Introduction to Critical Chain Project Management

by Georges Camy

When thinking about project management, surely the first thing that pops into your mind are Gantt and PERT charts. However, the real difficulty in project management is not creating fancy looking charts and graphs; it's knowing how to use them to keep control over the project, make the right decisions at the right time and most importantly: to save time and money.

When searching the Internet, I never found simple, easy-to-understand examples of critical chain project management. That's why I wrote this article. Please note that my intention is not to be academic. This is merely an attempt to provide the reader with a basic understanding of what critical chain project management is, and how to apply it.

About Critical Chain Project Management

Critical Chain Project Management (CCPM) is a technique for planning and managing projects that was developed by Eliyahu M. Goldratt, the author of the "Theory of Constraints". The Critical Chain method works well for projects in which tasks have volatile times. This is the case for most activities which rely on human skills. A typical example would be to write an article such as this one: depending on your inspiration and motivation it takes more or less time to complete the task. CCPM is not really suited for projects which require rigid scheduling and have very standardized tasks. An example of such a task would be to bake a pizza: no matter what, the pizza will need to remain exactly for 10 minutes in the oven.

The Barbecue

To illustrate how CCPM can be applied to a project, we will organise a garden party. Let's say a guy called Mark wants to organise a barbecue. But this year he would really like the barbecue to be a success, not a mess. Mark is lucky, the weather forecast has announced a splendid day and he managed to enrol some help: Julie and Steve were eager to offer their help in this daring endeavour. They decide to start the morning of the big day at 10:00.

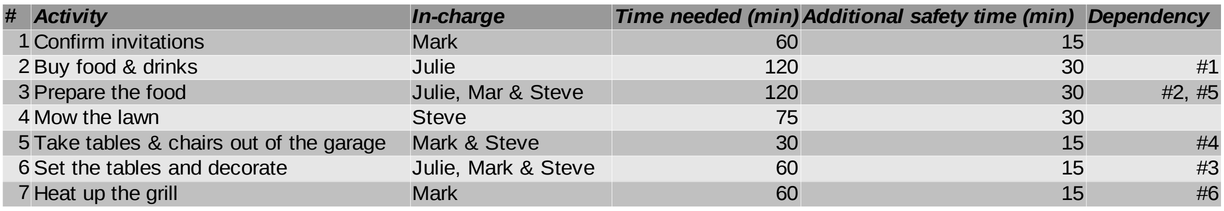
PART 1: PREPARATIONS

The Task list

Obviously the first step in any kind of project management is to think and plan ahead. So Mark finds a sheet of paper and writes down all the tasks that need to be done. He needs to buy food and drinks. Though for this he would need to know for sure how many people will come. So he should start by making some phone calls to those who have not responded to his invitation and are likely to come nonetheless. The food needs to be prepared as well. From last year he knows that the kitchen is too small to prepare all the food and that everything goes much faster if the food gets prepared directly on the garden tables. And these need to be taken out of the garage before, and maybe (probably) be cleaned as well. And then set and decorated. He also wants to have a barbecue that will be remembered, so decorations are a must. Looking out to the garden, it becomes clear that the lawn needs to be mown before anything can happen there. And let's not forget that it took over an hour last year to heat up that monster of a barbecue grill.

Definitely a lot of work lies ahead.

This is the plan in Mark's sheet of paper:



#	Activity	In-charge	Time needed (min)	Additional safety time (min)	Dependency
1	Confirm invitations	Mark	60	15	
2	Buy food & drinks	Julie	120	30	#1
3	Prepare the food	Julie, Mar & Steve	120	30	#2, #5
4	Mow the lawn	Steve	75	30	
5	Take tables & chairs out of the garage	Mark & Steve	30	15	#4
6	Set the tables and decorate	Julie, Mark & Steve	60	15	#3
7	Heat up the grill	Mark	60	15	#6

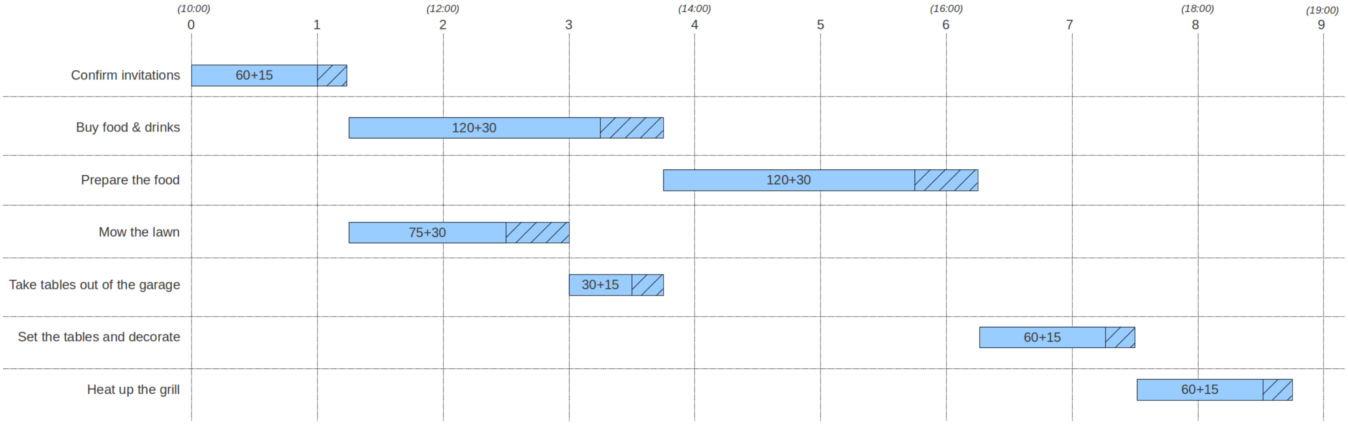
Mark is quite happy, everything seems well organised and under control. He found out that the preparations for the barbecue consist in 7 activities, which he divided among his 2 friends and himself.

Some activities depend on the successful accomplishment of previous activities: Mark noted this down in the column "Dependency". For example the food can only be prepared after it has been bought. Dependencies can also come from resource shortage: since everyone is helping to set and decorate the tables, Mark will only be able to heat up the grill after the tables are set. Indeed, Mark wants to supervise these two important tasks.

Mark also made an estimation of the duration of each task and added an additional safety time to it ("buffer").

The Gantt Chart

From the above table it is quite difficult to see whether or not everything can be done in time. Mark remembers having seen this nice-looking Gantt charts at work and after a quick lookup from Wikipedia, he draws the following chart:



Voilà, all tasks nicely displayed! Mark took care to add a project timeline in hours (starting at 0) on top of the actual timeline. Inside each task he wrote the duration of the task in minutes, plus the additional safety time (hatched). From this chart it is easy to see that if the preparations start at 10:00, everything will be ready by 18:45. So, having the guests come at 18:00 when the grill is nearly heated up seems like a good plan.

Unfortunately, on this Gantt chart it is difficult to see which tasks are dependent on which tasks.

PERT & AON

PERT stands for Program (or Project) Evaluation and Review Technique. PERT diagrams are more suited to display the dependencies that were not so clear in the example above. (To be more exact, the diagram we are going to use is actually called a Project network, or Activity on node (AON) diagram, but there seems to be a lot of confusion about these terms.)

To create the diagram, each activity gets drawn in a little box:

Early Start	Duration	Early Finish
	Task Name	
Late Start	Slack	Late Finish

The box contains six numbers:
- The earliest possible start date
- The expected duration of the task
- The earliest possible finish date
- The latest possible start date
- The latest possible finish date
- The slack or float

In our case, all numbers and dates are expressed in minutes, with the dates starting at t=0 (t being the project start or simply 10:00 on Saturday).

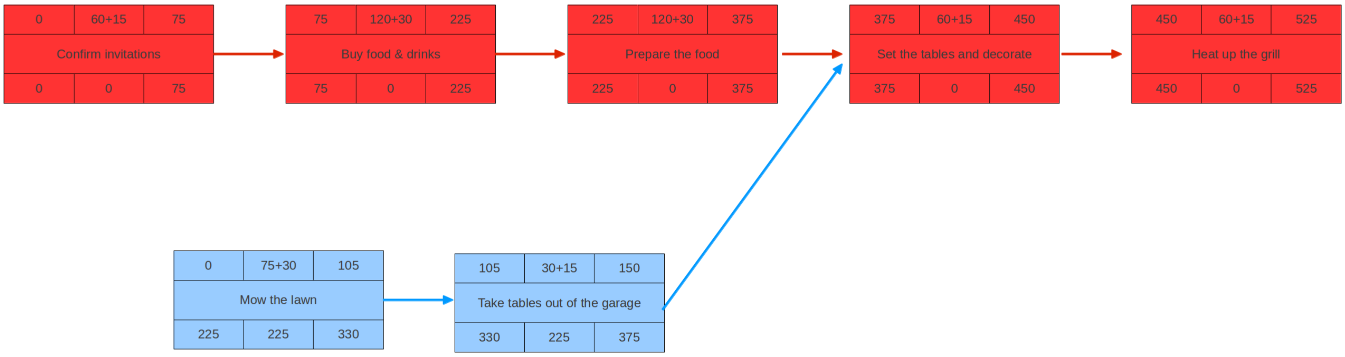
How to fill in the numbers?

To establish the Early Start and Early Finish dates we first go through the diagram from the first task to the last task: i.e., the first task “Confirm invitations” takes 75 minutes to complete, therefore the next task can only start at t+75, and so on. For the “Set the tables” task there are two preceding tasks. “Prepare the food” completes earliest at t+375 and “Take tables out of the garage” at t+150. Hence “Set the tables” can start earliest at t+375.

To establish the latest possible finish and start dates, we repeat the same procedure, this time only working backwards: the last task “Heat up the grill” terminates earliest at t+525 and thus needs to start at t+450 at the latest. The preceding task therefore completes latest at t+450 and so on.

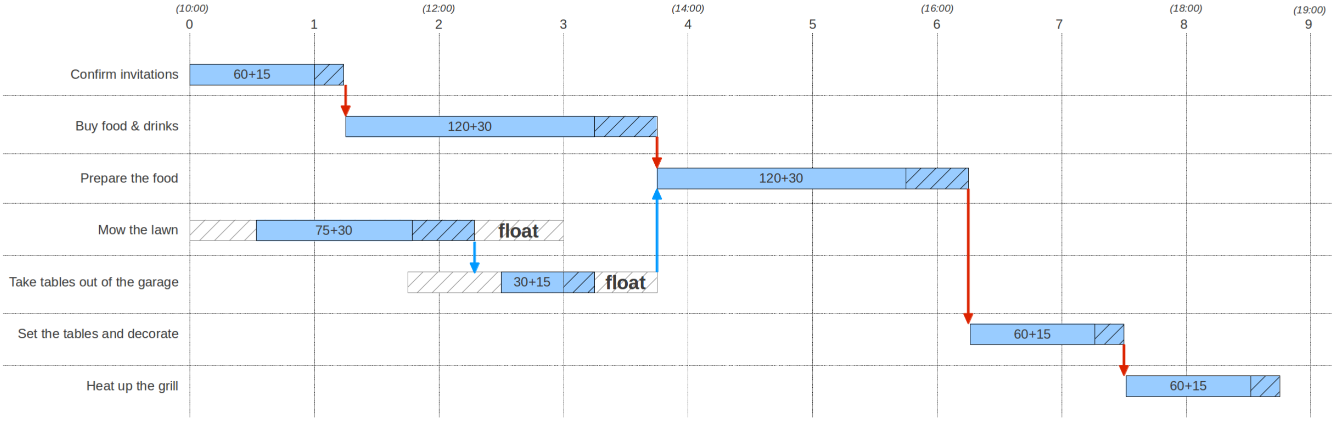
The slack (or float) is the time difference between the Early Finish and the Late Finish (or, alternatively, between the Early Start and the Late Start). It is basically the additional time that the task can take without impacting on the timeline of the other tasks. For this reason the slack is often called "float" to symbolise that the task has space to float in the timeline.

Now this all sounds rather complicated, so let's look at the diagram that Mark came up with. We can easily distinguish the dependencies and sequence of the different activities. We can also see that there is a chain of tasks where the slack is 0. This is called the critical path (shown in red). Any task in this chain that will take more time than planned will delay the whole project.



Since the tasks on the critical path have such an impact, it is important to keep a close eye on them.
Gantt and AON are quite complimentary. Whereas AON charts contain more information and clearly show the dependencies, Gantt charts are easier to read.

Luckily enough, the complexity of the Garden Party project is low, so it is quite possible to clearly show the dependencies and critical path also in a Gantt chart. This is the final diagram that Mark ends up with:



Mark is really happy. The above chart shows everything: the timeline, the critical path, the security buffers. Nothing can go wrong now, can it?

PART 2: EXECUTION

10:00
Mark picks up his phone and starts calling his friends. A few chats here and there, but in the end he finished the task at shortly after 11:00. A lot of people will come. This is going to be one hell of a party.

11:05
Julie shows up, she brought her friend Anna to help out. They are quite happy to hear that so many people will come. They then head off for the supermarket.

11:30
Steve hasn't shown up yet. He was supposed to be there at 11:15! Mark calls him on his mobile phone. Steve just had his breakfast and is not worried: after all they have time until 13:45 to set up everything? There is massive amount of time still ahead. Unfortunately, since Steve has the lawnmower (he borrowed it last week and today seemed the perfect day for him to bring it back), Mark cannot start yet.

11:50
Steve shows up. Mark is a bit upset but things get moving at least.

12:10
Julie calls to ask some details about what they shall buy. They are doing great, things are moving super fast and the girls decide to take a break and go to the clothes store.

13:05
Steve finishes with the mowing. Ok, there is 5 minutes delay, but everything is still on schedule.

13:06
The key to the garage is nowhere to be found.

13: 16
Still no key. Steve tries to pick the garage lock.

13:20
Julie calls and is amused by the guys being so chaotic.

13:21
The key was in the other jacket.

13:22
Wave of despair flowing over Mark as he realises that the tables are hidden behind a pile of other stuff that has accumulated since last summer.

14:15
The girls come back from shopping. They had a great time and have a bunch of new outfits. Half of the tables are out, but are not yet mounted and cleaned. The girls are a bit annoyed as they gave the boys additional time (after all they should have been back latest at 13:45) and the guys are not ready yet!

14:25
The girls start to get impatient and suggest preparing the food in the kitchen. The boys tell them that they will have finished in a few minutes and that they shall wait.

14: 30
The girls start preparing the food in the kitchen

14:35
The tables are set and cleaned. Looking at the messy situation in the kitchen, Mark guesses that there is no way the food preparation can now be moved to the tables without upsetting the girls.

15:00
Food preparation goes much slower than anticipated. The kitchen is just too small. Only one person can reasonably work. Julie and Anna take turns, the guys have a beer in the garden after telling the girls that they told them so.

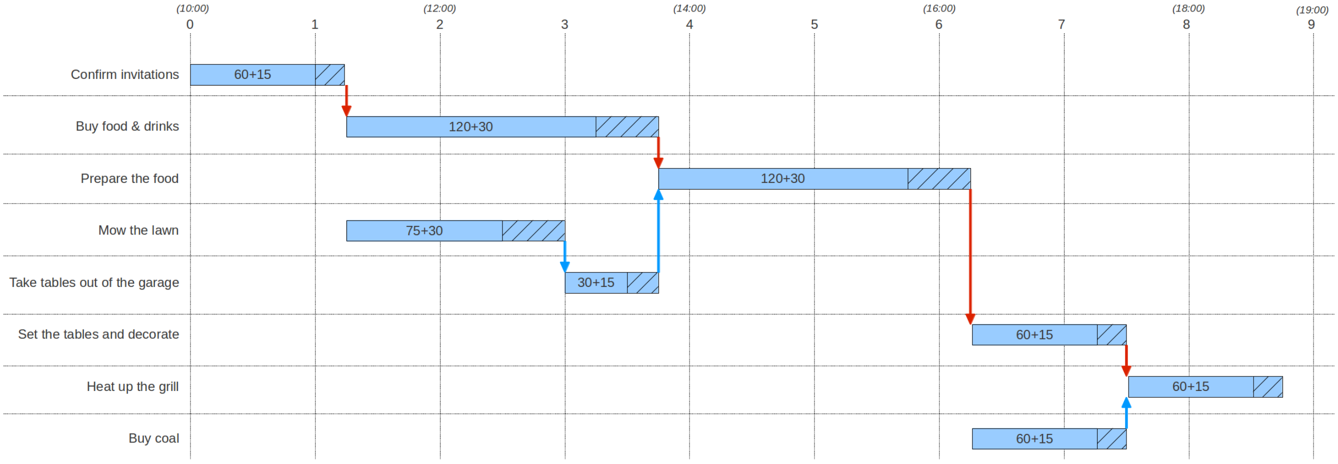
15:30
The girls get upset that they have all the work and the guys are doing nothing.

15:31
Steve points out that Mark does not have any coal for the barbecue.

15:32
Mark panics. That task has not been identified beforehand. Is the project at risk? Can they still finish in time?

15:33
A quick look on the project diagram shows that there is no need to worry. Buying coal only impacts the last task, “Heat up the grill”. They can still go buy the coal, while the rest finishes up setting the table.

Mark quickly updates the diagram with the “Buy coal” task:



15:34
Mark and Steve head off to buy coal.

16:00
The food preparation takes much too long. Julie starts crying. Anna tries to calm her down.

16:15
Anna starts transferring food preparation to the tables, so that she and Julie can work at the same time.

16:45
The guys are back, there was a huge line at the check out. Anna finished moving everything out.

17:00
Food preparation is now getting along fast.

17:35
Mark decides that even though not everything is absolutely ready, they have enough food. The tables must be set up. There are only 25 minutes left to set up the tables. This is not enough time to make the fantastic decorations that Mark wanted. Thankfully enough they have Anna to help them to make the best out of the situation.

17:45
The first guests arrive.

17:46
Delegating the decoration part to Julie and Anna, Mark decides to heat up the grill.

17:50
More guest arrive. Steve and Anna take care of them, while Julie tries to finish setting up the tables.

18:05
Julie is upset that everyone is having fun while she is working and stops decorating the tables. Anyway, the party has officially begun 5 minutes before.

18:06
Peter decides that he needs to show Mark how to heat up a grill.

18:07
Tom and George think the same.

18:08
There is a huge flame on the grill after additional spirit has been poured over the coals

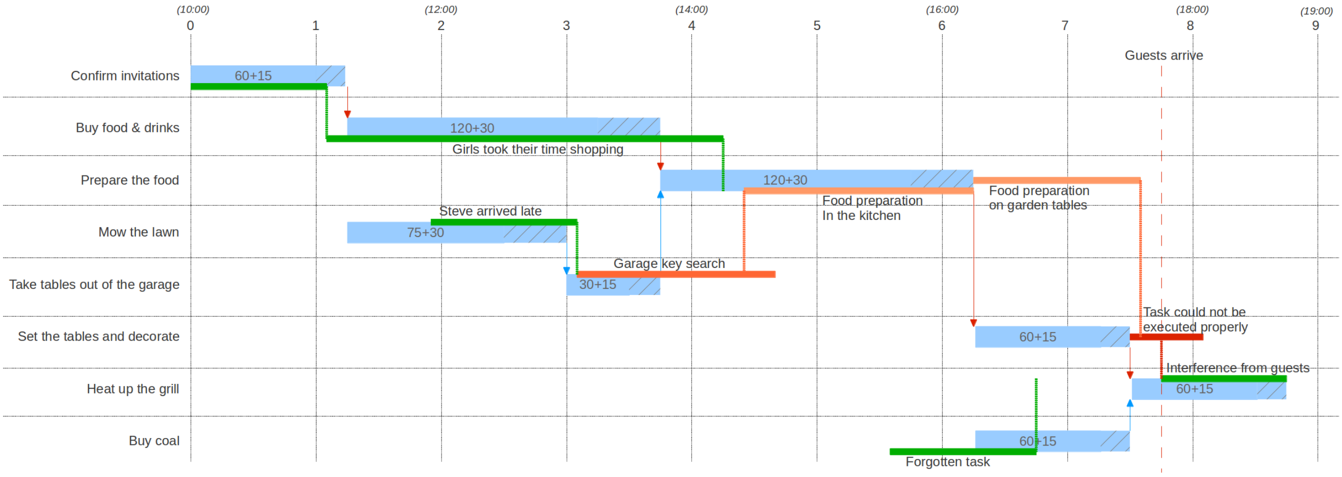
18:09
The fire in the grill is out.

18:38
The grill is finally on, the barbecue can start.

Everything went well in the end, but what a stress!

PART 3: AFTERMATH

In the end the party was a success, however there was much stress preparing it, and the decoration part could not be realised. Let us draw a chart of what happened and compare it to the original plan:



The tasks that are marked in green were still accomplished in time. The tasks marked in orange were realised but took longer than expected. The task in red could not be accomplished.

Let us look at the individual tasks:

#	*Activity*	*In-charge (planned)*	*In-charge (actual)*	*Comment*
1	Confirm invitations	Mark	Mark	This task went as planned.
2	Buy food & drinks	Julie	Julie & Anna	The girls took longer than expected, but still finished before the boys.
3	Prepare the food	Julie, Mark & Steve	Julie & Anna	Because the tables weren't set up, the girls decided to start preparing the food in the kitchen. This reduced the production capability dramatically. In the end Anna transferred the food preparation over to the garden tables, but this transfer required additional time.
4	Mow the lawn	Steve	Steve	Steve started late, but finished the job in time.
5	Take tables & chairs out of the garage	Mark & Steve	Mark & Steve	The garage key could not be found. Due to the unexpected cluttering of the garage, this task took much longer than expected. Thus the food preparation was delayed.
6	Set the tables and decorate	Julie, Mark & Steve	Julie (& Anna)	There was no time left for this task. All 4 of them started the task, but Marc soon had to take care of starting the grill and Steve and Anna had to take care of the guests.
7	Heat up the grill	Mark		There was interference from the guests. This could have been expected, and might well be the reason why lighting up the grill always takes an hour. But this is part of the fun, and the task was done in time.
8	Buy coal	Steve	Steve & Mark	This was task was unexpected, but could be dealt with in time.

PART 4: THE HEART OF THE PROBLEM

The real issue was that despite all the planning that Mark did beforehand, he didn't use his toolset to control and direct the project. There are two major forces that have to be taken into account when doing a project:

1. Parkinson's Law

"Work expands so as to fill the time available for its completion."[1]

This sounds quite humorous, but nonetheless is an observable certainty. There are different phenomenon that contribute to Parkinson's Law:

- **Bad multitasking**: Julie and Anna had plans to buy clothes and when they saw that they were quick with the food shopping decided to do the clothes shopping as well. In the end they finished both tasks (food shopping and clothes shopping) but exceeded the time that was allocated for food shopping.
  Bad multitasking often happens when a person has several projects. It would of course be fastest to focus on one project, finish it, and then move to the next. Unfortunately in reality there are often idle times in a project. It then seems appropriate to do some task from another project during that idle time. However, it often happens that this other task takes longer than the idle time allows for, thus delaying the first project. Also, shifting one's attention from one project to another requires a certain adaptation time and reduces the working speed. As a summary, doing several projects in parallel, i.e. multitasking, requires a non-negligible mastery in resource planning.

- **Student's Syndrome**: The theorem says that students wait until the last possible moment to start learning for their exams; that is when the task has become urgent. This might be due to several reasons: either students want more time because they want to do a deeper analysis, or they procrastinate or simply get interrupted by more important tasks. Steve did not start with the lawn mowing until the task became critical. Alas, we humans only do things when they are urgent or provide us with pleasure.

An immediate result of Parkinson's Law is that the safety buffers that Mark added to his project will be consumed, no matter what.

The only way to get around Parkinson's law is to make tasks important and urgent.

2. Murphy's Law

"Anything that can go wrong will go wrong."
"...and the likelihood of something going wrong is proportional to the urgency and importance of the task. "

Let us see how Murphy stroke in the garden party project:

- The garage key was lost. This was unexpected and caused additional work.
- The garage was cluttered. The "Take tables out of the garage" ended up being on the critical path as a consequence.
- The contingency plan to prepare the food in the kitchen ended up costing much more time than expected.
- There was a forgotten task: "Buy coal". Fortunately enough this did not impact the project timeline, but removed resources (Mark & Steve) from the project.
- Guests interfered with the lighting of the grill. (Although this can be considered part of the fun of a barbecue party)

The only way to fight Murphy's law is to either allow for safety margins, or incorporate safety precautions into the project. Typical safety precautions would be:

- Simulations and dry-runs.
  Example: the friends could have met up the day before and discussed each step of the project.

- Make a thorough assessment of the tasks beforehand.
  Example: the garage situation could have been checked beforehand.

- Empower and delegate responsibilities. When a task is delegated to a person, it becomes more important to this person, also because her honour and credibility is at stake.
  Example: the table decoration part could have been made solely Julie's responsibility from the beginning. But that also would have meant that she would not have been available for the food preparation.

- External fool proofing. Discussing the project with an external expert may yield additional insights and show faults in the

[1] Cyril Northcote Parkinson

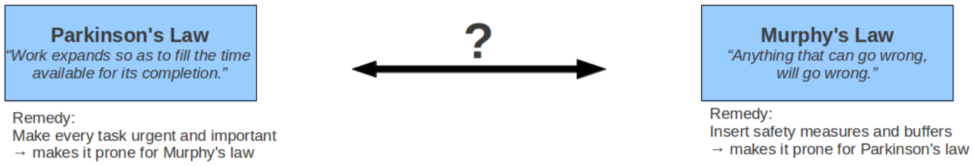
project.
Example: Mark could have called a friend who is very successful at barbecue parties and discuss the project with him.

- Insert controlling tasks or make tasks redundant. You can have someone going over the work after it is finished. You can also have a second team to jump in if the first one fails.
  Example: Mark could have asked Peter to help with the mowing and have him bring his own lawnmower.

In the end, all measures to wield off Murphy's law have the same drawback: they require additional time and/or resources.

Between Scylla and Charybdis

Therefore when doing project management, you are basically confined between the two extremes:



Parkinson's Law
"Work expands so as to fill the time available for its completion."

Remedy:
Make every task urgent and important
→ makes it prone for Murphy's law

?

Murphy's Law
"Anything that can go wrong, will go wrong."

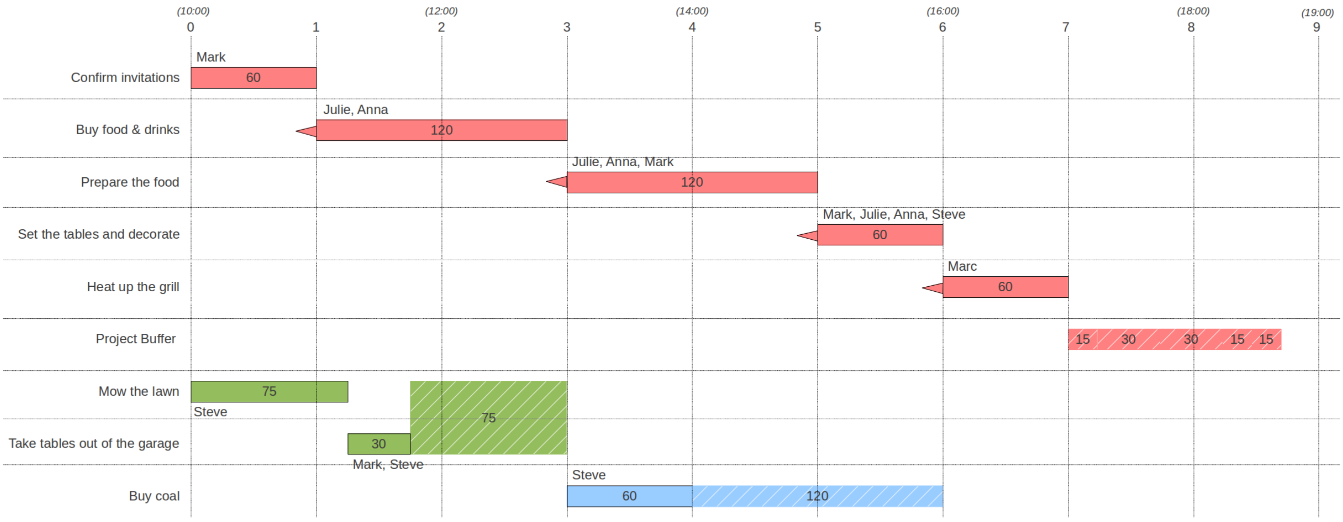
Remedy:
Insert safety measures and buffers
→ makes it prone for Parkinson's law

And this is the dilemma: the more safety time you allocate to a task to counteract Murphy's law, the more prone you are to fall into Parkinson's clutches. The less time you allocate to a task, the more urgent and important it gets, but at the same time the more the whole project schedule is endangered should something go wrong (and something will go wrong, according to Murphy's law).

PART 4: CRITICAL CHAIN PROJECT MANAGEMENT

CCPM tries to avoid this dilemma by:
- allocating the minimum amount of time to a task, so that the task stays urgent and the resource will be fully dedicated on the task until it is done
- keeping a critical chain critical by making sure that the resource will be available at the end of the previous task(s)
- using buffers to counter variances in time costs

It is vital to understand that Critical Chain Project Management is not a graph, but a method. Let us go back to the barbecue party to illustrate this. Using CCPM, we would come up with the following diagram:



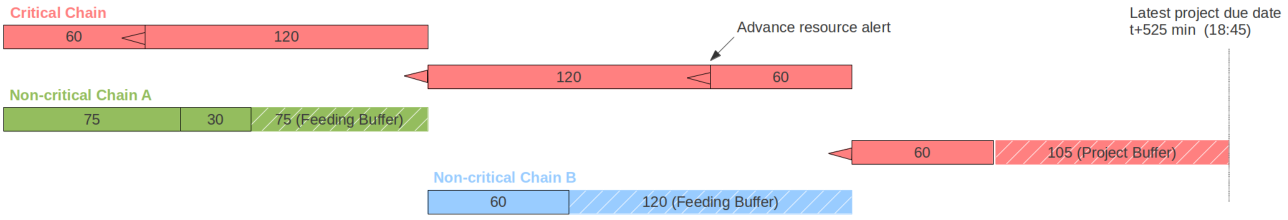
What happened:

- Like before we wrote down all the tasks and ordered them according to the dependencies

- Then we balanced the resources.
  Obviously Steve cannot buy coal and mow the lawn at the same time. Mark is on the front line when it comes to “Set the tables and decorate”, but also is the least important resource assigned to “Prepare the food”. It is crucial that the load for the resources is well balanced across the project.

- Reduce the time
  In Critical Chain Project Management you don't add a safety margin to an individual task; instead you usually take off a certain percentage from the estimated time. (The estimated time usually can be expressed as “there is a 90% chance that the task will take this amount of time to complete”). In most cases you go for a likelihood of 50% that you will complete the task in this time. The sliced off time (the difference between the 90% and 50% likelihood) gets added to the end of the project as project buffer. Indeed, some tasks may finish earlier, while others will finish later, evening each other out. For simplicity's sake, we will use Mark's buffers.
  The idea behind the accumulation of the buffers is that you need to protect the project and not the individual tasks. This project buffer is your safety.

To summarise, we will end up with the following visualisation:



In our project we have 1 critical chain, and 2 non-critical chains. The non-critical chains have their own buffer, the so called “feeding buffer”.

To the critical chain we add something that we call “Advance resource alert”. This is to ensure that the critical chain will stay critical. As soon as the preceding task nears completion we warn the resource assigned to the next task, so that they can start preparing themselves. This is extremely important because each task may not start according to the initial schedule if any of the tasks in the chain is completed later... or earlier! In our example this alert is not necessary as the tasks on the critical chain get executed by the same set of people. We can safely assume that they will be ready to start the next task as soon as they finish the current one!

However, Mark could have invited Julie and Anna for breakfast for example while he made his confirmation calls. That way, should he have finished earlier, he could have briefed Julie and Anna about the amount of food to buy and send them out right away.

How do we manage and control the project?

Critical Chain Project Management is all about managing buffers.

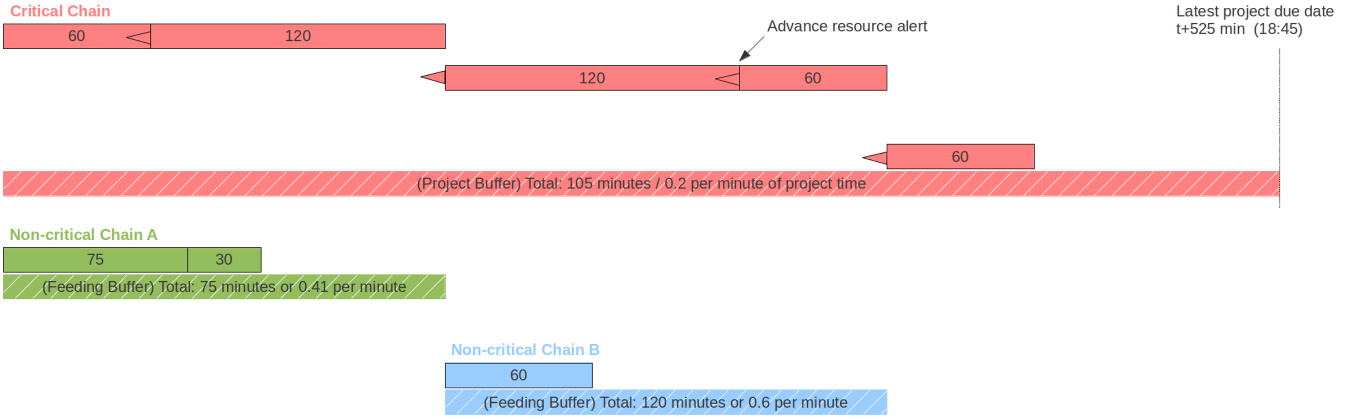
As tasks are completed they use or add to the project buffer. So really all we have to do is to watch how fast the buffer changes. If the buffer grows, tasks are finishing faster than anticipated and there is no need to worry. If the buffer shrinks, tasks are taking up more time. Depending on the rate at which this happens we will have to intervene.

Just keeping an eye on the current buffer status might be all you need to do in order to keep control over the project. But if you want to go all mathematical, an easy way to watch the buffer usage is to simply spread the project chain buffer evenly over the whole project timeline *(i.e. from 10:00 to 18:45).*

In our example we have 105 minutes of buffer time and a total project time of 525 minutes. This means we have 105/525=0.2 buffer time minutes per project minute. At 12:00 we should have used up a maximum of 22% of the buffer (=24 minutes of buffer time.) Given that there was only one task to finish before noon, this would mean that Mark could have been 1 hour and 24 minutes on the phone without impacting the project timeline. (His own estimation had only included a safety of 15 minutes. Our calculation method of just spreading out the buffer equally over all tasks is assigning to all tasks the same complexity and safety margin ratio.)

However, since Mark finished his task at 11:00, using up exactly 60 minutes, as was his estimation, we would have added 0 to the buffer. The buffer therefore wouldn't have changed.

We can visualise the calculation method like this:


Critical Chain
60
120
Advance resource alert
Latest project due date
t+525 min (18:45)
120
60
60
(Project Buffer) Total: 105 minutes / 0.2 per minute of project time
Non-critical Chain A
75
30
(Feeding Buffer) Total: 75 minutes or 0.41 per minute
Non-critical Chain B
60
(Feeding Buffer) Total: 120 minutes or 0.6 per minute

To summarise: if the project buffer decreases at a steady predicted rate, all is fine. If it decreases slower (or increases), even better so: tasks are finishing faster than scheduled.

However if the buffer starts to decrease faster than expected we need to keep a close look at the current task and act if necessary. At this point the proper course of action is to ask how much time is needed to complete the task and to analyse if there is something that can be changed in this task or left aside in order complete the task faster. If all this doesn't help then it might be time to either delay the overall project due date or consider contingency plans that may change the project outcome (like cutting out later tasks, e.g. “Decorate the tables”). You can of course also gamble and use up the entire project buffer on this task, hoping that the remaining tasks will be executed flawlessly and not eat into the buffer.

The non-critical tasks are treated in a similar fashion to the critical chain. The only difference is that it is unnecessary to go through the hassle of an advance resource alert system: when a non-critical chain used up its entire buffer, it simply starts eating into the project buffer. At this point of course - or better: shortly before this happens - it becomes necessary to act.

Why should you consider using Critical Chain Project Management?

I am sure that it should be possible to organise a barbecue party without having to resort to project management techniques. However, other projects of yours might be slightly more complex. In this case CCPM may prove itself a valuable technique for making your project a success. And – most importantly – diminish time requirements, and reduce overall stress by increasing your control and reactivity.

REFERENCES

There are many articles on the Internet and project management repositories such as The Project Management Hut (www.pmhut.com). However, this article was based mainly on the following references. I encourage the reader to have a look at them as well.

Francis S. “Frank” Patrick, Getting Out From Between Parkinson's Rock and Murphy's Hard Place
Focused Performance
www.focusedperformance.com

Savya Rafai, How Vulnerable are we to Uncertainties Faced in a Project?
Planstrat
www.planstrat.com

Wikipedia: Critical Chain Project Management (CCPM)
http://en.wikipedia.org/wiki/Critical_chain_project_management

Wikipedia: Critical Path Method
http://en.wikipedia.org/wiki/Critical_path_method

Wikipedia: PERT
http://en.wikipedia.org/wiki/PERT

Wikipedia: Activity on Node (AON)
http://en.wikipedia.org/wiki/Precedence_Diagram_Method

Wikipedia: Eliyahu M. Goldratt
http://en.wikipedia.org/wiki/Eliyahu_M._Goldratt