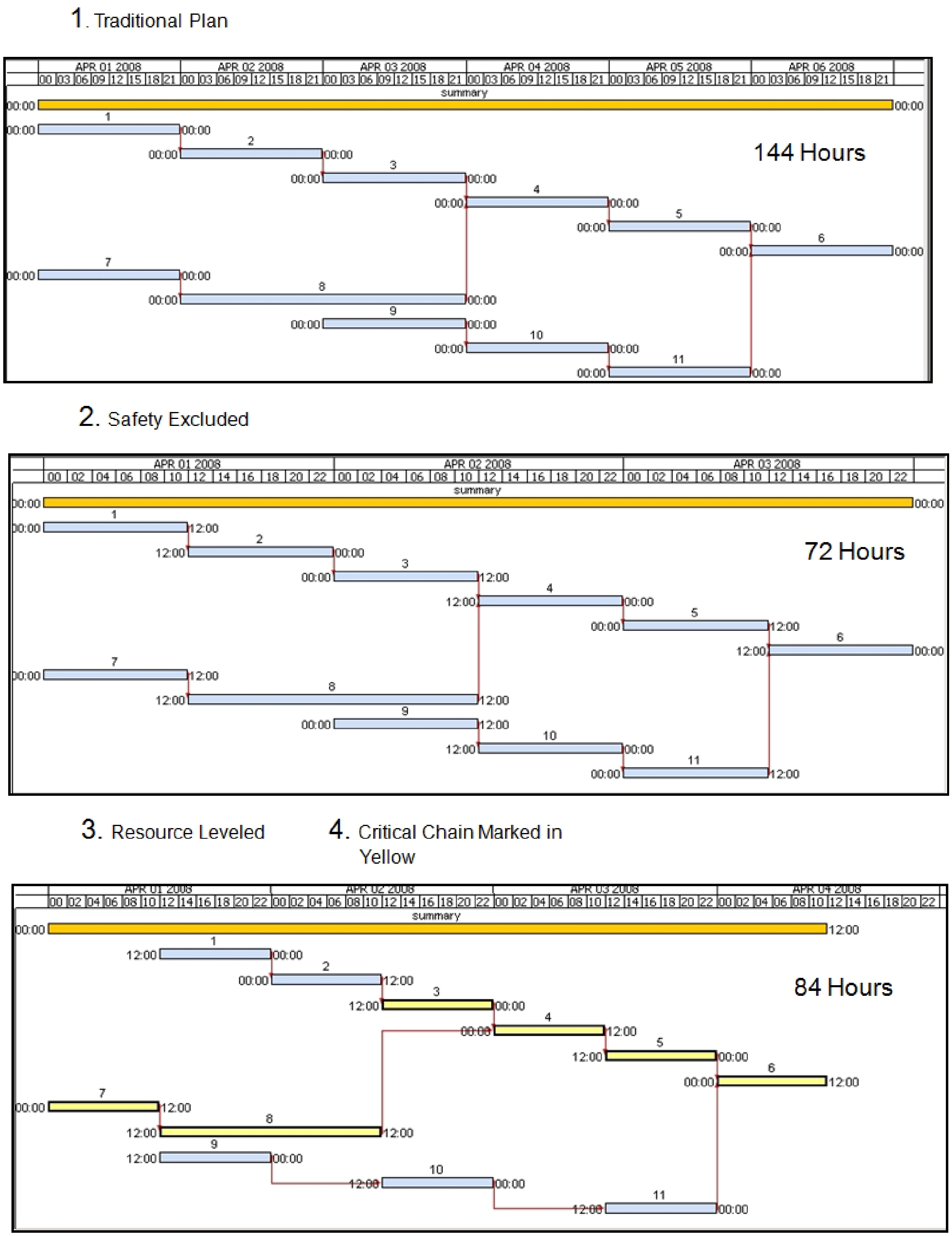
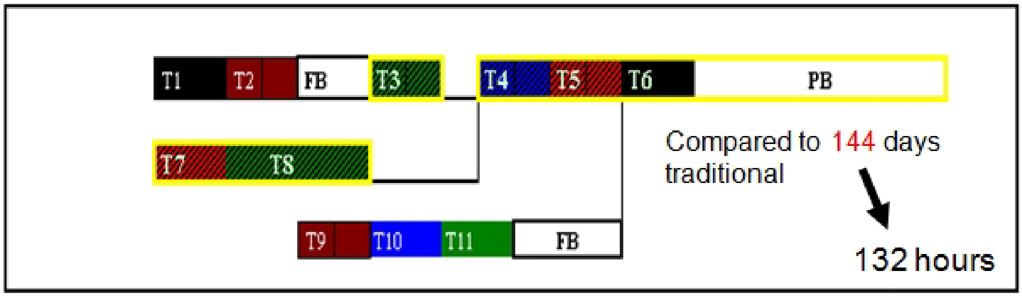
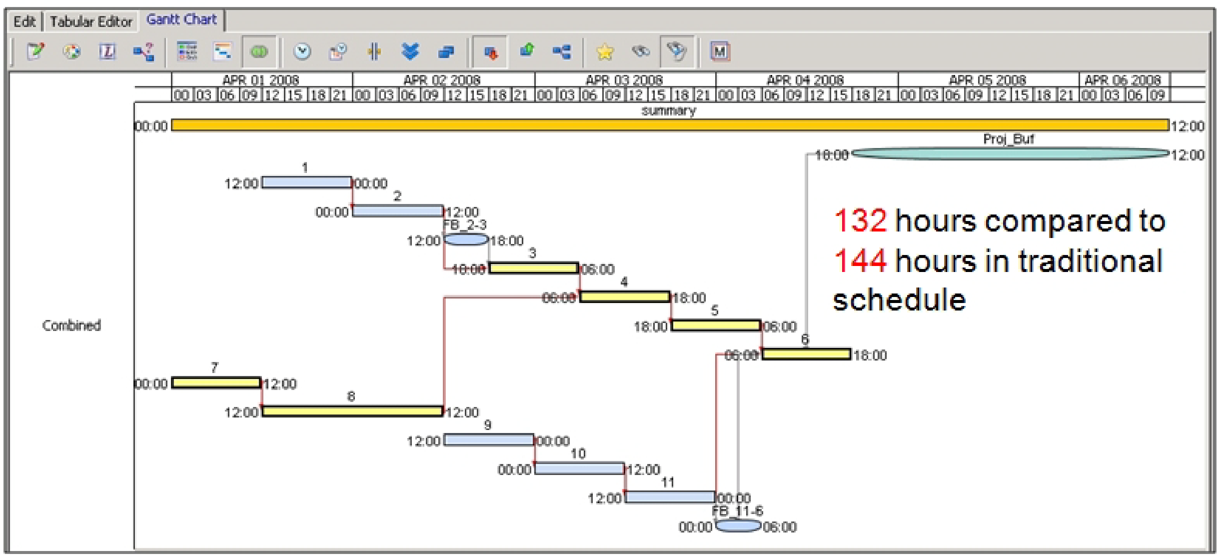


Figure 4. Going from a traditional schedule to an unprotected Critical Chain schedule in Critical Chain software



PB = Project Buffer   FB = Feeding Buffer

Figure 5. Fully Protected Critical Chain Schedule



Proj_Buf = Project Buffer   FB = Feeding Buffer

Figure 6. Buffered Schedule as shown in Critical Chain software

Note that the Critical Chain, unlike the Critical Path, will cross between logistical pathways as necessary to reflect resource constraints. All other factors being equal, when the 50% buffer sizing rule is used, the Critical Chain schedule is typically approximately 25% shorter than its resource leveled Critical Path equivalent but is substantially better-protected from uncertain events due to the explicit use of buffers as a means of containing schedule risk. For purposes of schedule stability, even though feeding paths may overwhelm their buffers and impact the Critical Chain, only under extreme circumstances is the Critical Chain ever allowed to be recalculated during project execution [1].

3. Buffer Management or Real Time Execution Management using CCPM

For organizations and projects that are sufficiently large and complex, a real time execution management system is another essential component of the Critical Chain Project Management solution. Buffer Management provides such an environment, with updated priorities that are consistently applied across the organization on an hourly, daily or weekly basis depending on the tempo of the decision-making cycle within the environment. In order to support decision-making effectiveness, a set of supporting practices has also been developed to accommodate the prioritization system.

The execution priorities are calculated based on the relationship [ratio] between the amount of buffer depleted and the remaining length of the Critical Chain. To compute these parameters it is essential that as each task is assigned and completed, feedback is provided regarding starts, completions and partial progress, in terms of *estimated time remaining* to complete each incomplete task. The traditional 'percent complete' by task is not used.

The Project Buffer is depleted as delays along the Critical Chain accumulate. The remaining length of the Critical Chain is computed as a percentage of its original length and the same is done for the project buffer. These percentages are further compared as a ratio where the larger the value, the better shape the project is in.

A value greater than 1.0 indicates that the rate of Project Buffer loss or delay accumulation is greater than the rate at which work is being completed along the Critical Chain. At any point during the execution of the project, this would be an indication that the risk of the project going late is substantial and that actions should be taken immediately to recover the schedule and therefore increase the buffer.

A simple *Fever Chart* is used to depict the current status of a project. A Red/Yellow/Green convention is used to depict the overall status of each project at regular intervals and a trend chart is used to project whether or not the project's status is changing for the better or for the worst. See **Figure 7** and **Figure 8** for an illustration of how schedule risk for a single project can be tracked over time.

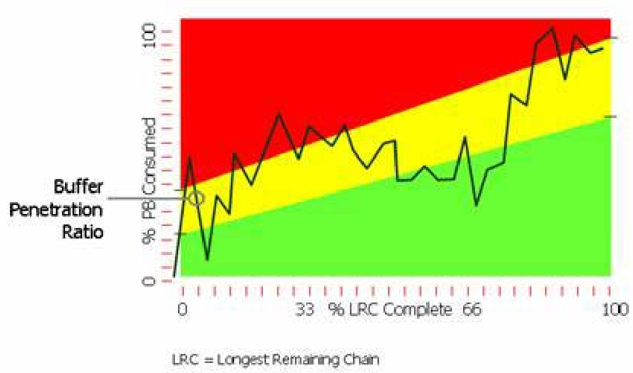


Figure 7. Fever Chart

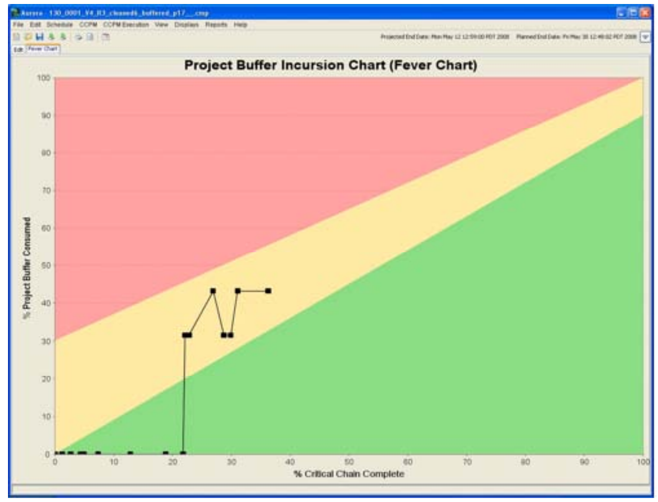


Figure 8. Fever Chart from a Critical Chain software package

Although the trend chart is only displayed at the project level, the data behind it is available for all buffers in a given Critical Chain project. Since all tasks feed at least one buffer, it is possible to compute for each task its current impact to all buffers, and therefore the relative risk it poses to the project as a whole. This information can then be used to establish the priority of every task in the project based on relative current risk to the project. When all projects in an organization have critical chain schedules, the result is the ability to have organization wide priorities for all project assignments at the task level. When this condition is in place, the organization can be operated as a highly synchronized and adaptive system, resulting in maximized effectiveness and speed of execution.

3. More Background Information & Software Options

Many new terms and concepts have been introduced; a good resource for definitions of these is provided at .focusedperformance.com/ccfaq.html. For example, the Critical Chain is: the resource-constrained critical path when using aggressive durations. The webpage referenced says the same thing, but essentially explains what a resource-constrained critical path is.

There are various software implementations of Critical Chain. Some of those available are listed below in alphabetical order.

- Aurora-CCPM by Stottler Henke, .StottlerHenke.com/Products/Aurora-CCPM
- CCPM+ by Advanced Projects, .advanced-projects.com.
- cc-Pulse & cc-MPulse by Spherical Angle, .sphericalangle.com.

- Concerto by Realization Technologies, .Realization.com.
- ProChain Project Scheduling by ProChain Solutions, Inc., .prochain.com.
- PS8 by Sciforma, .Sciforma.com, then use the Search function on the page for PS8.

Some of the above software runs as an extension to Microsoft Project. These include: CCPM+, cc-Pulse, Concerto & ProChain.

Unfortunately, there are not many reviews available on the different packages and there are even fewer comparisons. However, most Critical Chain implementations start with the recognition of the power and efficiency applying the Critical Chain Project Management will provide. Then the specifics of the implementation drive the software choice. That is, the benefits of deciding to apply Critical Chain far outweigh the benefits of any software choice, as long as the software supports all of the aspects required of the project. For example, Boeing utilizes various Critical Chain software packages ranging from PS8 on the low end to Aurora-CCPM for its most complex implementations; in all cases the decision to leverage Critical Chain is the decision that provides the greatest benefit.

7. Real-World Benefits/Improvements

There are many documented real-world examples that show the improvements possible via Critical Chain. A few are presented below.

Proctor & Gamble [4] prior to Critical Chain was completing circa three projects in a quarter with 25 projects in the system. After implementing Critical Chain, in less than a year they completed 8 projects in a quarter with 41 projects in the system. Improvements increased to completing 12+ projects per quarter.

Sood [5] presents a life sciences industry application where lead times were reduced from 8 to 12 weeks to typically 3 weeks, a reduction of around 70%. In addition, without using additional resources, around 50 studies were packaged every month, a throughput increase of 150%.

Kendall [6] presents examples, including Israeli Aircraft Maintenance Division, which cut the average aircraft wide-body conversion time from 3 months to 2 weeks. Seagate Technologies cut new product development times in half. Lord Corporation's I.T group went from completing 100% of their projects late to completing 85% early or on time. U.S. Marine Corps Naval Depot more than tripled workload completed using the same resources.

Various other success stories are documented at .tocinternational.com.

8. Conclusions

The Critical Chain Project Management solution minimizes the negative outcomes of the student syndrome and Parkinson's Law, eliminates multi-tasking, and reduces cost, time, and resource requirements. This solution allows Project Management to achieve its goal of making and keeping commitments while meeting the challenges of the project's complexity, interdependency, and/or uncertainty of the work environment. Using Critical Chain scheduling ensures improvements in all primary aspects of a project: schedule, resource, budget, scope, and quality, resulting in heightened returns for stakeholders.

These improvements have been demonstrated via numerous real-world applications ranging across many domains. These domains include:

aerospace,
automotive,
building & construction,
electronics,
engineering,
healthcare,
IT,
manufacturing,
media & publishing,
military,
pharmaceuticals,
sales & marketing,
software development, &
telecommunications.

Critical Chain-based Project Management provides a set of tools and processes to optimize the three separate dimensions of project management: 1) schedule or time, 2) resource or budget, and 3) scope, quality, or performance objectives.

References

[1] Goldratt, E. (1997, April). *Critical Chain.* Great Barrington, MA: North River Press Publishing Corporation.

[2] Rizzo, A. (2006, May) *Lean Enterprise Project Management.* Product Development Institute Train-The-Trainer Program, Everett, WA, USA.

[3] The British Computer Society (no date). The Holy Grail of project management success. Retrieved on February 15th, 2007 from ://www.bcs.org/server.php?show=ConWebDoc.8418.

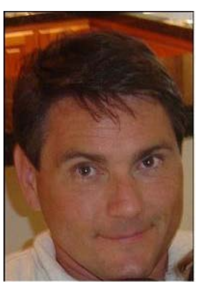
[4] Smith, Michelle. CCPM – A Sustaining Strategy at Procter & Gamble Pharmaceuticals, ://www.pharmpro.com/ShowPR.aspx?PUBCODE=021&ACCT=0000100&ISSUE=0903&RELTYPE=PR&ORIGRELTYPE=ATO&PRODCODE=0000&PRODLETT=UI&CommonCount=0

[5] Sood, Shailesh. Critical Chain Buffering: A breakthrough approach that makes R&D workflow run like a project factory. .tocc.com/Articles/SoodArticle.PDF

[6] Kendall, Gerald I., Critical Chain and Critical Path – What's the Difference?, www.tocinternational.com/pdf/Critical%20Path%20vs.%20Critical%20Chain.pdf

BIOGRAPHIES

Hilbert Robinson has fifteen years of experience in industrial engineering and project management, including ten years of work applying Theory of Constraints (TOC) and Critical Chain Project Management (CCPM) techniques to aerospace and military problems. Mr. Robinson is a member of the Theory of Constraints International Certification Organization (TOCICO) Critical Chain Certification Committee. He is a recognized expert in existing CCPM software tools including Boeing's proprietary TimePiece™ finite capacity modeling software, as well as PS8 and Concerto. He has worked on development of Stottler Henke's Aurora scheduler and its own CCPM implementation.



Robert Richards, Ph.D. is a Principal Investigator and Project Manager at Stottler Henke. Prior to joining Stottler Henke in 1999, Dr. Richards's work included Mechanical Engineering and Mechanical Process Engineering at the Stanford Linear Accelerator Center, this work included project scheduling/management and the improvement of the mechanical engineering process. Since joining Stottler Henke projects have dealt with training system development, decision support and critical chain project management, including much of the design work for the short-duration-task CCPM capabilities of Stottler Henke's Aurora-CCPM product