Dr. Sourav Basu

Rethink.
Challenge.
Solve.

A PLAYBOOK ON

How to Solve

The Late Project Problem.

FOCUS-&-FINISH
THE SCIENCE OF PROJECT DELIVERY

World over, it is quite common for projects to suffer from time and cost overruns. There are multiple theories about getting project execution under control. There are also many known best practices and experience-based wisdom (Rules of Thumb). Some are commonsensical, while others are counterintuitive. And then there is a technology that holds the promise of an inter-connected, real-time project ecosystem. But none of them seem to be adequate or robust enough to withstand the realities that the project encounters. Persistent de-synchronization, unreliable projections, day-to-day delays, frequent changes in requirements, rework, capacity limitations, inaccurate inputs, and
.flawed assumptions keep derailing the best of available systems. And, the desired control over projects remains elusive. Why? Because it is a tough problem

It is a problem where we have to execute a given scope without violating technical sequences, within a targeted duration, within a fixed budget, and without compromising on quality, while the very scope will keep changing in its details, rework will be inevitable, all the duration estimates will keep going wrong and resource shortages (money, material, decision-making capacity, expert capacity, issue resolution capacity, coordination capacity, direct resource capacity, etc.) are going to appear unpredictably but relentlessly. Moreover, many stakeholders, working in silos, will have to coordinate with each other during execution, with inaccurate and delayed progress data and unreliable committed dates. That's chaos written all over.

The good news is that it is possible to control the chaos. But for that, we need to identify the right levers. And as it turns out, in projects, it is the limitation of resources that gives rise to and foments chaos. This was pointed out as early as 1959 when the first formal method (Critical Path) for planning and managing project execution was proposed. There is no denying that the well-known best practices for managing and controlling time and cost overrun in projects – good planning (scope, cost, detailed schedules), good resource estimation, good risk mitigation, minute track, stringent monitoring, and transparent visibility – are, very obvious, necessary for successful project execution. And they are also sufficient considering real-world uncertainties like scope modifications (requirements change, a new scope is discovered, rework happens) and, day-to-day delays, that plague us throughout the duration of execution.

But add the reality of facing resource shortages that we also frequently encounter, and these are no longer sufficient to guarantee efficiency, predictability, or control, because:
a. Whenever resource shortages are faced, many outcomes are possible over a wide range
b. There is no known algorithm or mechanism to prioritize resources efficiently
c. The impact of day-to-delays and changes gets amplified unpredictably

Unfortunately, over the subsequent 60-odd years, the Project Management world chose to turn a blind eye to the ramifications of resource shortages in projects and continued to espouse the Critical Path Method, ignoring the warnings of its very proponents. Planning and monitoring systems are built over this method and academic and industry practices and regulations have formalized this as an inviolable standard for managing projects. In the first 2 chapters, we have provided a detailed explanation of the limitations that the authors tried to warn us about. Though we have used only direct resources to explain the problem, the definition of resources in projects is tricky. It can also mean cash flow, material, issue resolution capacity, decision-making capacity, expert capacity, or coordination capacity. In most cases, they are difficult to even define tangibly, leave alone estimate their requirement or availability. In short, any capacity limitation that forces different stakeholders to make allocation choices during the course of execution can be called resources.

Therefore, in the real world, on top of the necessary best practices, the criteria for a sufficient and practical solution are as follows:
a.We must have a simple resource prioritization rule that all stakeholders can follow uniformly
b.The rule has to be efficient (the prioritization has to ensure near-best outcomes, at the least)
c.The rule has to be robust (it has to remain efficient even when changes and delays happen on top)
d.It should be able to generate reliable, actionable predictions (even with approximate progress data)

In other words, whenever anybody in the project has to make any allocation choices, due to limitations of any type, the above must hold. Dr. Eli Goldratt's Critical Chain method brought attention back to resources and introduced such a workable solution in 1997. The concept of 'Multitasking' was used to demonstrate the resource problem and 'Staggering' or 'Low Work-in-progress' formed the core of the Critical Chain solution. Realization adopted this theory and embarked on the Critical Chain journey in 1998 and delivered countless successful adaptations of the theory in hundreds of organizations, over the next 10 years.

Post-2009, after we started our journey in India and began to focus on large, complex single projects, we started to feel that 'Multitasking' was not the fundamental explanation of the resource problem. Projects could turn chaotic during execution even when individual resource groups were not multitasking. In addition, when it came to projects with tens of thousands of tasks and dependencies, without any prior knowledge or experience of capacity limitations, just a conceptual description was not adequate for defining how a project should be planned to avoid multitasking, how exactly staggering has to happen when resource shortages are discovered during execution, or, how reliable predictions could be generated. These were left quite open to interpretation and only a handful of people around the world seem to be capable of doing it right.

Without a well-defined and repeatable grammar, its other solution elements (like Buffering) were reduced to rules that were impractical and often, inexplicable. They then needed to be imposed, often defying the common sense of experienced managers, and eventually, become useless paper exercises for reporting without any actionable value.

WHAT IF OUR BEST ATTEMPTS TO MINIMIZE TIME AND COST OVERRUNS IN PROJECTS ARE POINTLESS?

All over the world, the incidents of time and cost overruns and lack of synchronization in major projects are continuing unabated. Most think that the primary problem is of visibility and monitoring. And hence, technology seems to be the proverbial silver bullet for all our project woes.

Since the 1960-s, computational abilities have multiplied in jumps. In recent times, with high-speed internet and increased mobile access, we should expect huge improvements in visibility. With APIs becoming commonplace, there is also the promise of an interconnected and aligned project ecosystem. And all of these, appear to be easy and well within our reach.

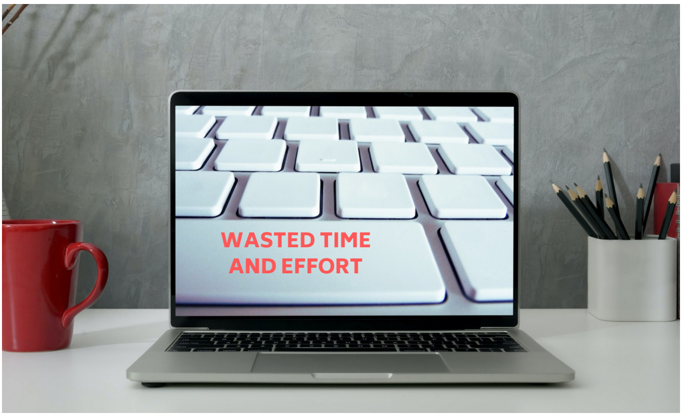
Yet, even during the last 15-20 years, time and cost overruns continue to be impervious to these advancements. Worse still – after countless man-months spent to create a unified system, organizations generally, settle for a compromised scope of semi-automatic and semi-manual processes and continue to experience the same de-synchronization as before.

But, how is it possible that technology cannot make a dent in the armor of time and cost overruns?
How are projects planned, executed, and monitored around the world? The answer is well known – we manage our projects using the Critical Path method. In this context, it might be important to refer to the original paper submitted by Kelley and Walker. When talking about how to deal with resource shortages this is what the authors had to say:

02

"All schedules computed by the technique are technologically feasible but not necessarily practical. For example, the equipment and manpower requirements for a particular schedule may exceed those available or may fluctuate violently with time. A means of handling these difficulties must therefore be sought – a method that levels these requirements…"

"…The difficult part of treating manpower leveling problem from a mathematical point of view is the lack of any explicit criteria with which the 'best' use of manpower can be obtained"



Given that projects almost always face a shortage of resources (Direct, Indirect or Managerial) and no alternate technique is available yet, is it possible that we can control project timelines by applying the Critical Path method?

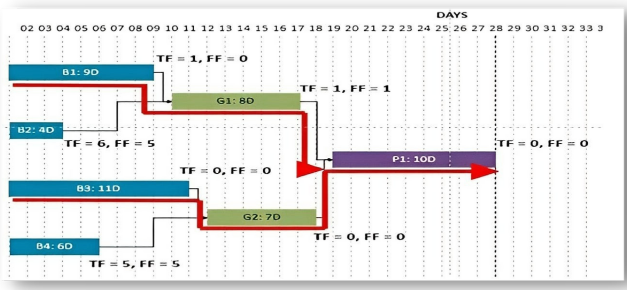
If not, then can we possibly benefit by applying layers of technology on something that does not work?

HOW MUCH DAMAGE CAN RESOURCE SHORTAGES DO TO PROJECTS REALLY?

We all know unitary method. If 10 resources can do 10 jobs in 10 days and if 5 turns up, we expect the same 10 jobs to take 20 days. The impact is known and predictable. So why should we worry about facing shortages in project situations? We should always be able to predict the impact and take recovery actions. The reason why Walker and Kelley highlighted this as a major problem, in their seminal paper on the Critical Path method, is because the above does not hold true when it comes to projects.

What do we do when we face resource shortages in projects? The answer is a no-brainer – we look at all the available work and prioritize and postpone some. Some prioritize based on critical (in terms of time) over non-critical, some look at best cash flow opportunities, some consider the risks involved etc. What may not be apparent is that each choice has a different impact on the rest of the project. If we cannot predict the choices that individuals are going to make, it is not possible to make reliable predictions about the fate of the project. And, prioritization choices are made by every stake-holder (internal or external), again and again, over the entire life cycle of the project.

If the above holds true, do you think reliable predictability is ever possible in projects facing resource shortages?

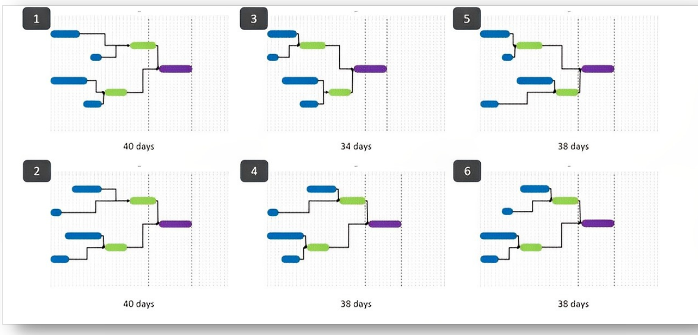


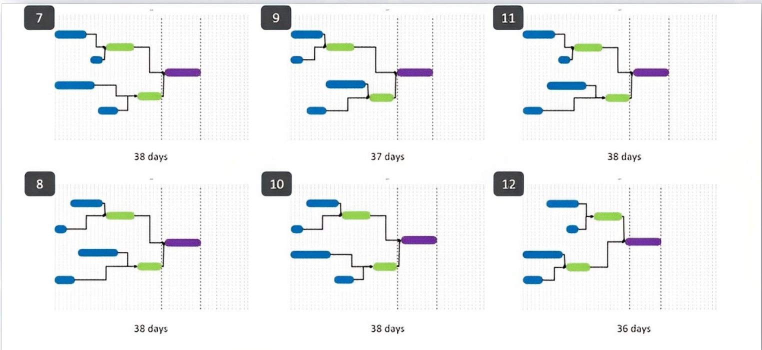
To test this, consider a simple example of 7 tasks and 3 different resource types, as shown in the picture: ***Each task color represents a different type of resource.***

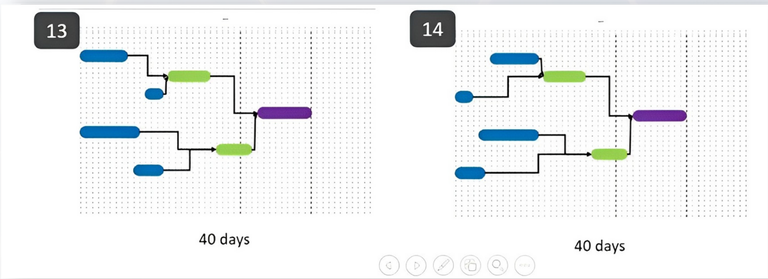
If we have to execute the project in the planned duration of 28 days, it is obvious that we need 4 Blue resources, 2 Green resources and 1 Purple resource.
(The Total Float and Free Float values are displayed in the picture)
Imagine that during execution we are able to mobilize only 2 Blue, 1 Green and 1 Purple resource

1. How much time will it take by the Critical Path logic (Float sorting, ascending)?
2. How many possible ways are there to execute?
3. What is the best time in which the project can be executed?
4. Is there any known logic by which we can arrive at the answer of (iii)?
5. What is the range of possible answers (Maximum delay/ Minimum delay)?

Possible ways to execute:







1.Answers to the questions posed in the section above –

2.How much time will it take by the Critical Path logic (Float sorting, ascending)? *40 days (Option 1)*

3.How many possible ways are there to execute?

Theoretically, 24. However, if we assume rational decision- making, then 14

4.What is the best time in which the project can be executed?

34 Days (Option 3)

5.Is there any known logic by which we can arrive at the answer of (iii)?

No

6.What is the range of possible answers (Maximum delay/ Minimum delay)?

Due to 50% shortage of resources
–Best case: 34 days - delay of 6 days over the 28-day plan
-Worst case: 40 days - delay of 12 days over the 28-day plan.
The worst case incurs 2 times the delay of the best case

Additionally, most of the possible outcomes are closer to the worst. Only 1 of the 14 is close to the best

What we can conclude from the results are the following:

- *As mentioned by Kelley and Walker, Critical Path method does not guarantee optimal solutions*
- *If we apply it nevertheless, it can multiply the delays (causes 2 times the delay in this case)*
- *Depending on the choices made, there is a different impact on the project*
- *There is no known logic by which we can ensure an efficient resource allocation*

If the above is true, what do you think happens in real projects with thousands of tasks and hundreds of resources when shortages are faced left, right and centre?

Are the local choices made by different individuals all throughout the life cycle of the project going to allow synchronization, control, or predictability?

HOW WE CAN SOLVE THE LATE PROJECT PROBLEM

Since the beginning of formal project management, projects have been suffering from time overruns, de-synchronization and lack of predictability. Even leaps in technology has not been able to counter this. What could be the reason for this inexplicable trend?

One of the usual external factors that almost every project has to deal with is “Resource Shortages” – mobilization gets delayed, required numbers are hardly ever available, scope additions and other delays increase requirement Management and Expert capacities are unknown etc.

One of the obvious management reactions to shortages is prioritization. What was shown in the earlier article was that prioritizing when facing a shortage, has a wide range of unpredictable outcomes and in most of them, delays multiply and synchronization is lost – ***exactly the same symptoms that projects, as a whole, exhibit.***

Only a very few of the prioritization decisions lead to vastly more efficient outcomes, but they follow no known rules and obeys no known logic that we can possibly deploy.

Critical Path method, as forewarned by its proponents 60 years back, is ineffective too. Even the best plans go astray, even the best project management teams cannot fire-fight enough.

Without fixing this, how could we ever get projects under control, especially when resource shortages are so omnipresent in projects?
Why are we ignoring other uncertainties - some would argue? Uncertainties like estimation errors, day-to-day delays, changes in scope, etc.?

Believe it or not, they are way simpler factors to deal with. They cannot hurt predictability. Because every such event has only a single, definite outcome. We can always plan and recover from future tasks.

If we have unlimited resources, these can be countered through textbook planning and control or sheer experience.

Have you ever heard of a late project that had resources (including money) poured into it?

But when resources are limited/ inadequate, these factors pile up on the chaos and amplify the impact.

The complexity of Resource Shortages

1.It has no impact on individual tasks but generate invisible dependencies between tasks that are, otherwise, unrelated to each other in every conceivable way.
2.Resource dependencies, unlike technical ones, can possibly run in any direction.
3.Before prioritization has happened it has many possibilities. Till someone prioritizes we cannot know which one is going to be true.

These are the reasons why predictions tend to become so unreliable. Even the best planners cannot possibly model these possibilities in a plan – there are just too many. And they have no way of knowing how much shortage individual groups might be facing and how they are going to prioritize their work as a result of that.
Even if they did, remember, there are no known rules for good prioritization – it is not possible for anyone to know whether by prioritizing according to criticality, cash flow or risks etc., he or she is benefitting the projects or causing more harm. Even if one could put accurate values in a scheduling or optimization software and run it, it is not possible to conclude whether the answer it churned out is a good answer or not. These are important considerations to take note of before we castigate planners for ‘poor planning’ and explain away project delays.

ARE WE THEN FACING A DEAD END?

Thankfully, NO. If we look at the nature of resource dependencies and how it affects projects, we can find a way to avoid most of the possible detrimental outcomes.

We cannot still claim that this is the best outcome for our projects, but we can definitely say with confidence that this is very near the best that can possibly happen.

And the quantum of improvement in terms of the only timeline is a lot (> 20%). In addition, it ensures synchronization and makes predictions very reliable during execution.

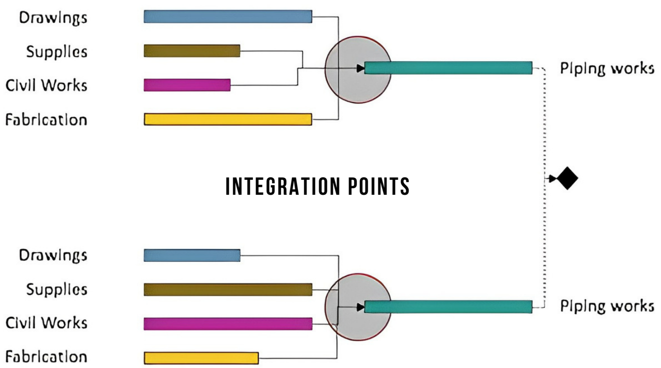
The solution is also very simple – only requires a change in the way we conceptualize projects.

Sounds too good to be true? Let’s figure out how.

HOW ARE GOOD AND BAD SOLUTIONS GENERATED?

Resource shortages are not harmful on their own. They only generate many options, but the impact is constant. E.g. If there are 3 tasks A, B and C and only 1 available resource to execute them, then they can be executed in ABC, ACB, BAC, BCA, CAB, CBA sequences but the time impact is constant – the sum of durations of A, B and C.

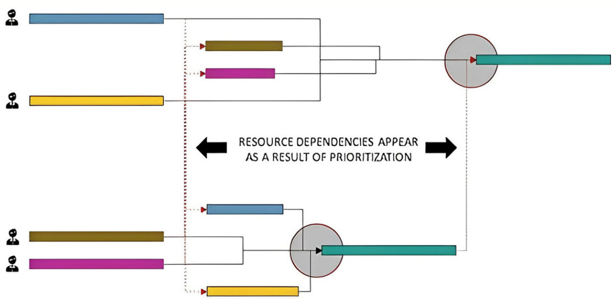
However, in projects there exists Integration Points
and together, they act as partners in crime. Together, they produce many possible outcomes.



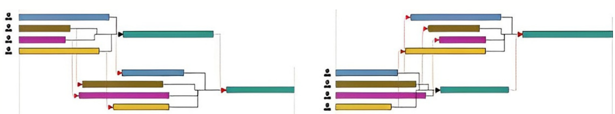
Consider the picture above – the red circles show the Integration Points. These are the places in projects where synchronization is necessary. The project team has to coordinate drawings, supplies, civil and fabrication to make sure Piping can happen.

Now, if every stakeholder (Engineering, Supplier, Civil, Mechanical and Piping) has enough capacity to only support one of the 2 tasks, how could we execute this mini project?

For example, if each of them considers what is critical for them individually, and postpones what seems to have time, this is what happens:



Since each of the stakeholders can have 2 options, there are 16 possible solutions to this problem and the ones below are the 2 best outcomes:



In fact, out of the 16, there are 14 bad solutions and only these 2 good ones.

THE KEY TO FINDING THE SOLUTION TO ON-TIME PROJECTS

When we try to execute any project, we can expect to face resource shortages – we just do not know where we might face it, when it might strike and what will be the quantum of the shortfall. How then could we plan our projects and what could be the mechanism for prioritization that would make them impervious to the bad possibilities, whenever we face the eventual reality?

Inference:

1. Resource shortages are not harmful on their own. They only generate many options, but the impact is constant. E.g. If there are 3 tasks A, B and C and only 1 available resource to execute them, then they can be executed in ABC, ACB, BAC, BCA, CAB, CBA sequences but the time impact is constant – the sum of durations of A, B and C.

2. However, in projects there exists **Integration Points** and together, they act as partners in crime. Together, they produce many possible outcomes.

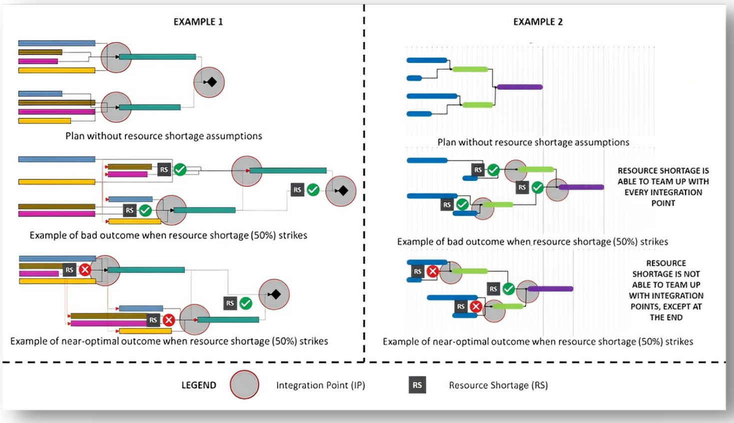
HOW TO GET PROJECTS UNDER CONTROL EVEN IN THE FACE OF RESOURCE SHORTAGES

WHAT ARE WE TRYING TO OBSERVE?

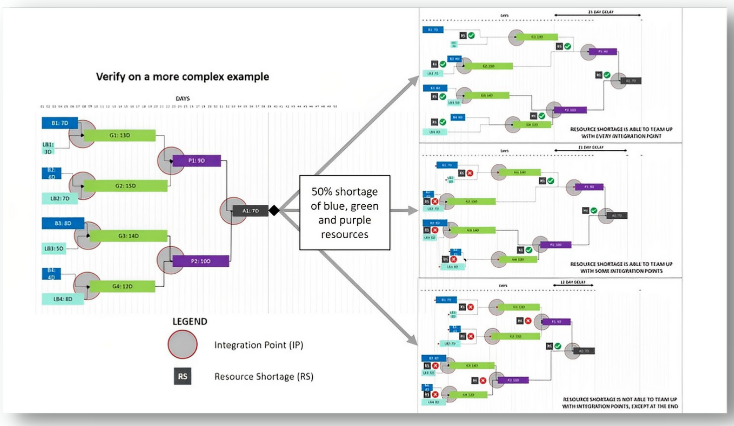
What we are trying to do here is sort of reverse engineering – isolate the best solutions by enumerating all possible prioritization combinations and try to look for clues that can get us to a general prioritization rule that can be applied in any real resource constraint situation

In the following part, we will try to compare good and bad impacts of prioritization on multiple examples and see if we can spot some cause-and-effect relationship

I will let the following pictures speak for themselves.



Simple examples showing the interplay of Resource Shortages and Integration Points



A more complex example showing the interplay of Resource Shortages and Integration Points

WHAT ARE THE POINTERS FOR AN EFFECTIVE INNOVATION THAT CAN REVERSE THE TREND OF TIME-OVERRUNS?

The ray of hope is the following idea:

Even if there is a resource shortage in the project, it is possible to execute it in a way that Resource Shortages never get to meet the Integration Points till the very end.

And if they cannot team up, they cannot multiply delays
So all we need to do is to figure out:

1. A Planning Method that sets the project up for facing real-world constraints
2. A uniform Prioritization Mechanism that can keep resource shortage and integration points separate, during execution

To summarize, we need a new method that is simple yet resilient to all real-world constraints/ challenges and can deal effectively with whatever reality eventually throws at us

FOCUS-&-FINISH:

THE NECESSARY AND SUFFICIENT SOLUTION TO DELIVERING PROJECTS ON-TIME

Abstract: There are some well-known best practices for managing and controlling time and cost overrun in projects – good planning (scope, cost, detailed schedules), good resource estimation, good risk mitigation, minute tracking, stringent monitoring, and transparent visibility. These are, very obviously, necessary for successful project execution.

And they are also sufficient considering real-world uncertainties like scope modifications (requirements change, a new scope is discovered, rework happens) and, day-to-day delays, that plague us throughout the duration of execution.

But add the reality of facing resource shortages that we also frequently encounter, and these are no longer sufficient to guarantee efficiency, predictability or control.

Over the last few articles, we have highlighted the difficulties of handling resource shortages in projects –

1. Whenever shortages are faced the outcome is unknown and has a wide range
2. There is no known way to prioritize resources efficiently
3. The impact of day-to-delays and changes gets amplified unpredictably

We have highlighted that the project management methods (e.g., Critical Path, Resource Constrained Project Scheduling etc.) known to us are mostly detrimental for practical application and there are no known ways to avoid the many possible bad outcomes that applying these methods can lead to.

In order to find a solution, it is important to accept the problem that requires solving – We have to execute a given scope without violating technical sequences, within a targeted duration, within a fixed budget, without compromising quality, while the very scope will keep changing, all the duration estimates will keep going wrong and resource shortages (money, material, decision-making capacity, expert capacity, issue resolution capacity, coordination capacity, direct resource capacity etc.) are going to appear unpredictably but relentlessly.

Therefore, it is quite expected that, during project execution, we are going to be forced to keep deviating from the plan. And all we can do to is keep prioritizing the scope, make efficient use of whatever resources we have and find ways to recover delays and keep clawing back.

In such a reality, a sufficient solution for handling resource shortages must be available:

1. We must have a simple prioritization rule that all stakeholders can follow uniformly
2. The rule has to be efficient (the prioritization has to ensure the best possible outcomes)
3. The rule has to be robust (it has to remain efficient even when changes and delays happen on top)
4. It should have reliable predictability of outcome

Now, we are going to propose such a solution.

What we will show is that dealing with the project reality is actually very simple. The formula for success in project execution is simply to have ample resources. Since that is not really a practical solution, barring a few exceptional cases, the next best thing is to divide it into sub-projects and ensuring maximum resource loading on as many of them as possible, inside the larger project.

But again ‘sub-project’ is a vague term. Choosing any sub-project does not help. If we have to define a definite mechanism for handling reality, we cannot leave this choice open to interpretation.

In this article, we will propose a repeatable, reproducible grammar for defining such special sub-projects – Streams, and batches for concentrating available resources – Focus-&-Finish batches and derive an overall solution that meets all the sufficiency criteria listed above. We are proposing that the key solution to our project woes lies in the modeling of the project, not in some perfect run-time optimization technique (ad-hoc or software-driven) or data accuracy/availability. We have used simulations to test the model and have shared the findings at the end.

We showed in the document earlier (Page No. 3-11) that the Critical Path method or ad-hoc prioritization cannot possibly meet the above criteria because the combination of resource shortages and integration points renders them inefficient and unpredictable.

We also observed (Page No- 13-15) that near-optimal outcomes are generated when resource shortages are able to bypass integration points throughout the project, except right at the very end.

And hence, we concluded that if we have to meet the criteria listed above, we need to find a way to plan and devise a prioritization rule that can guarantee this even when there is a shortage at the project level.

A NEW APPROACH FOR PLANNING AND EXECUTING PROJECTS FOR THE REAL WORLD

To solve this problem, instead of banging our heads against the wall trying to find an optimization solution, we are instead going to propose a modeling solution that allows us to avoid the problem altogether. And this modelling solution requires the new concepts of Streams and Focus-&-Finish Batches, instead of paths/ chains and tasks.

We will first introduce a new concept of “Streams” to replace traditional task sequences e.g. paths/ chains.

A Stream is more than a mere sequence of tasks. It can be thought of as a collection of paths, encapsulating all Integration points amongst them, that runs right up to the final integration of the project, independent of other Streams. Look at the more complicated example below:

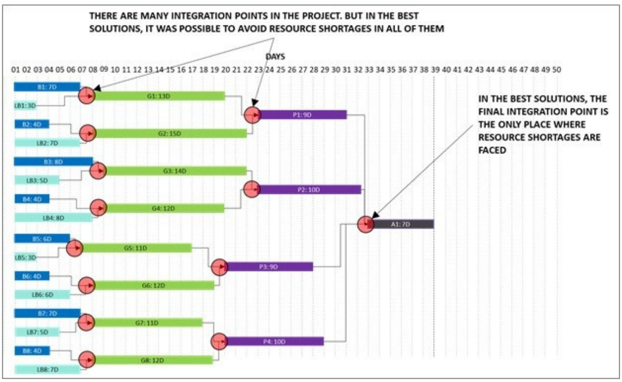


FIGURE 1: A COMPLICATED PROJECT PLAN

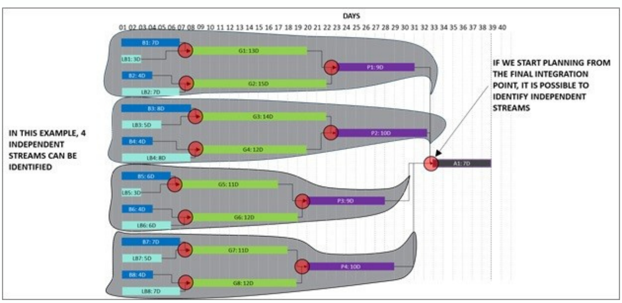


FIGURE 2: STREAM IDENTIFICATION

Once the streams have been identified, the strategy for execution is pretty simple – Whenever there is a resource shortage, make sure the available resources are used to load individual streams. Streams that cannot be loaded have to wait. This simple strategy ensures the outcome that we seek – integration points will never face resource shortages.

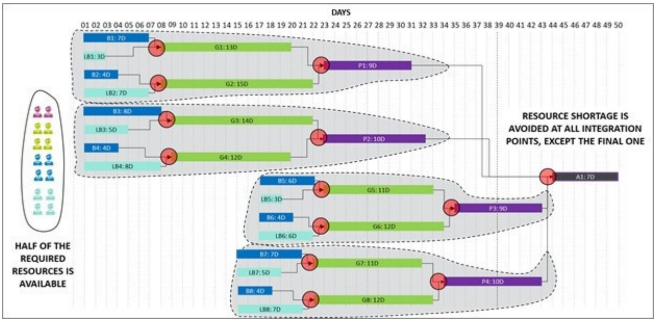


FIGURE 3: HANDLING RESOURCE SHORTAGES BY PRIORITIZING STREAMS

In essence, the streams define special sub-projects. And the simplest and the best way to control the project timelines is to make sure that we load resources on and ensure the fastest possible progress in only as many streams as we can afford, given the available resources at our disposal.

To ensure predictability with respect to which stream(s) should be chosen, if and when shortages are faced (we do not know where shortages will strike and when), we obviously will need to define stream priorities that are uniform across stake-holders.

Given that shortages require us to concentrate available resources on priority streams, we must therefore, also find a way to ensure that once deployed, resources work for the shortest possible time on a stream and can be available for deployment in the next as quickly as possible.

To ensure that, we simplify the streams further by defining a sequence of Focus-&-Finish batches of work inside each. Each F&F batch defines the resources to concentrate and the scope to be executed before they can be deployed to the next priority stream.

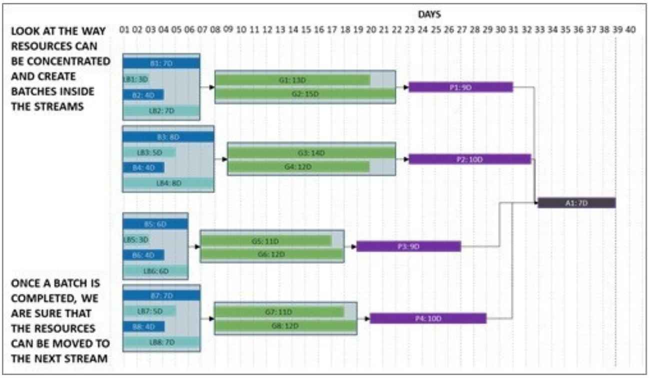


FIGURE 4: FOCUS-&-FINISH PLAN

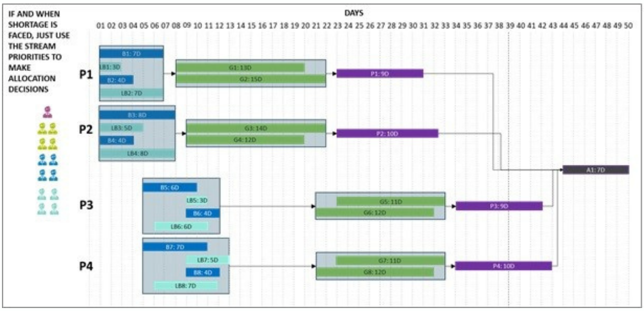


FIGURE 5: USING FOCUS-&-FINISH PLAN TO DEAL WITH RESOURCE SHORTAGES IN EXECUTION

We have thus defined the grammar for creating a simple project network (in the example, a network of 13 batches vs. a network of 29 tasks, 4 streams vs 16 paths) – A single prioritization mechanism guaranteeing predictable and efficient (near optimal/ optimal) outcomes, even in the face of resource shortages, changes and day-to-day delays.

AN INNOVATION IN PROJECT PLANNING NETWORKS: KEY TO APPLYING FOCUS- &-FINISH

Applying this method requires a new network definition for planning projects – traditional project networks based on tasks and technical dependencies cannot be used for modelling.

We need to introduce an alternate network type for modeling project plans – a 2-tier project network (Figure 4).

The top tier represents the stream and batches. And, prioritization (manual or Critical Path) needs to happen at this level.

Each F&F batch, in the lower tier, houses a local, detailed task level network. We can run Critical Path locally inside each batch but not across the task-level network (as in Figure 1).

When in execution, task priorities need to be derived from the stream – all tasks belonging to a batch will have the same priority to ensure resource concentration around all the integration points residing inside the stream.

Testing the model

We explained the theoretical basis for this new way of project modelling and execution and the expectation is that this simpler plan is a much better tool for minimizing time and cost overruns in projects.

To test our hypothesis, we have run simulations to test whether the expected benefits can be demonstrated.

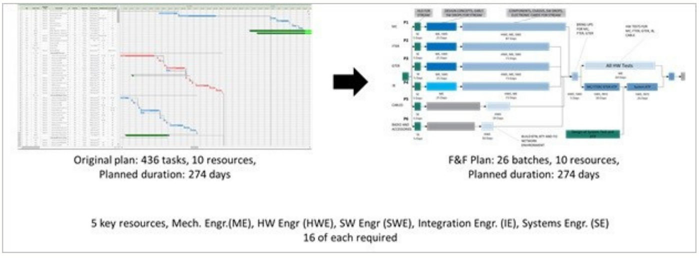
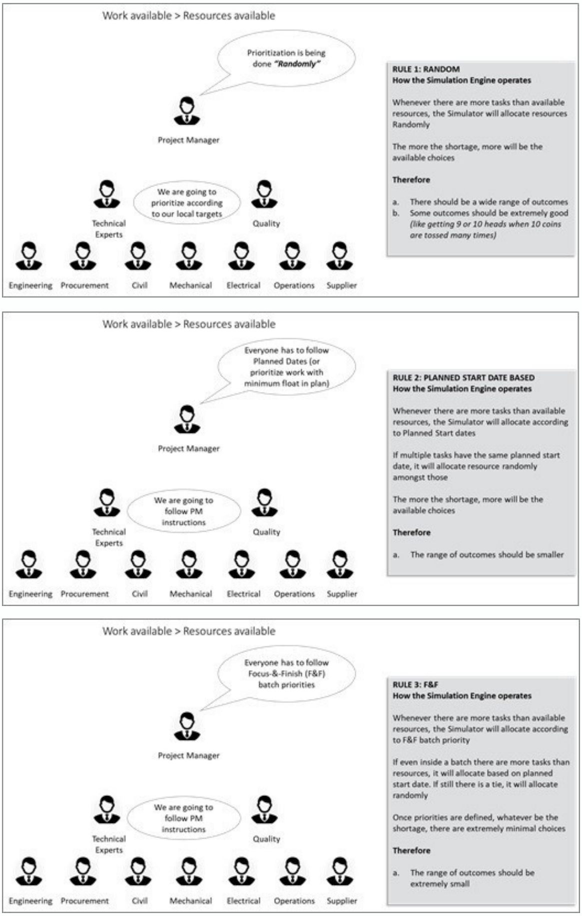


FIGURE 6: TRADITIONAL PLAN VS. FOCUS-&-FINISH PLAN

We have used the following techniques to mimic prioritization decisions during project execution:



FIGURES 7, 8 & 9: HOW THE SIMULATION WORKS

To test the performance of the different approaches we have tested them against the following cases:

The conclusions of the simulation results are presented as followed:

When there is no resource shortage, prioritization is irrelevant

1. There is no difference in the outcomes when only task durations are increased

The impact of resource shortages can be as damaging as the impact of changes and day-to-day delays

2. Resource shortages cause as much damage as increase in task durations, there is a wide range of possible outcomes in both Random and Planned Start

When resource shortages happen, F&F performs better on all parameters

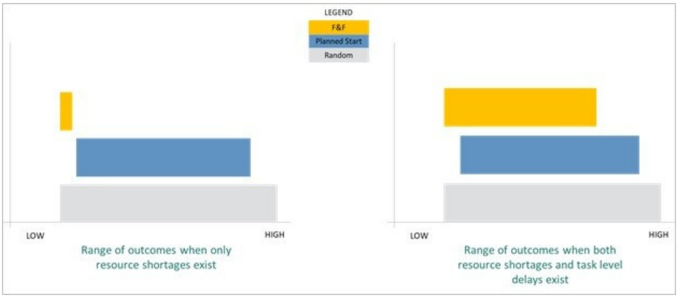


FIGURE 10: COMPARISON OF TREND OF RESULTS

i ***Average performance***

3. When only resource shortages occur, average performance of **Planned Start** is better than Random. Average performance of **F&F** is at least twice as better than **Planned Start** consistently

4. Higher the shortage, better the performance of **F&F**

ii. Predictability of outcome

5. **Random** produces many outcomes over the widest range, **Planned Start** produces many outcomes over a wide but smaller range than **Random, F&F** produces one definite outcome

iii. Efficiency of outcome

6. **Random** produces a very small percentage of highly optimal outcomes; the best outcomes of **Planned Start** are worse than that of **Random; F&F** solutions are near or better than best solutions found by **Random**

v. Robustness of outcome (how resilient is the solution to task delays)

7. Even when task durations increase in addition to resource shortages, average performances demonstrate the same trend as 3

8. The best solutions of **F&F** are near or better than the best solutions of **Random**; the best solutions of **Planned Start** are worse than both

9. Because of variations in task durations, all 3 generate a wide range of solutions, but **F&F** solutions are able to avoid the worst outcomes of the other 2 and are confined to an optimal zone.

Conclusions

We have presented a simple solution for executing projects that allows us to control the timelines in the real world. It is not an optimization solution for projects – it does not promise a theoretically optimal answer to a complex mathematical problem. Instead, it provides a modelling solution involving Streams, Focus-&-Finish batches and a 2-Tier planning network, that guarantees a near-optimal outcome every time we execute a project in the real-world and is easy to plan, monitor and control.

Using these, it ensures that whatever the real-life complexities that emerge, there is a simple practical approach that can be implemented in project organizations and ensure an outcome that is pretty close to the best possible.

While planning, all it requires is to divide the entire project, however large and complex it might be, in to independent Streams (a special type of sub-project) that are independent from each other.

And the only efficient, predictable and robust way of handling resource shortages, changes and day-to-day delays are to provide ample resources to as many streams as possible and holding the rest.
Other useful aspects of this approach are that

a. Prioritization of Streams is an easy concept in execution. There can be confusion over how to prioritize. The good news is that prioritization can follow the Critical Path logic or can also be fixed manually based on the project need. There is not much difference in outcomes whichever way the streams are prioritized

b. Stability of this structure is key to creating contracts with external parties, preventing priority switching during execution, enabling a single version of truth and allowing digitization of the project eco-system.

There are a few important considerations that should also be highlighted.

This way of planning does not in any way dilute the importance of best practices like good scoping, validation of durations, resource estimation, risk mitigation etc. during planning or detailed tracking, review processes, full kitting etc. during execution. Instead, it enhances the effectiveness of these practices and makes them more relevant.

It also does not dilute the ongoing optimization efforts that good managers on the ground deploy from time to time. Again, it provides a stable starting solution and a guiding framework on top of which these efforts can become more effective.

Some people often point out that it is already well accepted that a high-level plan is a more effective tool for better project performances than detailed plans. Some also point out that the staggering of the green batches that we had shown in Figure 5 is what Critical Chain has already proposed.

It is important to understand that the definition of a Stream is by no means just 'high-level'. The term 'High-level' is a vague description for a plan. If we can't define how 'high' it should be, or, what should be the logic used to determine a high-level batch, it is open to interpretation. Any high-level plan cannot prevent the interplay of resource shortages and integration points during execution.

The second problem is even having the right high-level plan is not foolproof enough, in order to execute it we must have a way to define the detailed scope inside the high-level tasks. Again, a 2-Tier planning network definition is necessary for doing that – the prevalent Level 1, Level 2, Level 3 schedules that are generated in practice today, provide no means to work and monitor around the high-level plan once the project goes in to execution and real-world disturbances are faced.

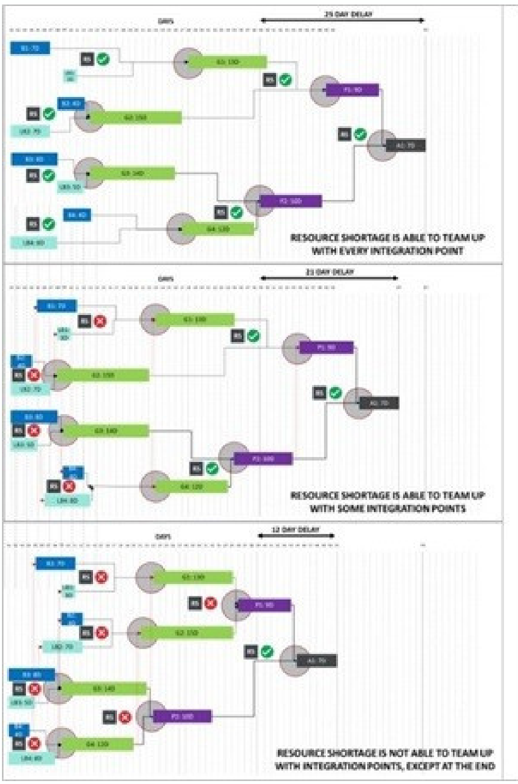
As far as Critical Chain is concerned, it has introduced the concepts of multitasking, staggering and WIP control and highlighted the risk of integration points. It has described how to find the Critical Chain and has proposed the idea of subordinating all other resources to the critical resource. A combination of all these ideas, in special cases, could lead to the same solution as the F&F solution proposed here, especially in case of a multi-project environment.

However, the exact solution is quite open to interpretation. Even when all of its solution elements are being complied with, it is still open to possibilities of resource shortages at integration points.

Especially when it comes to large, complex single projects, where resource shortages and constraints cannot be ascertained from before and different resources are activated at different points in time, it is very difficult to apply the concepts without knowing the Streams.

E.g. The 'staggering' of work for the green resources in Figure 5, could have been done in many ways – without the definition of streams and F&F batches, it is not possible to know HOW the staggering has to be done.

`Look at the examples we had shown in earlier articles. In none of the cases are individual resource groups 'Multitasking'. The work has automatically gotten staggered for each individual resource type. Yet, this does not stop the delay multiplication due to the combination of resource shortages and integration points. All 3 cases shown here, are CCPM compliant. The last case is also the Stream-based solution.



Without defining what Streams are, Staggering, Multitasking, WIP control are terms that are open to interpretation. Without defining Focus-&-Finish batches, constituents of a Stream can easily get separated during execution.

Also, when resource availability is something that is discovered along the way, there is really no basis for deciding a WIP limit that Critical Chain postulates.

However, once Streams and F&F batches have been defined, WIP control and Staggering are natural outcomes of the shortages that we might facing.

Again, we would argue that once we have applied the planning logic proposed in this article, it only strengthens the Critical Chain concepts. Concepts of Full Kitting, Buffering are very effective means for managing timelines and mitigating risks, and can become immensely valuable when applied on top of the F&F Batch level plan (Figure 4).

To summarize, the method that has been highlighted here does not undermine the industry best practices in any way – be it traditional techniques, IT-based enhancements or new management philosophies like Critical Chain or Agile etc. techniques. It provides the framework on top of which each of these can become truly effective in managing complex projects.

Dr. Sourav Basu

DIRECTOR, INDIA BUSINESS OPERATIONS
REALIZATION TECHNOLOGIES