Under pressure to develop more products while holding their budgets constant, New Product Development (NPD) organizations are operating under severe resource constraints. Little margin for error remains in allocating and controlling capacity under these conditions.

By the same token, developing more products faster is the NPD key to increased competitiveness and higher profits. Therefore businesses that can leverage their available NPD resources better can reap substantial rewards.

Most initiatives for improving speed and productivity in NPD to-date have focused on projects: whether it is through stage-gate reviews, team organization or project accounting. Unfortunately, not only do they require all project participants to change their behavior, but also fail to serve the needs of NPD managers.

What NPD managers want is the ability to allocate their resources to the most profitable opportunities, and set stable priorities for all project participants – based on the global impact of their decisions, and the local actions required to support those decisions. For example:

- At the portfolio level, it is impossible to accept a project on its own merits. What other projects will have to be given up so that resources can be released for a specific project? What are the resulting business implications? What is the optimal portfolio?
- The most common complaint of a project manager during execution is that resources are not available when needed (even when promised) – somehow required people are always working on other projects. "Visibility" is not enough. Project managers need assurance that their project will get the required resources in a timely manner.

This paper presents how finite capacity based planning and execution can serve the needs of NPD managers.

Why focus on capacity

Portfolio and project decisions all have to be made within the confines of finite capacity, and NPD capacity has become the major limitation to developing more products, faster.

The capacity of an organization at a point in time is finite … and always has been. What has happened in the last few years is that businesses have been pressured to develop more products, while holding capacity at the same levels. As a result, they are reaching the limits of their finite NPD capacity. Capacity now underpins all decisions – from deciding which products to fund, creating technology roadmaps, to planning and controlling operations:

I. Portfolio Selection: Finite NPD capacity more than just blocks a company from pursuing all good opportunities. More fundamentally, it implies that cost-benefit-risk analysis is no longer sufficient for making portfolio decisions. For example, low cost-high benefit-low risk projects that also consume significant capacity might have to be foregone in favor of projects that can be undertaken with available capacity.

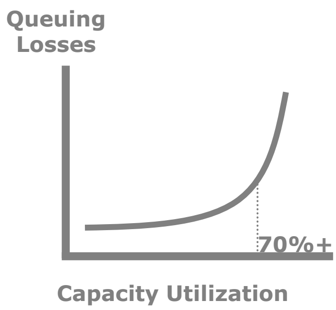
II. Technology Strategy: as we shall see later, lead times for developing new technologies are determined by capacity – and often times the best option is to use an off-the-shelf technology or extend the old platform[1], than wait for capacity to be freed up for leading edge technologies and new platforms.

[1] Platform is a collection of common technical elements that can be used for developing multiple products.

Finite NPD capacity is also forcing businesses to focus on technologies that distinguish them from competitors, and forego those that are nice to have but not critical.

III. NPD Organization: In order to maximize utilization of their available capacity, companies are increasingly creating central resource pools that can serve multiple projects. Some companies are even moving to "virtual design centers" so that they can leverage highly specialized resources across multiple business units.

IV. Development Scheduling: during *planning* resource loading has to be carefully managed because NPD operations experience substantial queuing losses beyond a certain level of capacity utilization. Overload operations by even 10% and the entire pipeline gets clogged … start 10% less projects than what NPD capacity allows and you sacrifice substantial opportunities.



During *execution* capacity should be made available to a project no later than when required, yet making them available earlier also carries serious penalties. When capacity is tight, giving a project resources earlier than when absolutely required means interrupting the progress of other projects!

It is clear that little margins for errors remain in allocating and controlling finite NPD capacity under conditions faced by management today. However NPD capacity optimization has hitherto been impossible; the next section examines why.

Obstacles to finite capacity optimization

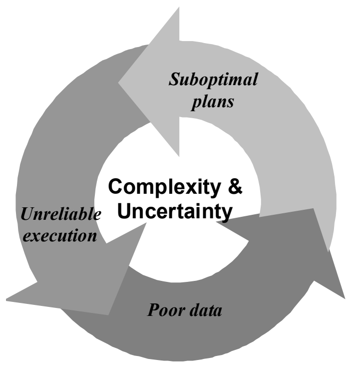
NPD capacity optimization is one of the most difficult managerial problems.

While finite capacity optimization solutions have been successfully applied in manufacturing, the following complications still block businesses from making the most profitable use of their finite NPD capacity:

I. Complex dependencies: developing products involves hundreds to tens of thousands of interrelated activities. Dependencies exist not just within a project, but also across projects: e.g.: many product projects depend on platform projects, which in turn are fed by technology projects. With such complexity, balancing finite resources becomes virtually impossible.

II. High variability: product development environments are in constant flux: markets are difficult to forecast, and capacity needs are difficult to estimate. These conditions impede management from setting stable priorities. Lacking stability the organization loses faith in any priorities at all.

Effects of variability are compounded by poor data. Not having been able to manage their capacity in a systematic manner, most businesses have poor data in regards to their NPD capacity. Time and effort required for collecting and refining required data become an additional obstacle to undertaking finite capacity optimization.



The result is a vicious cycle that NPD organizations find difficult to escape from. We need a powerful but pragmatic solution for tackling these obstacles. Such a solution is possible if we subordinate mathematics of the problem to the business needs of finite capacity optimization.

Business objectives

Before considering a solution, we need to establish what results we want in NPD.

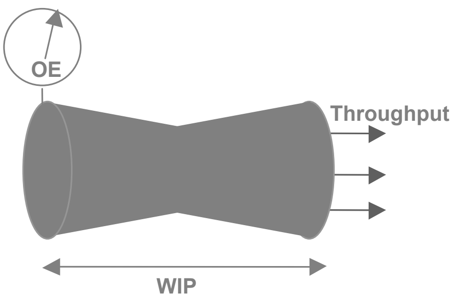
We suggest the following measurements to judge the bottom line impact of finite capacity optimization:

I. New Product Throughput (T): Throughput is the rate at which product development creates future profit streams for the business.

Throughput is calculated only for projects that will generate revenue: a) products that are ready to be manufactured, and b) technologies that can be licensed to other companies. Technologies that will not generate any revenue without further consumption of NPD capacity are not included in throughput calculations.

T = rate at which projects will be completed x their risk-adjusted Net Present Value (NPV).

II. Operating Expense (OE): OE is the rate at which money is spent to operate the pipeline capacity. It includes all departmental and overhead expenses, and excludes project-specific expenses.



In order to operate a given level of capacity, an NPD organization has to maintain a baseline OE that is independent of project-specific decisions.

III. Work-in-progress (WIP): WIP includes ideas, concepts and technologies that have entered the NPD pipeline, but have yet to be converted into throughput.

The indirect impact of reducing WIP far outweighs its direct impact:

a. WIP delays processing of new ideas. Cutting WIP cuts lead times.

b. WIP interferes with pipeline flow, causing delays and "multitasking". Cutting WIP improves the effective capacity across the pipeline.

As we describe the solution in the following sections, we will also explain how it allows managers to start from baseline capacity, and create and execute plans that maximize Throughput with minimum WIP and minimum additional OE.

Finite NPD capacity planning: maximizing throughput, minimizing WIP

The inescapable reality of finite capacity can be used to optimally prioritize and sequence projects.

No matter how many activities need to be completed, how complex their interrelationship is, or how many different resources are required, flow of projects through the NPD pipeline is governed by its bottlenecks, i.e., resources that have least capacity compared to the demand placed on them. As a result:

- Throughput of the pipeline equals throughput at the bottlenecks.
- Releasing projects in violation of bottlenecks' capacity creates unnecessary WIP.
- Capacity (and OE) at non-bottlenecks has to support throughput at the bottlenecks.

The following is how we exploit these inescapable facts to easily create optimal plans:

I. **Optimize project portfolio:** In an unconstrained environment, the rule for selecting projects is very clear. The correct decision is to take a project on if it has positive NPV. But let us assume that there is a finite capacity constraint imposed on the business[2]. How should the NPV rule be modified to consider projects?

As an illustration, Table 1 shows five projects. With no capacity constraints we would accept all five because they all have positive NPV. Now suppose there is a test lab whose capacity is required by these projects for a total of 85 weeks, but only 50 weeks of capacity is actually available.

If we prioritize projects according to their individual NPV, we would accept project 1, skip project 2 (because it needs more than remaining capacity), and accept project 3. Portfolio throughput would be $75,000,000.

But we can increase total throughput to $105,000,000 by accepting projects 5, 2, 4 and 3. These projects have the greatest combined NPV among those combinations of projects that use no more than 50 weeks of test lab capacity.

The logic leading to correct decision is formalized by prioritizing projects on NPV per unit of constraint's capacity required for each project (constraint-indexed NPV).[3] Using constraint-indexed NPV leads us to first accept project 5, then 2, 4 and finally 3.

Table 1: Illustration of constraints-based portfolio optimization

Project	Net Present Value (risk-adjusted)	Capacity required at test lab	Decision with simple NPV	NPV per unit of test lab capacity	Decision with constraint-indexed NPV
1	$50,000,000	35 weeks	Select	$1.43 M/ week	Discard
2	$45,000,000	20 weeks	Discard	$2.25 M/ week	2nd choice
3	$25,000,000	15 weeks	Select	$1.67 M/ week	4th choice
4	$20,000,000	10 weeks	Discard	$2.00 M/ week	3rd choice
5	$15,000,000	5 weeks	Discard	$3.00 M/ week	1st choice
		Portfolio Throughput:	**$75,000,000**		**$105,000,000**

[2] These are also called constrained capital budgeting problems (reference: "Financial Theory and Corporate Policy", 3rd edition, by Copeland and Weston, Page 55). The constraint could be availability of budgets, or capacity, for example.

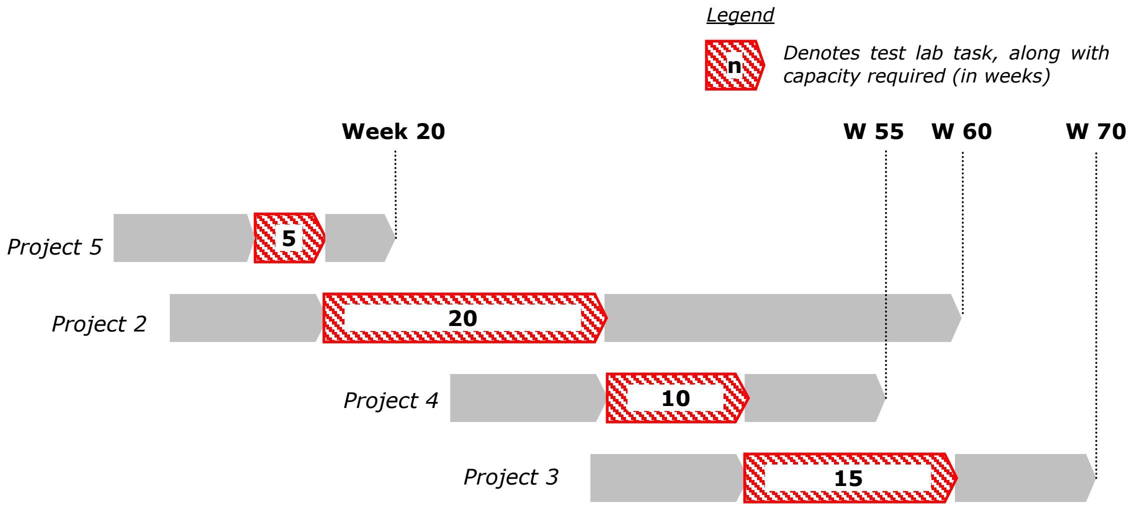
[3] Use of constraint-indexed NPV yields the same results as linear programming optimization. Therefore, not only is using constraint-indexed NPV simple, but also optimal.

II. **Optimize pipeline**: in unconstrained environments, lead-time of a project equals its critical path. However in capacity-constrained cases, project schedules and lead times depend on when capacity is available at the constraints.

Continuing with our previous example, once we select Projects 5, 2, 4 and 3, what should be due-dates of those projects. For simplicity assume that testing lies on the critical path of each project. Table 2 and accompanying figure show how test lab is loaded and, then, how project due-dates established.[4]

Table 2: Illustration of constraints-based due-date quotation

Project	Pre-test lab duration of critical path	Capacity required at test lab	Project schedule in test lab	Post-test lab duration of critical path	Project lead time
5	10 weeks	5 weeks	Week 11 to 15	5 weeks	20 weeks
2	10 weeks	20 weeks	Week 16 to 35	25 weeks	60 weeks
4	10 weeks	10 weeks	Week 36 to 45	10 weeks	55 weeks
3	10 weeks	15 weeks	Week 46 to 60	10 weeks	70 weeks



If projects are released earlier than when test lab has capacity, they will only disrupt the flow of existing projects – causing throughput and due-date performance also to deteriorate.

<u>Increasing the constraint's capacity cuts lead times</u>. In our example, if the company doubled the capacity of the test lab, they can finish projects 4 and 3 in almost half the time! Thus, an understanding of the constraints and their capacity also forms the basis of a uniquely valuable piece of information: tradeoff between lead times and capacity.

[4] At the planning stage it is not necessary to create detailed project schedules. Due to high variability and strong inter-dependencies in NPD, schedules will change anyway on a daily basis.

Finite NPD capacity execution: meeting project due-dates, containing costs

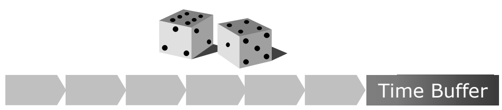
Task-time variability limits us from achieving the full throughput potential of constraints.

Task-time variability makes local delays inevitable, and due to strong dependencies among tasks those delays tend to propagate rapidly. When delays mount, even non-constraints start experiencing persistent peak loads that interrupt overall pipeline flow. As a result:

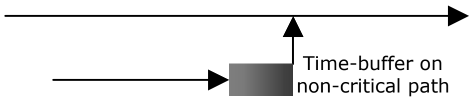
- WIP goes up.
- Project due-dates start slipping, jeopardizing throughput.
- People are shuttled randomly between tasks – reducing their productivity by 20 to 80 percent. (Knowledge workers are not machines that can be switched on, and made to produce full-stream instantly. And if switched off, half-finished work decays rapidly).

If businesses want high project due-date performance and highest possible productivity, they need to contain the effects of task-time variability. Constraints-based planning is not enough, and execution tools are required to assure highest level of global performance:

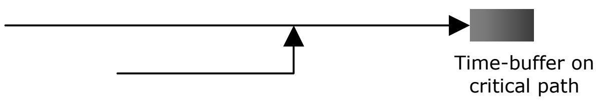
I. **Time-buffers**: these are blocks of time with no scheduled work – typically placed at the end of a set of activities to absorb variability in those activities:



- On non-critical paths, time-buffers <u>protect integration points</u>, without increasing project length.

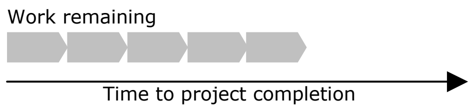


- On the critical path, time-buffers <u>protect project due-date</u> but add to project lead-time. Therefore they should be decided by those responsible for overall performance of NPD.



II. **Queue control**: even with adequate average capacity, task-time variability during execution can cause peak loads. These peak loads can cause queuing losses in the form of delayed projects and expediting costs. "Just-in-time" queue control can be used to contain those losses:

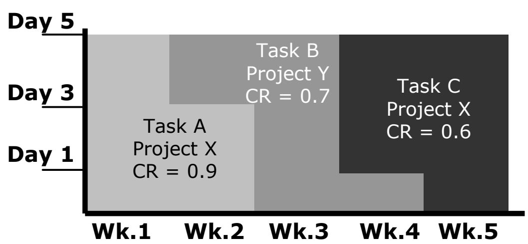
- Since it is difficult to predict actual timing of tasks, they are scheduled when they are actually available to be worked on.
- Critical Ratio is calculated for various tasks in the queue (= work remaining through to project completion ÷ time to project completion.)



- Those tasks with the highest Critical Ratio are the ones most critical to the due-dates of their respective projects, and get first priority. Table 3 and accompanying figure illustrate how to create a just-in-time schedule.

Table 3: Just-in-time queue control using Critical Ratio (CR)

Task (project)	Work remaining	Time to completion	Critical Ratio	Capacity needed
A (X)	18 weeks	20 weeks	0.9	8 days
B (Y)	14 weeks	20 weeks	0.7	8 days
C (X)	9 weeks	15 weeks	0.6	9 days



Instituting a process of ongoing improvement

Improvement initiatives should be targeted at the sources of biggest disruptions to NPD flow.

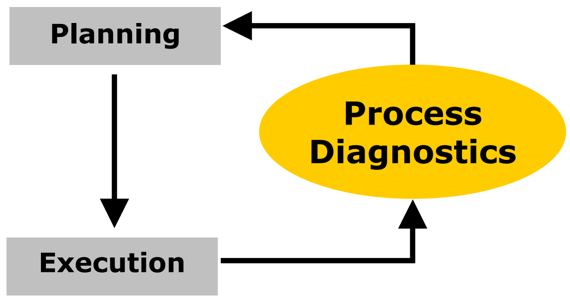
NPD has become mission-critical for many businesses, and competition is intensifying. NPD organizations are being tasked not just with sporadic improvements – but to progressively and rapidly improve their contribution to bottom line.

However, neither the time nor resources available to make improvements are infinite. Therefore managers need to clearly identify areas where a local improvement will yield immediate and substantial gains in global performance.

As discussed earlier, constraints establish the upper limit on NPD throughput. And task-time variability limits a business from achieving that full potential. Thus, improvement efforts can be directed at either reducing variability or removing the constraints. While constraints are few and easily identified, the difficulty is in selecting areas for reducing variability.

Using buffer performance history to diagnose processes

When time-buffers are used to protect schedules, we could also keep a history of which activities actually used up that protection. If we classify those activities (by the type of resources required to perform those activities; type of work they represent; and type of projects they are in), we can have the data to find the biggest sources of disruptions to flow.



Whichever areas consistently consume the highest amount of protection should be targeted for process improvement - tightening of technical processes, improvement in task estimates, deployment of computer-aided engineering tools, all can be prioritized.

Getting started despite very poor data

Data can lie, your intuition will not.

As discussed earlier, capacity and projects data is either poor or non-existent in most NPD environments. However organizations do know which 20% of their resources are most overloaded (where there is perpetual need to hire more people or out-source work).

Such intuition can be used to focus data collection and cleanup efforts (capacity data for likely bottlenecks, and task-estimates for work performed by those resources), and quickly establish a good enough model to set project priorities and realistic due-dates.

Buffer performance data from execution can then be used to progressively make the model accurate, allowing managers to perform sophisticated analyses within a few months.

1. Use intuition to pinpoint probable constraints.
2. Clean up data on constraints to reflect your intuition.
3. Rationalize due-dates for projects flowing through those constraints.
4. Create time-buffers to protect rationalized due-dates.
5. Monitor buffer performance for a few weeks.
6. Analyze buffer history to refine the model.
7. Repeat steps 1 through 6 for a few cycles.

Summary of benefits

Developing more products, faster is the NPD key to increased competitiveness and greater profitability. Finite NPD capacity is what stands in the way. We have presented a pragmatic solution to finite capacity planning and execution that provides the following benefits:

1. Boost performance through high leverage managerial decisions (not cultural change).
2. Allocate resources to the most profitable opportunities.
3. Achieve higher productivity by creating central resource pools[5].
4. Determine tradeoff between project lead-time and global finite capacity.
5. Accurately estimate how much money to spend to achieve desired throughput.
6. Contain queuing losses while providing high levels of capacity utilization.
7. Quote feasible project due-dates.
8. Set stable priorities for all project participants – assuring high due-date performance.
9. Focus local improvement efforts on areas that cause biggest disruptions to throughput.
10. Break the vicious cycle of poor data, suboptimal plans and unreliable execution.

Please contact TOCCA for feedback or additional information.

Phone: +61 2 9344 3464
e-mail: tocca@tocca.com.au

[5] It is now possible to resolve contention for resources among concurrent projects despite complex interactions and high variability, and no longer is it required to maintain artificial silos of capacity.