Cutting Edge

INTRODUCTION TO CRITICAL CHAIN PART III
MULTITASKING, RESOURCE LEVELLING AND MULTI-PROJECT SCENARIOS

BY
MIKE MANNION AND SVEN EHRKE

THE IMPACT OF MULTITASKING

Remarkable as it may seem, traditional project management regularly fails to include explicit account of resource conflicts in its schedules. It is extremely common, for example, to find workers implicitly assigned to tasks on two, three or more projects simultaneously. Let us consider the implications of this.

Figure 1 shows schedules for two high-priority projects, which have been constructed on the implicit assumption that there are no resource restrictions, and hence no resource conflicts. Assuming each task takes 15 days to perform both projects will complete in around 45 days.

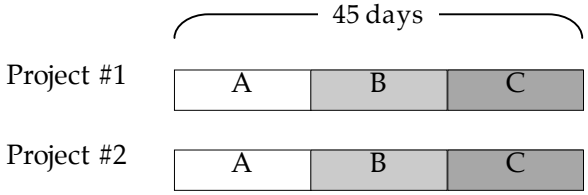

Figure 1 Two project schedules, no resource conflicts

Now, in this organisation there are *three* distinct resources, which are specialists in tasks A, B and C respectively. However, since it is not possible to literally work on two things simultaneously, the resources have to switch from one project to another – to *multitask*. The outcome is shown in Figure 2.

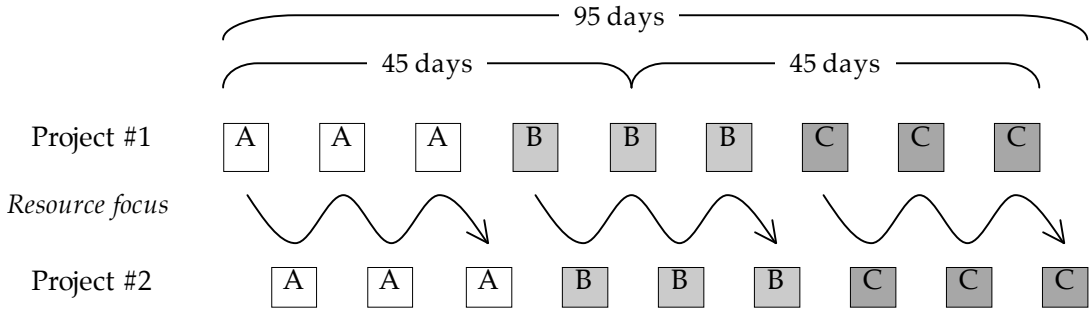

Figure 2 Two project schedules, limited resource, multitasking

Firstly, notice how the duration of each project has grown significantly. Project #1 now takes 90 days and Project #2 slightly longer. Secondly, it would probably be correct to assume that the picture is actually *overoptimistic*. Real people do not operate like computers. When a person switches from one task to another he requires a degree of ramp-up time. This can be anything from a few minutes to several hours depending on the nature of the task and the personality.

The typical reaction from project managers when a project appears to be taking twice as long as originally envisaged is to cry "I need more resources!" Under these circumstances project workers, too, find more substance with which to justify their pessimistic task duration estimates (see Part II on *behavioural realities*).

In an alternative arrangement, shown in Figure 3, we have staggered the starts of the projects until resources are available to work on them. Consequently, Project #1 is delivered in 45 days and Project #2 in 60 days. Without the addition of any new resources, both projects are delivered significantly earlier than in the previous setup!

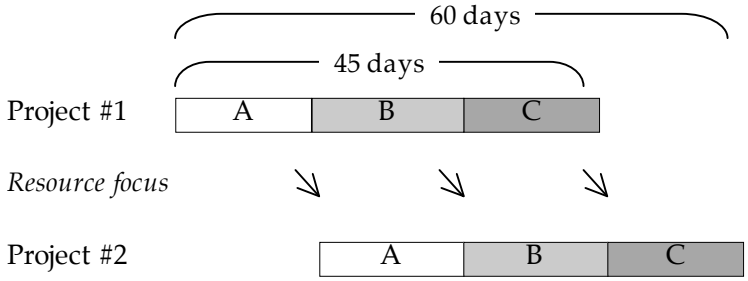

Figure 3 Two project schedules, limited resource, no multitasking

Yet this is not the only advantage: Firstly, resources are *fully concentrated* on the tasks they must perform. No ramp-up penalty is incurred because of context switching. Quality, too, will be positively affected by this. Secondly, from a financial perspective, work-in-process is *significantly* reduced – less capital per project is tied up in the system. Return on Investment is up. Thirdly, again highly significant in fast-moving environments like software development, requirements have less time to change, which means less work will have to be redone. Fourthly, the principal cause of student syndrome has been quashed because a worker, once assigned to a task, views this task as *the* number one priority. And finally, project workers who are not required to multitask sense (correctly) that they actually have a realistic chance of delivering a result according to the original estimate. And given that context switching is no longer necessary means, two major sources of stress have also been dealt with. The result is a happier, healthier staff.

All this adds up to an almost outrageous increase in productivity, which is achieved *simply by scheduling tasks for when resources are available*. Ironically, executives generally perceive multitasking as something positive, because it appears that resources are being fully utilised. They are following the gross misnomer that maximising utilisation is a good thing (see Part I).

RESOURCE LEVELLING

We have now seen that a *description of task dependencies* alone does not constitute a plan. A schedule based on critical chain can only be finalised once resources have been assigned to tasks. And only once this happens can we assign task start dates. Figure 4 presents excerpts from two projects *prior* to resource assignment.

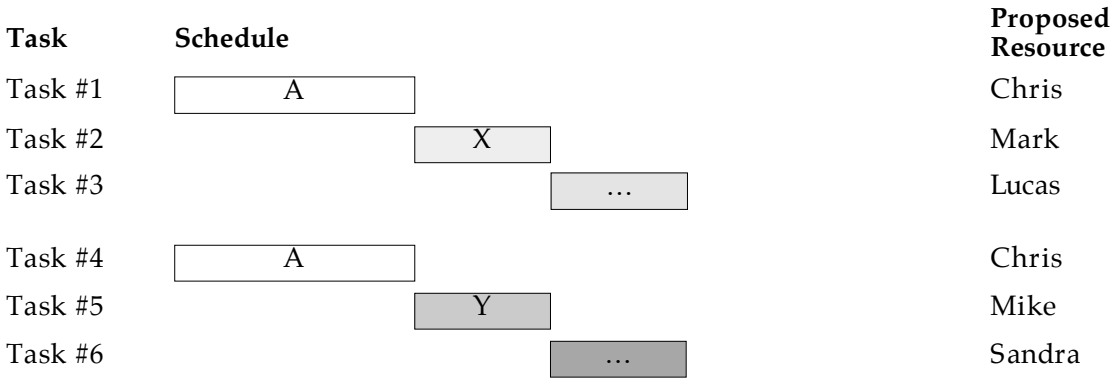

Figure 4: A single-project schedule with a proposal for resource assignments

Clearly there is a resource conflict. Chris will be unable to work on tasks 1 and 4 simultaneously. He could, of course, multitask, but we have seen where this leads us. So let us adjust the schedule to take this into account.

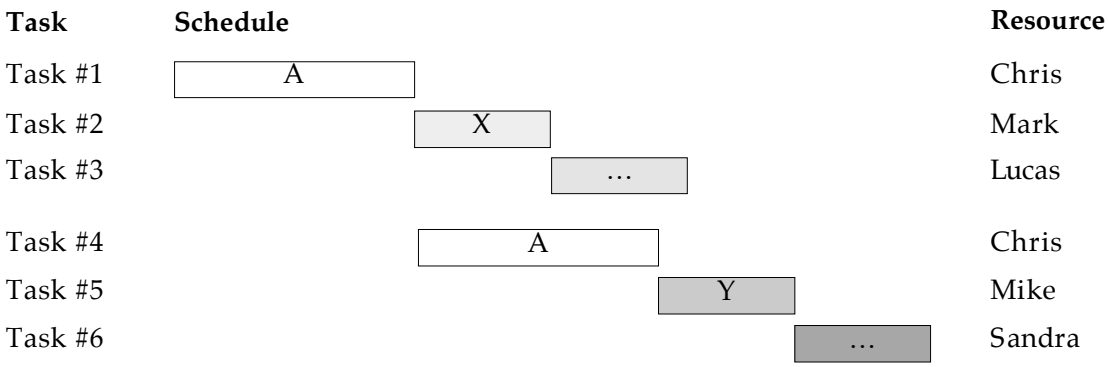

Figure 5: A single-project schedule after resource-levelling

Figure 5 demonstrates the result of this process, which is called *resource-levelling*. Although there is no algorithm for determining an "optimal" resource-levelled schedule, project management software can assist in this process (MS Project supports automatic resource levelling, for example).

RESOURCE BUFFERING

In critical chain project management, task start dates are highly flexible in nature. Flexible start dates are necessary because of variation: some tasks will be delivered earlier, some later than originally estimated.

So although resource-levelling removes conflicts from the plan, conflicts can still occur because of variation. There is, of course, no way to predict in advance when this will occur. But what we can do, at least, is build in a warning for the PM that a *potential* resource conflict exists – one which will cause delay if it occurs.

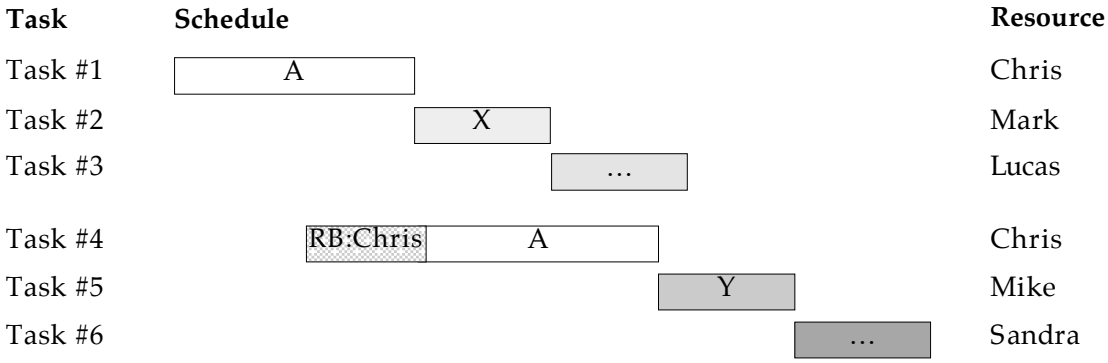

Figure 6: A resource buffer warns the PM of a potential resource conflict

Just such a warning is shown in Figure 6. A *resource buffer* entitled "RB:Chris", which is specific to the resource "Chris", indicates that should a delay occur in task #1, he will not be able to begin task #4. In this way the PM knows precisely when she should inquire about the status of task #1. If task #1 looks like it is going to exceed the original estimate, the appropriate action can be taken.

MULTI-PROJECT CRITICAL CHAIN

What we have learned so far greatly increases our chances of delivering individual projects in a competitive and timely manner. Thousands of projects around the world already bear testament to this. However, for the technique to work in a multi-project environment, the *program manager* must take resources into consideration, which are *shared across projects*.

Fortunately, it is not necessary to consider each and every possible resource conflict, which in any case would be an impractical undertaking in most organisations. Instead it is the job of the program manager (or project management office) to identify the one shared resource, which has the smallest capacity compared to other shared resources. This resource is termed the *capacity-constraint resource* (CCR), as it determines the number of projects which can be executed by the organisation in a given timeframe.

Once the CCR has been identified a centralised schedule for its usage is drawn up based on project prioritisation. Typically the highest priority project gets to use it first, the second highest second etc. Individual project plans are developed around to the availability of the CCR. From here on the capacity

constraint is termed *drum resource* because it determines the rhythm – the beat – with which the organisation executes projects. A simple example of this is presented in Figure 7. In this case the CCR can serve only one project at a time and projects 1, 2 and 3 and staggered in respect of this.

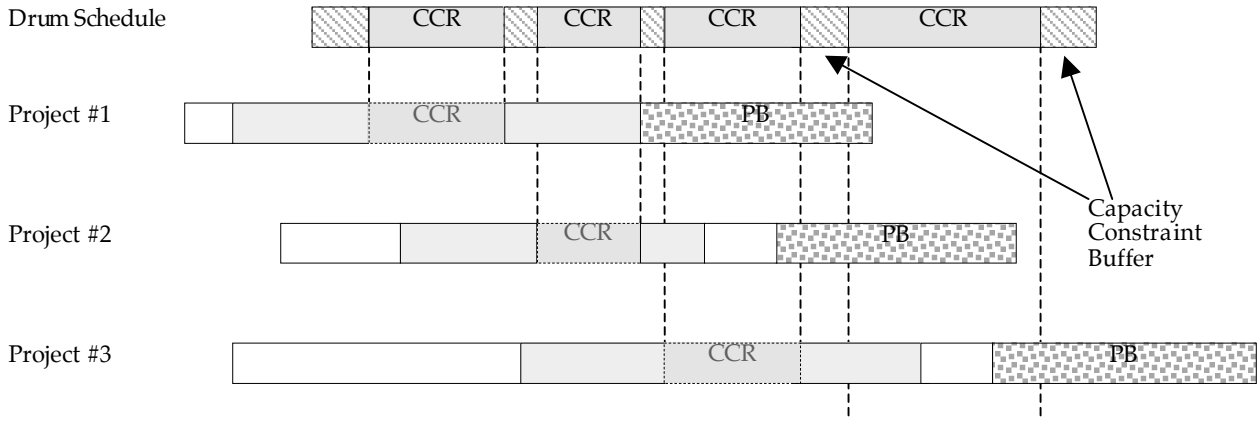

Figure 7: A Multi-Project Critical Chain Scenario

Since the drum resource is – by definition – a constraint, we can apply all the lessons of TOC which we encountered in parts I and II. To obtain the highest possible throughput of projects in an organisation over time, the drum resource must be kept constantly busy. And whilst critical chains are protected from common-cause variation using project, feeding and resource buffers, the drum is protected from common-cause variation by means of a *capacity-constraint buffer* (also shown in Figure 7). In addition, risk management must ensure that the drum is not subject to outages because of special causes. To increase the number of projects which can be performed by the organisation over time, we turn to the five focussing steps. If we succeed in elevating the drum resource so it is no longer a capacity constraint, the next capacity constraint acquires the role of drum, and individual project plans are rescheduled according to this resource's availability.

Behavioural Realities (again)

If multitasking is so counter-productive for the organisation, how come it is such a common phenomenon? When we last addressed the issue of behaviour, we saw that individuals motivated to protect themselves from variation (local optimisation) ended up causing the entire schedule to expand, which in turn obliged the PM to override her workers' estimates etc. The solution to this problem was to encourage a *collectively beneficial behaviour* which was geared to achieving a common goal – that of meeting the project delivery date.

In the multi-project scenario, executives are typically motivated to ensure that the projects for which they are responsible get done and get done as soon as possible. From the point of view of the individual executive, he usually tries to ensure that his project gets to start sooner rather than later. Consequently, multiple projects are started in parallel, and as long as everybody is obliged to *demonstrate progress* on each of the projects, project workers end up multitasking. And as we have already mentioned, many people view this positively.

Yet if multitasking were not already devastating enough for productivity, we encounter *even more* delay because of all the projects simultaneously calling upon, and subsequently overloading the CCR. Unfortunately, failing to identify the CCR (or any other constraint for that matter) does not mean this problem goes away.

Is it any wonder, then, that executives end up battling – sometimes furiously – for resources? Or that "task-forces" are created to ensure that – at the very least – the highest priority projects get delivered more or less on time? As for projects not privileged enough to receive the attention of a task-force, these are frequently forced to deliver a result regardless, which is achieved either by compromising the original specification (be that product features or quality), or by demanding overtime from project staff (and typically both). All remaining projects simply get shifted into the unforeseeable future – or are simply cancelled. According to Standish [Sta04], 66% of all projects examined in their most recent study fell into these latter two categories.

Breaking the Behavioural Deadlock (again)

What can be done to break this self-perpetuating cycle?

Again, a new, *collectively beneficial behaviour* is required. The new behaviour must be enacted at the executive level and must drive the organisation towards a common goal: That of maximising project throughput, which in turn implies *ensuring that optimal use is made of the capacity constraint resource*.

To achieve this, executives must understand that it is in everyone's best interest to ensure that the CCR is neither underutilised, nor overloaded. Although counter-intuitive at first, by eliminating multitasking, staggering project starts and introducing capacity-constraint buffers to schedule, the number of projects that can be completed by an organisation in a given timeframe increases dramatically. The reader should not now be surprised to learn that the quality of the results is also significantly improved.

PRACTICING AND IMPLEMENTING CRITICAL CHAIN PROJECT MANAGEMENT

We shall complete our brief introduction to CCPM with a few remarks on what it takes to implement it in the real world.

Note firstly the difference between *practicing* critical chain and *implementing* critical chain. *Practicing* critical chain means successfully running individual projects according to critical chain concepts regardless of how the remainder of the organisation runs (or fails to run) its projects. *Implementing* critical chain means making it the daily business of everybody in the organisation, from executives to project staff.

Because of the challenge associated with implementing critical chain on a large scale, many organisations demand that a pilot project is performed first. Success of the pilot is of course by no means guaranteed. The greatest challenge facing a would-be critical chain project manager is to sufficiently isolate the project's resources from other mandates. Unfortunately, staff usually feel that multitasking is expected of them, and many even enjoy it. Also, if staff come from other line organisations, their bosses may be reluctant to "let their resources go" for block periods, even if these are comparatively short. It is therefore vital that everybody is made aware of the implications beforehand – and that everybody is reminded of the implications as the pilot project proceeds.

We now list the biggest obstacles facing a full-blown critical chain implementation:

- *Addressing and changing the organisation's culture*. Without doubt, the biggest challenge facing the adoption of critical chain is an organisation's existing culture. For critical chain to work, collectively beneficial behaviour at both the team and the executive level needs to be established. Unfortunately, many of those executives who would be required to support or sponsor a critical chain initiative have acquired their positions through mastery of existing strategies (see *Part I: Process Efficiency*). The desire to retain the status quo is therefore large.
- *Measurements and rewards*. These are developed over time as a means to reinforce an organisation's strategies. Unfortunately TOC and critical chain raise very serious questions for the operational strategy of most of today's organisations, which implies a complete rethink of existing incentives. Again, many employees are accustomed to being rewarded for what is essentially dysfunctional behaviour. Convincing them that a new behaviour is required is a major challenge.
- *Education*. In management today there is an enormous gap in knowledge concerning the nature of *the process*. Before critical chain can be widely established, executives need to have grasped the concepts of constraints, elementary statistics, variation, buffering, the effect of multitasking, and how dysfunctional behaviour and strategy impact process efficiency.
- "Not Invented Here" or "It's obvious" syndrome. This is simply the issue of human pride – particularly prevalent among successful executives.

Combating these obstacles requires systematic approach, a plan and patience. Various authors provide strategies for addressing these and other challenges, in particular [Lea05], [Ken03].

SUMMARY OF PART III

- Most project environments fail to take resource contention into account when developing schedules. Not doing so indirectly obliges project participants to multitask. This has a *devastating* impact on productivity.
- A sequenced list of tasks does not constitute a plan. A realistic plan *must* take resource contention into account.
- Resource contention can be eliminated by staggering activities within projects which depend on the same resource. Eliminating resource conflicts from a plan by adjusting task commencement is called *resource levelling*.

- Although resource levelling reduces the chances of resource conflict, conflicts can still occur because of variation. A *resource buffer* is used to forewarn the project manager to the possibility of this occurring.
- In multi-project scenarios, the project program manager must identify the capacity-constraint resource (CCR), designate this as a drum resource, and build individual project schedules around this.
- Executive behaviour is typically responsible for starting too many projects in parallel. The result is yet more multitasking and an overloaded CCR.
- Successfully piloting a critical chain project prerequisite making clear and repeated statements concerning the implications for resources.
- Successful critical chain implementations prerequisite the addressing of cultural, strategic, educational and human issues.

For more information, please contact: Elke Gemperlé (Email: elkegemperle@cuttingedge.ch)

For book recommendations, references and links, check out http://www.cuttingedge.ch/resources

REFERENCES

[Lea05] L. P. Leach, *Critical Chain Project Management, Second Edition*, Artech House, 2005

[Ken03] G. I. Kendall & S. C. Rollins, *Advanced Project Portfolio Management and the PMO*, J. Ross, 2003

[Sta04] Standish Group, http://www.standishgroup.com/press/article.php?id=2 (last accessed November 2004)