Let’s TOC

THEORY OF CONSTRAINTS

Himanshu Pancholi
Back to the Basics

Before rushing into the detail explanation on Theory of Constraints, we need to understand “What is a Constraint?” Similarly before learning about Constraint, we should understand about System. Going a step further, “What are Goals of a System?” In short, we are about to start from the basics.

System, in simple words, is the collection of interdependent tasks. System can be thought of as a chain, and the tasks as the links of a chain. Now think about this: How strong can a chain be?

Let’s say all the links in a chain can hold up to 50 pounds of weight, but only one single weak link can hold 10 pounds; then how much weight can a chain as a whole hold? - 10 pounds, I guess.

This proves that the entire capacity of a chain is limited to the capacity of its weakest link. Applying this example to the System terminology, we can safely conclude that: the performance capacity of a System is limited to the performance output of its weakest element. This element or a task is nothing but the Constraint.

Thus, a constraint is defined as a limitation of a system which restrains it from achieving the desired level of performance. A constraint determines the maximum capacity of a System.

Now let’s go back to the Chain philosophy. What is the main objective or goal of the Chain? - To lift the load; again a guess. Now ask this to yourself- As the owner of a Chain what concerns you the most: How much load a Chain can hold? Or how much load individual links of Chain can hold? If your answer is the latter one, then you really need to put a lot of thoughts in your chain.

The Goal of a System as a whole is far more important than the Goals of its individual elements. This holds true because: When it comes to delivering a product, the customer doesn’t care about the time taken by individual tasks, what he cares about is the final Delivery Date. And what hampers this final delivery date or the true Goal of the System - is nothing but one of its elements, one of its tasks, one of its weak links.

Being part of the System, our task is to exploit this weak link in the System and strengthen it. As we all know, our System is not getting any simpler each day. Thus to deal with this complex set of entities, we have a well-defined set of steps or a procedure.

In 1984, an Israeli physicist stirred the business world when he introduced a brand new concept, a System-oriented process improvement methodology; which says that System has a single Goal, and that systems are composed of multiple linked activities, one of which acts as a constraint on the whole system. In his fiction novel, The Goal, he explains the story of a character named Alex Rogo, who manages a metalworking plant where everything is always behind schedule. His distant acquaintance, Jonah, who represents this Israeli physicist himself, helps him solve his company’s problems.
It’s not the story of Alex Rogo that provoked the business world, but it was the concept introduced by the physicist that brought about a subtle revolution in the way systems are managed. This Israeli physicist was Dr. Eliyahu Goldratt and the concept introduced by him was Theory of Constraints.

**What is Theory of Constraints?**

There are tons of good definitions such as: “Theory of Constraints is an overall management philosophy that aims to continually achieve more of the goal of a system.” Or this one: “It’s a business philosophy which seeks to strive towards the global objective, or goal, of a system through an understanding of the underlying cause and effect dependency and variation of the system in question.”

But for the time being, I put my money on my definition: TOC is a management philosophy which, through the process of logic reasoning, helps the management to identify the constraint in the system and change the system to eliminate this constraint.

In previous section, we laid some groundwork by learning few basic concepts. Now let’s get on with the show. What’s all the hype about TOC? How can I use it in managing my projects? What is TOC?

So far we know that there is a weakest link in the System, which is causing the System to under-perform. Our job is to exploit this weak link and take necessary actions, such that System will generate maximum expected output.

Here is an attempt to visualize my explanation. Below is a typical scenario of a Software Project Lifecycle. The sequences of events are:

- **High Level Design** (10 Days)
- **Detail Design** (15 Days)
- **Coding** (20 Days)
- **Unit Testing** (15 Days)
- **Subsystem Testing** (20 Days)
- **System Testing** (22 Days)

Please note that the above is very hypothetic scenario, just to simplify the explanation. In reality, the tasks are split to be divided among resources.

Below each phase is mentioned the **Planned Duration** for that phase. It is clear that one task can not be started until its previous one if finished. Another thing evident is that the **Total Planned Duration** for the entire project is **102 Days**.

Once the project completes, we calculated the Planned and Actual duration taken by each phase and the result is as shown below.
It is clear that in reality the Project took **115 Days**, which is longer than planned. Now our task is to identify what went wrong. Why was the project delayed?

Evident from the below graph, there are 3 phases delayed. But the phase that is delayed the most is Unit Testing. Without any doubt, we conclude that our weakest link is Unit Testing, which acted as a constraint and jeopardized the entire project.

The small explanation below is general to all areas, not just for Project Management. Theory of Constraints is a tool that guides the user to find answers to the few basic questions relating to the change. These questions are:

- What to change?
- What to change to?
- How to cause the change?

TOC views an organization as a chain consisting of many links. The chain, however, is only as strong as its weakest link. TOC states that the overall performance of an organization is limited by its weakest link. Hence if an organization wants to improve its performance, the first step must be to identify the system’s weakest link, or constraint.

The following are the five basic steps:

1. **Identify the constraint**: Identify the operation that is limiting the productivity of the system. There may be more than one constraint, but we need to focus on the constraint impacting the most.

2. **Exploit the constraint**: Achieve the best possible output from the constraint. Plan necessary actions so that the constraint can be used in the most effective way possible.
3. **Adjust other activities as per the constraint**: Link the output of other operations to suit the constraint. Smooth work flow and avoid build up of work-in-process inventory. Avoid making the constraint wait for work.

4. **Elevate the constraint**: In situations where the system constraint still does not have sufficient output invest in new equipment or increase staff numbers to increase output.

5. **If anything has changed, go back to step one**: Assess to see if another operation or policy has become the system constraint.

If I explain further about TOC, in little more detail, it will stop making sense to Project Managers. Since I have written this paper to explain the use of TOC in Project Management, I will now move to the more relevant part.

---

**Project Management – The TOC Way**

If you have already fallen asleep, I don’t blame you. But now it’s time to wake up. Help yourself with a coffee and buckle up your shoes, for now we begin the most interesting topic about TOC.

I would recommend you to not just read; but learn, grasp and digest the below two laws. From hence forth, I will refer to them as Parkinson’ Law and Murphy’s Law respectively.

- “Work expands so as to fill the time available for its completion” - Parkinson's Law
- “Whatever can go wrong, will go wrong” - Murphy's Law

These 2 Laws have a biggest impact on project management and thus on implication of the Theory of Constraints.

The method that we commonly follow to manage projects is called as **Critical Chain**. This method helps manager to have the whole system view of the project. Critical Chain also involves a method called Buffer Management, to manage the buffer associated with each phase.

**Problem with Critical Chain:**

The question is: If the Critical Chain method is being used happily by managers and team members, then why do we need Theory of Constraints to rule over this method?

This is where Parkinson’s and Murphy’s laws pitch in. Conventional method of Critical Chain ensures that each task of a project is delivered on planned date and with desired level of reliability. But in reality; the only date which matters is the final project date, and the only reliability that matters is of the final developed project. And as a matter of fact, the happiness that matters the most is of the customer.
Let me explain you in short what Critical Chain method is all about. As it is important to understand Critical Chain and Buffer management, to get a grip about TOC; I will explain it in points.

- In Critical Chain, there are 2 set of tasks: **Critical** and **Non-critical**. The delay in critical tasks results into delay in project.
- Since project cannot be delayed, critical tasks are required to finish on time. Here, Murphy's Law takes action, jeopardizing the entire project.
- To protect against Murphy’s Law, we add some buffer time to each Critical task. Thus we now have 3 time estimates: Minimum Time, Average Time and Time with Buffer included.

![Diagram showing Critical Chain]

- Our Critical Chain along with some buffer looks like this:

  - Design: 10 Days
  - Buffer: 2 Days
  - Coding: 15 Days
  - Buffer: 5 Days
  - Testing: 14 Days
  - Buffer: 5 Days

- We allow each Critical Task to consume some buffer, only if required. This gives us some protection against Murphy’s Law.
- But sooner than we can say “Time”, Parkinson’s Law comes to life and now each task starts taking the **Time with Buffer**. This concludes that most of the time, we waste the buffer that is allotted to each critical task.
- Here we ask: Why does it happen? Why do we run into the spare buffer always? Why can’t we finish the task on average time? The answer to all these questions is the most wrongly used term in project: **Multitasking**.
- When the project resource is multi-tasking, it often happens that the resource is busy with other tasks when it is time to start the critical task.
- Effectively, we start late and we are about to run into the buffer even before Murphy’s Law is materialized, even before anything wrong has happened. And if anything wrong happens (as per Murphy’s Law), then the task duration are further shifted and soon the gap between planned and actual time starts widening.
• We not only waste the critical tasks buffer, but we overtake the allowed margin. This way the Project gets delayed, one day at a time...

**TOC to the Rescue:**

We saw the problems associated with the conventional Critical Chain and Buffer Management method. To help against the Murphy’s Law we add some buffer to the tasks. But soon this buffer materializes the Parkinson’s Law. How can we achieve some kind of balance that will neutralize both these laws? Why can’t we have a negative buffer for tasks that finish earlier than expected, and start the adjacent task before its planned start time?

Parkinson’s Law can be handled in the following ways:

- **Remove the buffer that we have allotted for each task.** The shrink in the time will bring some discipline and respect for planned start dates. Since we start on time, and there is no buffer to expand the work, the effect of the Parkinson's Law is likely to disappear.

- **Remove the task due dates.** This means no task has planned finished date. Each task will be planned to complete in the earliest time possible. This sounds gross, only if you didn’t understood it completely.

  By removing the due date I meant remove the minimum, average and buffer time for each task. Each task will be planned to complete until the start date of following task. This will help managers to avoid Parkinson’s Law, and this will help team members to freely express their views of task duration, based upon their personal attributes. This means, if task is supposed to finish in 10 days, but employee thinks he will require 12 days; he can simply state this to the manager. This will not only contribute to team member’s involvement in planning but will bring certain level of work satisfaction among the team.

Murphy’s Law can be dealt in following ways:

- **Say No to Multi-tasking.** Avoid unnecessary distractions caused by multitasking or working on multiple projects simultaneously. Since each team member will be now working on single task at one particular time, there are less chances that he will have something at his desk, when it’s time to start the critical task. This will certainly remove the transition time to learn, unlearn and relearn the activities involved in varied tasks.

- **Introduce something called as Resource Alerts.** These are inserted into project plan by the manager. Resource Alerts are used to inform people that they are about to go onto critical chain. During this time they should clear their desk of other tasks and should not be asked by others to do tasks.

- **Have some contingency buffer, but not at the task level.** This buffer should be hidden from task level to avoid any kind of ill-effect caused by Parkinson’s Law. This contingency buffer is required because although we start on time and we work only on single task at a particular time, there are things that can go wrong as per Murphy’s Law.
and if don’t have any backup or contingency then we are in a pickle. The only question that remains here is **where can we place this buffer if it is hidden from the task level?**

Here’s the answer. The difference between average and time-with-buffer estimates is the buffer at the end of each task. Instead of spreading it around among the tasks, where it usually gets wasted, we can take a "whole system" view and concentrate it where it will help us the most. The buffer associated with the critical tasks can be shifted to the end of the chain, protecting the project due date (the real due date) from variation in the critical chain tasks. This concentrated aggregation of buffer is called as “**Project Buffer**.”

Our critical chain will now look something like this:

![Critical Chain Diagram](image)

Next possible question that we have to deal with is: What happens with the non-critical tasks? Whatever we discussed so far holds true for critical tasks only, which directly impacts the project timeline. But what if non-critical tasks take longer than expected and runs into the buffer? Won’t it affect the timelines of critical tasks?

Well, I haven’t forgotten about them. For non-critical tasks we still stick to the buffer management approach. Padded buffer is used to absorb any delay caused by the non-critical tasks. This buffer is called as **Feeding Buffer**. The below approach is also called as **Critical Chain Scheduling**.

![Feeding Buffer Diagram](image)

Now we have two buffers which protect us against the project promise of delivering on time. To summarize the TOC approach for Critical Chain, I explain the approach briefly below.
• The Critical Chain Schedule avoids expansion from Parkinson's Law by eliminating due dates and allowing us to take advantage of early task finishes.

• This schedule is also protected against untimely availability of critical resources by the alerts of work coming from preceding tasks.

• The project promise is protected from variation (Murphy) in the critical chain by the project buffer and the critical chain is protected from variation in non-critical work by the feeding buffers.

**What’s in it for the Managers?**

Well, managers can simply count on everything provided by Theory of Constraints. Even the attributes helping team-members also prove useful to managers, since it provides him ease of management and better productivity.

• Most important advantage is the reason for which TOC was introduced in Project Management – handling Parkinson’s and Murphy’s Law effectively. When the manager is assured that the project is free from various ill-effects such as delay, lack of quality, complexity of multiple tasks; he is in much better shape to make meaningful decisions.

Thus, this point can be concluded as: TOC helps managers to make better decisions by taking away the risk factors by some extent.

• Each project has to be customer centric. By buffer management technique, managers try to minimize the extra overhead which is otherwise introduced in conventional critical chain process. By removing the task level buffer and padding it at the very end of critical chain, managers are protecting the date that matter the most – the final project due date.

There are greater chances that project will not consume the entire buffer that is padded at the end. This is because; buffer will be consumed only when something wrong happens (as per Murphy) and the tasks take longer duration than expected. Also, to consume the entire buffer, you seriously need to do a lot of things wrong.

This way, we not only protect against the project timelines, but we are a better shape to deliver the project earlier than expected. And how much value does it add when it brings a smile on the face of the customer.

• One good feature of TOC is avoiding multi-tasking. Resources which were working on multiple projects or multiple tasks previously are now required to work on a single project, on a single task at one point of time.

Another good feature is the daily interaction which TOC suggests. This approach says that team should meet for a short duration of time, on a daily basis. This is to resolve any issues and conflicts as soon as possible.
The above first feature provides ease of management to the manager. It's always easier to do a single task in a systematic way rather than handling multiple tasks and trying to finish them all at once.

The latter feature helps manager to achieve the much needed interaction among the team members. This quality goes ahead to form the bigger attribute of team spirit. And I guess it's nothing more important to a manager than an actively interacting team. Once the team is managed properly, the projects get managed automatically.

I am pretty sure you can find plenty of strong and useful points about TOC, once you dig deep into it. It is like you look for Gold and the only thing you can find is Gold.

**What’s in it for the Team Members?**

Like every great ideas, TOC also received a cautious look from the crowd which it benefits the most, the project Team Members. Since one of the attributes of TOC is to finish the assigned task at the earliest, project members began to believe that they will be exploited and made to work extra. They thought to resist the change brought about by TOC by thinking that they will be questioned daily, if the team meeting happened daily. Here are few questions for those people:

- How can you work extra if there is no Parkinson’s Law (Work expands to fill the time) involved?
- How can you be exploited and made to work more if Murphy’s belief (Whatever can go wrong, will go wrong) is nullified?
- How can you be questioned in daily meetings, if there are no task due dates available?
- Do you really think that daily meetings are for questioning the employees or to help employees resolve & answer the questions?

The only reason that TOC has delivered countless numbers of booming projects is that it has been successful in building strong project teams. Below are few of the blessings of TOC, on which team members can count on.

- Gone are the times when employee used to work on single slice of the project. In conventional system, employee never understood his role, neither the importance of his task in the project.

  TOC has brought the integrated view of the entire project, which is also available to the team members. Not only team can see the impact of tasks on each other, but understand the importance of one's task in the entire project.

- Since TOC clearly mentions avoiding multi-tasking, team members have everything to smile for. This not only means the freedom from multiple hanging tasks at one’s desk, but also reducing the overhead of learning and relearning different things simultaneously.
Manager knows what each member of the team is engaged with, at particular point of time. This helps him too; to resolve the problems that team member might face because of the individual task assigned to him.

Performing single task not only simplifies the employee’s situation, but also helps manager to address the problems in a better way. This surely sounds like a win-win situation for the project.

- There is a clear separation of the individual tasks. In other words, there is a prioritization of tasks in the projects (Critical tasks).

Due to this, each team member know what he is supposed to do next, which tasks should follow which task, and which task is more important to finish first.

Since the high priority tasks are clearly marked, they are bound to finish first and there is less rework. This leads to better quality in the project as well as in the individual tasks. Naturally, people benefitted from this the most will be team members.

- There are no due dates for individual tasks, as per TOC. This means that team members are expected to finish the task in the earliest feasible time period. And who is in a better shape to give this estimate than they themselves?

According to TOC, team members are supposed to give the estimates of their tasks, of course with manager’s help if required. This way, no team member complains that he is being exploited or asked to finish the work in unrealistic time; because the person who has set timelines for his task is he himself.

Another useful aspect here is the exposure to planning that an employee gets. Each team member is literally planning for his own task; he is setting expectations and giving out estimates. This early exposure to managerial activities allows him to grow quickly.

- If you have started feeling that there are enough of good things about TOC, there is one more here. Daily or frequent meetings suggested by TOC, is one of the best ways to come closer to the team. Work becomes so much interesting and worthwhile, if you get to know the people you work with.

There is no need of external team building event or counseling, if the collaboration among the team members grows naturally by daily meetings.

Apart from team spirit, these meeting do help to resolve the most difficult of the problems in an easier and rapid fashion, by brainstorming and by experience of the team members in their own tasks.

“Anything that can be changed will be changed...until there is no time left to change anything...”