Critical chain

The critical chain technique was developed by Dr. Eliyahu Goldratt, and first published in his 1997 business novel, Critical Chain[1]. It is an approach to scheduling and managing the execution of projects. It consists of a set of formal techniques and associated work and management practices.

Critical chain was originally developed as a method for managing a single project, and then expanded in scope to cover portfolios. Proponents claim it use brings improved project performance and that it isn't 'just' an alternative technique. This document will not discuss the pros and cons of critical chain, and limits its scope to describing the approach.

Underpinning critical chain is the goal of maximising the flow of project work and minimising interruptions and impediments to flow. It embraces the inherent uncertainty of estimates and focuses on facilitating the right management interventions during execution (and preventing the wrong interventions) more than trying to develop a perfect plan.

A: Scheduling

CC scheduling builds on the principles of critical path analysis and resource limited scheduling to identify chains of activities that are constrained by both dependencies and resource availability. Importantly, the technique then goes on to take aspects of human nature into account.

The 'critical chain' itself is "*the longest sequence of dependent events through a project network considering both task and resource dependencies in completing the project.*" (TOCICO, 2012).

On projects with constrained resources the critical chain may not be the same as the critical path (which is based on only task dependencies). This is why the term 'critical chain' was introduced[2]. It is not that the CPM doesn't consider resources, but that practices evolved such that many CPM schedules ignored them at least in the earlier schedules. CC requires tasks to have resources allocated.

There are other differences between a critical chain schedule and one using critical path.

Most CP schedules are built from tasks estimated using highly reliable durations. This means time allowances for uncertainty are built into individual task estimates. In execution, task estimates are usually treated as a commitment and task managers are expected to meet them.[3]

---

[1] Goldratt, E M (1997). Critical Chain. North River Press, Great Barrington, MA.

[2] Although Goldratt used the term critical chain to differentiate it from the critical path, he was not the first to identify this difference. Jerome Wiest one of the critical path method pioneers used the term "critical sequence" for the same idea, some 30 years before Goldratt (Wiest JD. (1964). Some Properties of Schedules for Large Projects with Limited Resources. Operations research, 1964, Vol.12 (3), p.395-418).

[3] Whilst the PERT variant of CPM considers the distribution of task durations using a three-point estimate, this is usually used to calculate a single task duration from a weighted average of the three values. The key

If each task estimate includes task-level safety, Goldratt suggested that this is far more than is needed to protect the overall project.

If a project schedule has so much built-in safety, it would be reasonable to expect that it would be met every time but Goldratt identified three reasons why this doesn't happen.

- *The student syndrome*: because people know there is built in safety, they wait until the last minute to start the work. Then any interruption to the work or unexpected issue causes the task to be late.
- *Bad Multitasking*: when people have multiple activities on multiple projects to perform, they attempt to work on them simultaneously and this creates inefficiencies. It also fuels student syndrome – the task furthest away is put off because of the other current work.
- *Dependencies between activities*: delays accumulate faster than advances, i.e. if an activity finishes late then the following activity will be delayed but if an activity finishes early the following activity is less likely to be started early. This too is made worse by control/management conventions that see an early task completion as bad ("You must have padded your estimates. I will cut your other estimates"). This can inhibit tasks which are complete ahead of the estimate from passing their deliverables to the next task. This leads to early completions being lost to the project, whereas a late task cannot be hidden, so negatively impacts project.

Many people add Parkinson's Law to these three, i.e. the work will fill the time available to do it. This encourages bad multitasking because resources feel they have more than enough time to do a task, so they can start another in order to increase their productivity and believing that starting tasks early will be beneficial to the project (another wrong assumption according to Goldratt).

In simple terms, critical chain addresses these issues by applying the following principles in scheduling:

- The start point is the same as with the critical path method – a dependency network based on logical dependencies between tasks.
- Estimate activity durations for the tasks at a 50/50 chance of completion. Also termed an 'optimistic' duration – the likely task duration assuming focused (100% allocation) work with no interruptions, delays or rework. This is similar to the optimistic duration used in PERT.
- A time-based schedule is developed based on (1) logical dependencies and (2) ensuring no resource is given more than one task at any time[4].
- Identify the critical chain.
- Time allowances for uncertainty are added to the schedule at strategic points. They protect the project completion date because the individual task estimates are based on optimistic durations, and it is extremely unlikely that all tasks wil achieve their optimistic duration. These time allowances are known as 'buffers'. Whilst buffers can be sized equal to the total amount of safety removed from the individual task estimates in changing from highly reliable to optimistic durations, experience has shown that buffers of half the total removed task safety are usually sufficient. This is shown in the worked example below.

---

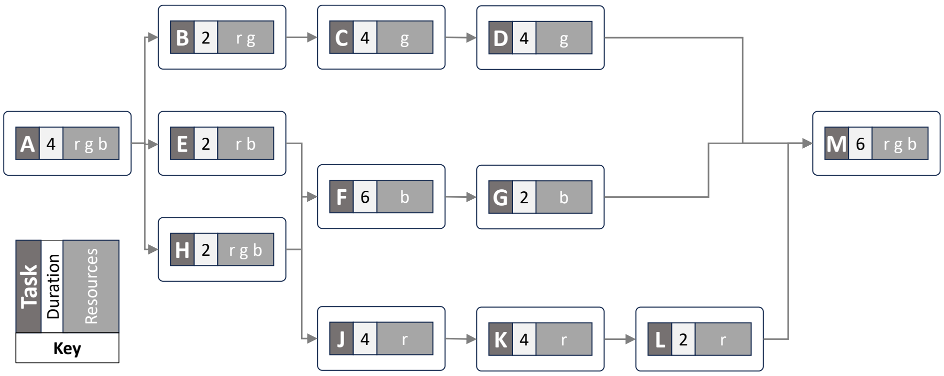
difference with critical chain is not in estimating it is in execution, and in particular whether a task estimate is seen as a commitment.

[4] This document presents a simplified view, assuming 'resource' is a single person. Where resource groups have many staff, the schedule will be based on a task minimum team size.

- A project buffer is added to the ‘raw’ critical chain, and feeding buffers are added at all the integration points where non-critical paths join the critical chain. .

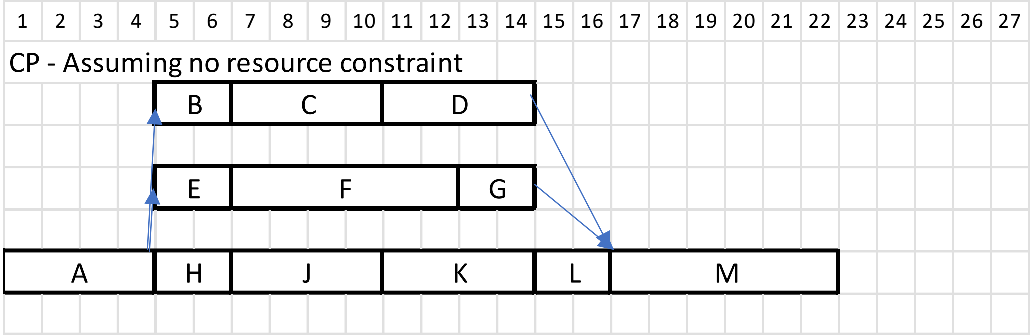
A worked example comparing critical path and critical chain schedules.:

1. We will use this simple project network



This project uses three different resources r, g and b. Each task’s duration assumes all the available resource is used to deliver the task, and is a highly-reliable duration.

2. A CP Gantt form with no account taken of resource availability.

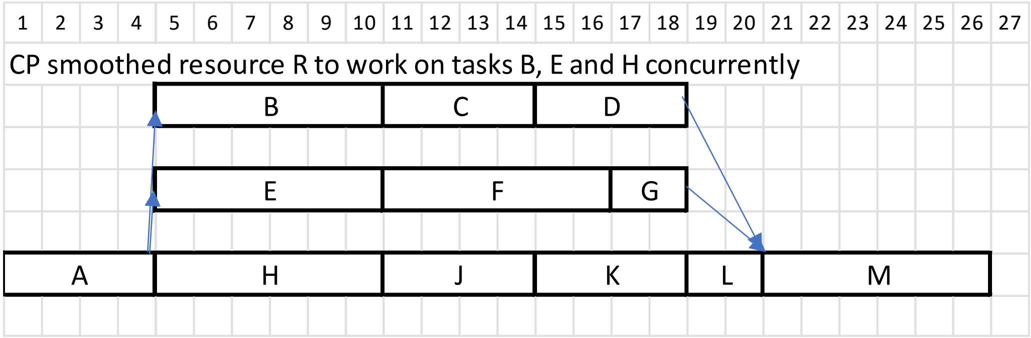


Duration = 22.

CP = A-H-J-K-L-M.

3. A CP Gantt form with resource r shared across tasks B, E and H running concurrently.

Given tasks B E and H all require full time input from resource r to complete in 2 time units, a resource-smoothed CP might extend B, E and H to 6 units long, allocating 33% of r to each.

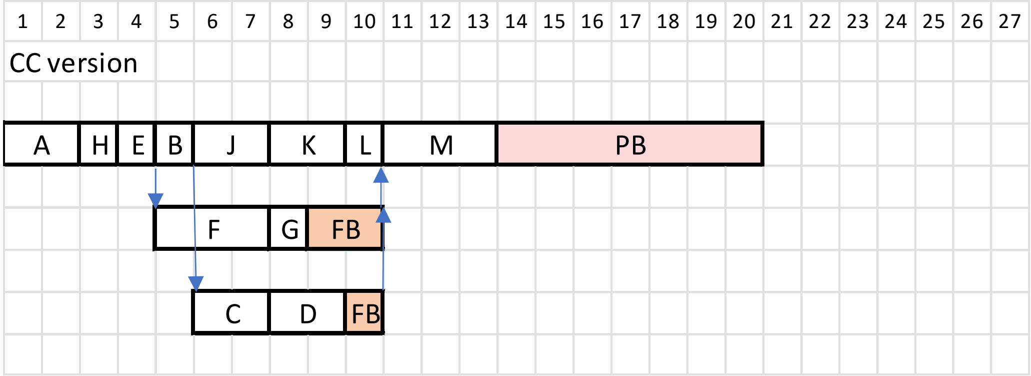


The CP is the same sequence but is now 26 units long. As before the non-critical paths of B-C-D and E-F-H both have 2-days of slack/float.

Note that sharing resource across parallel tasks usually incurs a productivity reduction due to the task switching. No productivity loss has been factored into the above Gantt.

4. A CC version of this project.

The Critical Chain version would not schedule B, E, and H in parallel, so might look like:



The task durations have been reduced to be ‘optimistic’ (in this case using the rule-of-thumb that the optimistic duration is half the highly reliable durations).

The CC is A-H-E-B-J-K-L-M, and the ‘raw critical chain’ is 13 units long. A project buffer of 7 units has been added to protect the whole project. This is sized using a CC rule-of-thumb that the buffer is 50% of the duration of the ‘upstream’ chain it protects.

The buffered critical chain results in a planned project duration of 20 days.

There are two feeding chains, F-G and C-D. The integration points where these join the CC are each protected by a feeding buffer. This is to protect the critical chain from the risk that feeding chain tasks take longer than the optimistic duration. As with the project buffer, this would normally be sized at 50% of the protected chain. In this case the buffer protecting C-D is only 25%. This indicates that C-D is close to being critical. In effect half of its nominal feeding buffer is already consumed when the project begins.

In the above illustration it was assumed that a task’s optimistic duration is half the same task’s highly reliable duration. This is a CC rule-of-thumb that has been shown by experience to be a reasonable estimate. It is a convenient and fast way to convert an existing critical path schedule into a critical chain one.

The other CC estimating technique is to use ground-up estimates and ask the task experts to provide an optimistic duration. For example by asking "*How long would the task take if you were allowed to work exclusively on it without interruption, where you had everything you need available before you begin, everything goes smoothly, and you are not being assessed on achieving your estimate?*" Using this method needs the task manager to be confident that the optimistic estimate will not be taken as a commitment or used in their performance measure during execution.

Another feature of a critical chain schedule is that it is relatively high level. The project schedule does not aim to include every detailed task necessary for a successful project. Much of the detail is left for the work package and task managers to plan and manage in a way that suits the work. A typically critical chain schedule will consist of some 50-500 logically-linked tasks.

Another CC rule-of-thumb is that a task that is included in the project CC schedule should be between 1-5% of the project duration in length. Clearly some iteration and intuition is needed since the initial schedule is prepared before the total duration is known.

In this way critical chain has been used in conjunction with other techniques popular for managing the detailed work. These include so-called agile methods and LastPlanner (Lean Construction), which are used to plan and manage short-term work and roll-up into tasks that for the project critical chain schedule.

Project control then focuses on the management of the buffers rather than the individual activities. It is during execution that the critical chain method helps to deliver projects in less[5] time and to high reliability. This is through a combination of

1. critical chain indicators of progress that help identify when an intervention or change is needed, and when it isn't.
2. an emphasis on important management and team member behaviours that help maximise the rate of project work.

B: Execution.

During execution, the status of the project and feeding buffers provides key management information. This 'status' is based on the *buffer burn-rate (BBR)* – the amount of buffer used relative to progress.

BBR = (% Buffer consumed) / (% Critical Chain Progress)

Buffer is used when a task takes longer (or expects to take longer) than the optimistic. If a task takes less, then the buffer increases.

---

[5] Less compared to conventional critical path schedules. As can be seen from the worked examples, the CC duration, at 20-days, is 23% shorter than the equivalent CP duration of 26 days.

On average, buffer is expected to be used in proportion to progress, as shown by the straight line in Figure X.

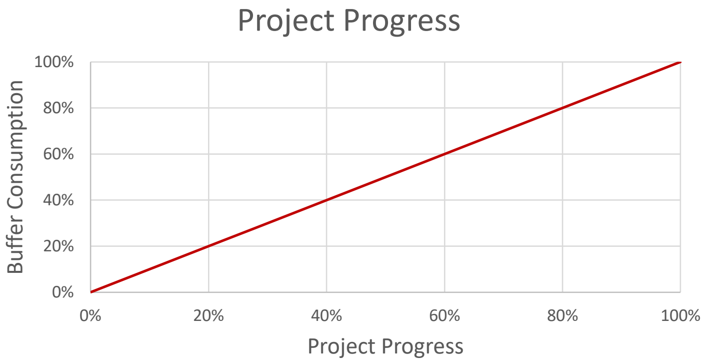


Figure X: Idealised use of buffer relative to project progress. BBR = 1

This underlines that buffer is an integral part of the project duration. It is not fat or padding, and it is expected to be used.

Due to the inherent uncertainty in project estimates, Critical Chain does not expect a project to follow this BBR=1 baseline exactly (red line in Figure X). In order to avoid over reacting to variation from this baseline, CC draws a tolerance band around the BBR=1 baseline . 'Normal' project progress will take place inside this band and this would not trigger management or control interventions. If progress goes outside the control band then management interventions would be expected, especially in the red zone.

The bands/regions of the graph are normally coloured red-yellow-green and the graph is called a fever chart, as shown in Figure X.

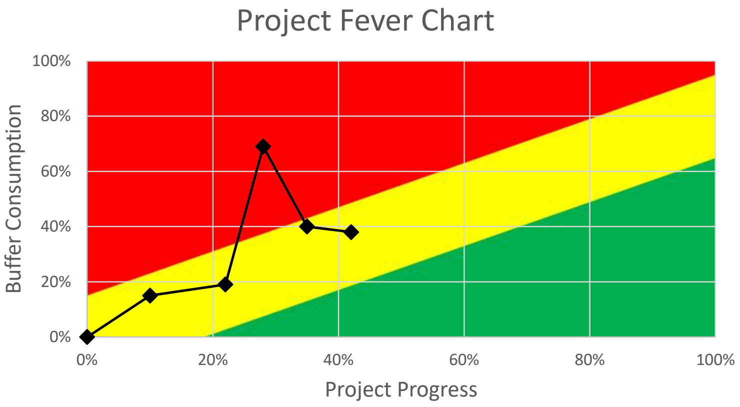


Figure X: Critical Chain Fever Chart

Figure X shows a project trend, with 5 progress data points. Report #3 was the concerning one (69% of the buffer used for 29% of project progress and well into the red zone). After positive intervention, buffer was recovered with the latest report saying 38% of the buffer has been used against 42% progress.

The fever chart colour coding is read as

- Yellow = expected progress zone.
- Red = high buffer use. Risk of being late - act to recover buffer.
- Green = low buffer use. Continue, or later in the project, plan to exploit early completion.

The slope of the yellow band is slightly different to the BBR=1 baseline in Figure X. This is because project completion dates are usually seen as a maximum completion date, rather than an average +/- x-days. The yellow zone has been adjusted to that if a project completes 'in the yellow', a small amount of buffer remains.

The key to using the fever chart is acting on the early warning that it can give when buffer is being consumed at an unsustainable rate.

The control limits implicit in the fever chart RYG zones are rarely based on historical statistical analysis, due to the low volume nature of project work[6] . Some CC users choose to set the boundaries between red yellow and green differently to shown here.

C: Management and Team Behaviours

As mentioned above, the claims of significant performance improvement from organisations who have adopted critical chain, do not come from using different schedule and report formats. They come from project team working practices that are supported by the formal techniques described above.

These behaviours can be seen as the '*principles of project workflow*'. They are not unique to critical chain, and many are well known to experienced project managers. What critical chain does is to explicitly acknowledge their importance and embed them into training, practices, and systems.

The principles of project workflow can be used without the formal critical chain techniques. They include:

- ***Focus-and-finish working***.
  Resources work one task at a time, finish it as quickly as they can, and pass it on before beginning the next task. So-called bad multitasking and task switching is discouraged and seen as wasteful. It is better for an under-utilised team member to support ongoing work or help prepare full-kits (see below) than it is for them to pull work forwards to keep busy.
- ***Full-kit***.
  Ensure tasks have all the prerequisites available before they are started. A managerial

---

[6] The fever chart is similar in principle to the control chart used in manufacturing. In volume manufacturing there is often good data on actual performance and the control limits are often set at =/- 3 standard deviations from the mean (target) value.

rule to not start a task that has incomplete full-kit. The idea is similar to look-ahead planning used by the LastPlanner System.

- ***Task estimates are not commitments***.
  The purpose of a CC task estimate is to help calculate a reliable project end date and associated buffers. Once the project is in execution task estimates are virtually irrelevant to the people carrying out the work. They are accountable for how they work rather than achieving an estimate. IE working in a focus-and-finish way, passing work forward as soon as it is completed, and not starting work unless it has a full-kit.
- ***Simple Task Progress Reporting***.
  When a project task is underway, the task manager frequently reports just two simple pieces of data (1) when they expect to be finished and (2) what help they need.
- ***High cadence progress reporting and team review***.
  Task managers report on progress at most weekly and often daily. Ideally task mangers input directly into the CC software and the project progress report is updated immediately. The frequent and short progress meetings focus on preparing for handover to the next resource and overcoming obstacles to workflow.
- ***Minimising WIP*** (work in process).
  This is related to focus-and-finish working. Project managers ensure that too much work is not underway at once. This applies at team, task, project and portfolio level.
- ***Clear task prioritisation***.
  Project resources have a single unambiguous view of task priority. This come into play when they finish a task and are ready to start another, and there is more than one task that could be started next. No more 'who shouts loudest'. CC software systems are designed to provide clear task prioritisation across the whole portfolio, based on the schedule and buffer status.
- ***Minimise undiscovered rework***
  Position QC activities to minimise the delay in identifying and rectifying rework[7].

Together these practices help embed and facilitate team wide collaboration and problem solving, supported by the metrics from critical chain. Without a collaborative project team the benefits from using critical chain will be hard to achieve.

On projects that involve significant procurement and contracting, Heptinstall and Bolton[8] suggest collaborative contracting methods are a prerequisite to using critical chain.

D: Continuous process improvement with critical chain

Causes of buffer penetration are captured on an ongoing basis during project execution, using a simple low-touch process.

From time-to-time project or portfolio managers review the causes, and prioritise some for further study/improvement.

---

[7] Whilst in a perfect world there would be no errors or rework, until then it is important to discover necessary rework as quickly as possible. See Cooper, KG (1998) Four Failures in Project Management, in The Project Management Institute Project Management Handbook, Pinto J (ed).

[8] Heptinstall ID & Bolton R. (2016) The Executive Guide to Breakthrough Project Management. Denehurst Publishing.

The process is similar to the Plan-Do-Study/Check-Act cycles popularised by Shewhart and Deming, and Scrum retrospectives. CC adds specific data that helps identify systemic obstacles to efficient workflow.

E: Critical Chain and Portfolios (Multi-project Critical Chain, MPCC)

If all projects in a portfolio are using critical chain, this allows the progress of all projects to be viewed in a single diagram. The Portfolio Fever Chart.

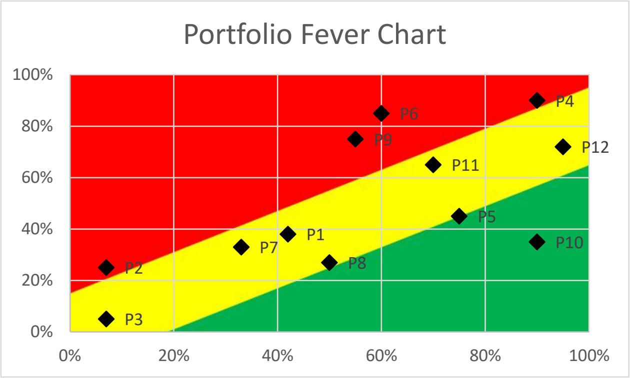


Figure X: MPCC Portfolio Fever Chart

This shows the most recent progress data point for each project. At a glance, portfolio managers can see which projects seem at risk and may require some support (P6 and P9 in this example). As well as which seem to be progressing as expected, and which might complete significantly early (P10 and P5).

MPCC applies the above workflow principle of “Minimising WIP (work in process)” to the overall portfolio, or the pipeline of projects.

When considering the portfolio, the flow of all projects through the organisation has a different constraint to that of an individual project (ie the critical chain). This is the part of the overall system with the lowest capacity to process project work. This might be a resource such as a key skill, an asset or physical space, or cash. Or it might be the attention span of senior management or subject matter experts which makes a specific project phase of projects most critical.

MPCC calls the identified constraint the ‘drum’ because it is this that sets the drumbeat for the whole project system. It dictates the maximum rate at which projects can be completed.

Projects ready to start are listed based on a single organisation-wide priority system. The next in line is held back until the latest moment that will allow it to arrive at the drum in time to be worked on. Or rather to arrive a little ahead of that time. The drum will have a small buffer of projects in front of it ready to be worked on. Not too many so that the system is overloaded with open projects, but enough to ensure the drum is never left with nothing to do.

This process is known as gating or staggering of projects[9].

Compared to other portfolio management practices, although there are fewer open projects when using MPCC, the project duration tends to be shorter and the throughput of completed projects tends to be higher.

Since the drum is the organisation's capacity constraint, almost all other project resources will not be used to their capacity. They have protective capacity which means if they have performance issues, they can catch up without reducing the overall project completion rate, this being constrained by the drum resource.

This protective capacity of non-drum resources is often a cause of management risk. It can be seen as 'waste' and trigger moves to reduce it and balance the capacities of all resources working across the portfolio. Whilst this feels instinctively 'good management' it can destabilise the overall system and result in a significant performance reduction. Managerial KPIs applied to individual functions, such as time allocated to projects, can be equally damaging to the high flow of project work if they reward busyness and penalise maintaining protective capacity.

Protective capacity is like the airbags in a car. Not using them doesn't mean it is a good idea to remove them.

Organisations that have transitioned to using MPCC from other portfolio management methods have generally found that before MPCC they had too many open projects. Step one of implementing MPCC is often simply to stop a significant portion of lower priority projects. Stopping 25-50% is quite common, though worrying to senior management. One extreme example was at the healthcare division of Japanese electronics manufacturer Omron. They stopped almost 90% of ongoing new product introduction projects – moving from almost 500 open projects to 50. IE putting 450 on hold. In the three months after doing this they had completed the same number of projects as in the whole of the previous year. And in the first two years of this new 'low-WIP' portfolio they more than doubled the rate of project completions and reduced the average duration by more than 50%. Mazda cars reported similar figures from their R&D department when it too introduced critical chain.

A common mantra for MPCC is "Stop starting and start finishing your projects".

Another common situation when organisations move to MPCC is they discover the actual portfolio constraint is not in the best strategic position. In the case of one aerospace manufacturer, it was found to be in a later stage quality control department. That was where project work was mostly delayed, as a result of several years of well-intentions across the board cuts in overhead budgets. This constraint was very easy to remove once identified, simply by adding resource and a few basic workflow improvements. The project drum then moved to a more logical strategic position (a piece of bespoke equipment that cost over £50M). An additional two staff costing less than £200,000 pa allowed this manufacturer to increase their revenue by over £100M pa.

---

[9] There are several ways to stagger/gate projects using MPCC. At the simple end of the continuum numerical limits are set for open projects ahead of the drum and a 'one-out, one-in' rule is used to start the next top priority project as soon as a project completes its drum work. At the other end, the drum is scheduled in detail and the upstream buffer of projects is dependent upon how much of the drum's time each project in-flight is expected to use.