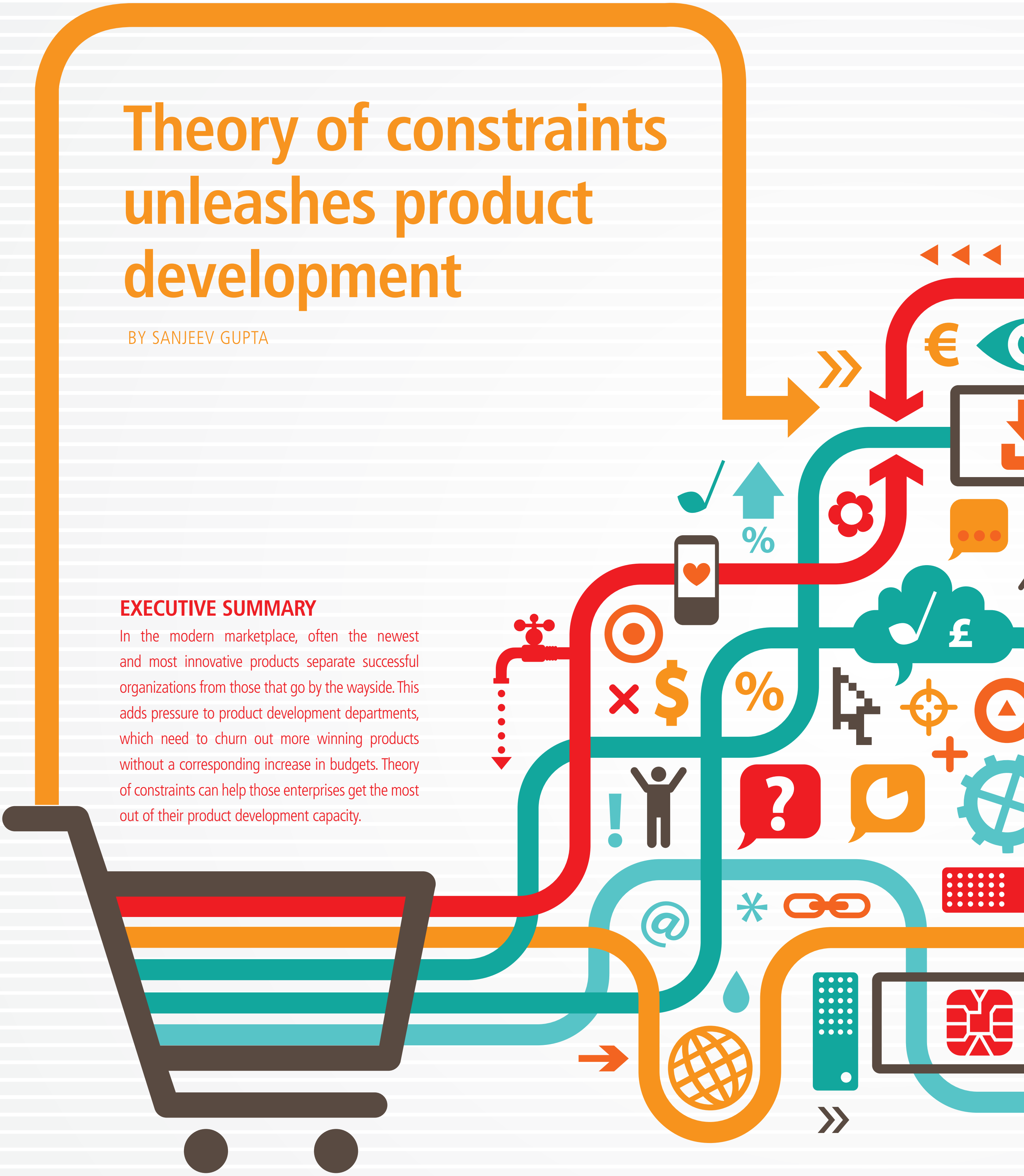
Theory of constraints unleashes product development

BY SANJEEV GUPTA



EXECUTIVE SUMMARY

In the modern marketplace, often the newest and most innovative products separate successful organizations from those that go by the wayside. This adds pressure to product development departments, which need to churn out more winning products without a corresponding increase in budgets. Theory of constraints can help those enterprises get the most out of their product development capacity.

Under pressure to develop more products while holding their budgets constant, product development organizations are operating under severe resource constraints. Little margin for error remains in allocating and controlling resources under these conditions. By the same token, developing more products faster can be the key to competitiveness and higher profits. Businesses that can get the most out of product development capacity will reap substantial rewards.

This article uses the theory of constraints to outline a practical solution for optimizing product development throughput.

Over a period of years, product development organizations have had to increase the rate at which they develop new products, all while holding resources at the same levels. As a result, product development capacity has become the major limitation today for many businesses. These capacity limitations affect all aspects of product development – from deciding which projects to undertake to creating technology roadmaps to coordination and control in execution.

Capacity limitations affect all aspects of product development.

Think about the changes as they relate to portfolio selection. At the portfolio level, it is no longer possible to accept a project on its own merits. The question one must answer is: Which current or new projects will have to be sacrificed to free capacity up for the project being considered, and what is the net effect on business goals of sacrificing those projects?

In addition, lead-times for developing new technologies are affected by capacity. Often the best option might be to use an off-the-shelf technology or extend an old platform than wait for capacity to be freed up for leading edge technologies and new platforms.

Regarding pipeline planning, projects-based operations experience substantial queuing losses (contention for resources, wait times, multitasking, etc.) beyond a certain level of capacity utilization. Overload resources by even 10 percent and the entire pipeline gets clogged. However, if you start 10 percent fewer projects than what development capacity allows, you sacrifice substantial opportunities. Pipelines must be carefully loaded and projects properly sequenced to maximize the flow.

This flow falls under the arena of pipeline execution. The most common complaint of project managers during execution is that they do not get resources (people, equipment, facilities, etc.) when needed (even if promised during planning). Somehow, the required resources are always working on other projects. Resource managers, on the other hand, complain about conflicting priorities and about being forced to spread their resources thin.

Determining good priorities for resources and assigning them to the right tasks at the right time can dramatically shorten lead-times and increase project completion rates.

While little margin for error remains today in managing product development capacity, managing the process of developing new products hitherto has been impossible. The next section explains why, while later on we explain how using theory of constraints can help tackle this thorny management issue.

Obstacles to managing product development capacity

While capacity management has been successfully applied and integrated into manufacturing management, three factors make it difficult and often block practitioners from managing product development capacity: Complex linkages, high uncertainties and poor data.

Developing products involves hundreds to thousands of interrelated activities. Complex linkages exist not just within a project, but also across projects. For example, many product projects depend on platform projects, which in turn are fed by technology projects. With such complexity, balancing resources becomes virtually impossible.

Unconstraining government operations

Applying theory of constraints has improved numerous government operations in Utah, from the crime lab to the agriculture department.

Kristen Cox, executive director of the Utah Governor's Office of Management and Budget, has been called "the most prominent and articulate person" advocating for theory of constraints in government, reported *Governing* magazine. And last year she was named the state's "public official of the year," according to utahpolicy.com, a website that covers politics and government in the state.

She has schooled dozens of agency leaders in the theory of constraints management method, the website reported, and can engage people and explain it in a way that is easy for them to understand. As a result, the state crime lab processes cases 66 percent faster, the agriculture department treats nearly twice as many invasive weeds each year and a public health lab has tripled the number of tests it runs each week, cutting its backlog.

Likewise, a high degree of uncertainty exists because development tasks cannot be perfectly estimated, and delays are inevitable. These delays quickly multiply through activity dependencies and shared resources, preventing management from establishing stable plans and priorities.

The effects of these uncertainties are compounded by poor data. Not having managed their product development capacity in a systematic manner, most organizations have poor data about the process. The time and effort required to collect and refine data become additional obstacles to undertaking capacity management.

The result is a vicious cycle that organizations find it difficult to escape from. Enterprises need a pragmatic but powerful solution for tackling these obstacles.

When judging product development performance and considering a solution, it is important to keep in mind that we want to increase throughput, reduce operating expenses and reduce work-in-process.

Throughput (T) is the rate at which projects are completed. These projects could be products that are ready to be manufactured, technologies that can be licensed to other companies, experiments that generate useful information for subsequent steps in the development process, designs that can be incorporated into products or platforms off which new generation products can be developed.

A high degree of uncertainty exists because development tasks cannot be perfectly estimated.

Operating expense (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. Works-in-progress (WIP) are the projects that have entered the new product development (NPD) pipeline that have not yet been finished.

The longer a project stays in the pipeline, the less useful it becomes because the market changes and new technologies get created. WIP interrupts project flow. Cutting WIP cuts lead-times. WIP also creates confusion, causing delays and multitasking. Cutting WIP, therefore, increases effective capacity and throughput.

Ideally, WIP should be measured in time units, i.e., how many weeks or months of potential throughput it roughly represents. Stated another way, if no new projects were started, how much time would pass before the pipeline dries up.

The following sections describe the theory of constraints (TOC) solution for managing product development capacity and how it allows managers to maximize throughput and minimize works-in-progress and operating expenses.

Portfolio selection and project sequencing

Your organization's most limiting resources govern the flow of projects through your product development pipeline. No matter how many activities

TARGET CONSTRAINTS TO GET AHEAD

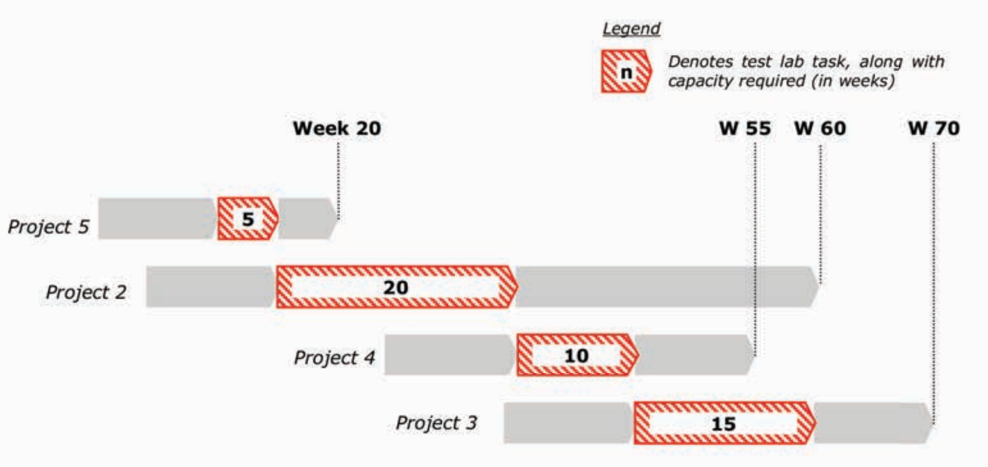
Figure 1. Constraints-based portfolio optimization allows us to choose projects that are worth a total of $105 million instead of only $75 million.

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 M/ week	3rd choice
5	$15,000,000	5 weeks	Discard	$3 M/ week	1st choice
		Portfolio throughput:	**$75,000,000**		**$105,000,000**

PROPER SCHEDULING INCREASES THROUGHPUT

Figure 2. Using theory of constraints to plan product development testing means this organization can test more options without increasing resource use.

Project	Project length prior to test	Testing capacity required	Testing schedule	Project length after test	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



need to be completed, how complex their interrelationship is or how many different resources are required, the flow of projects through your NPD pipeline is governed by its constraints, i.e., resources that have the least capacity compared to the demand placed on them.

The throughput of your pipeline cannot exceed the work that can be done at the constraints. Releasing more projects than allowed by the constraints' capacity creates unnecessary WIP. And all other resources should support throughput at the constraints. These simple facts allow us to optimize resource allocation in the portfolio selection and pipeline planning stages.

Regarding portfolio selection, 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 net present value (NPV). But let us assume that there is a finite capacity constraint imposed on the business. How should the NPV rule be modified to consider projects?

As an illustration, Figure 1 shows five projects. With no capacity constraints, we would accept all five because they all have positive NPV. Now, suppose your organization has a test lab with available capacity of 50 weeks, but these five projects need the test lab for a total of 85 weeks.

If we prioritize projects according to their individual NPV, we would accept project one, skip project two because it needs more than the remaining capacity and accept project three. Portfolio throughput would be $75 million.

But we can increase total throughput to $105 million by accepting projects five, two, four and three. These projects have the greatest combined NPV among those combinations of projects that use no more than 50 weeks of test lab capacity.

The flow of projects through your NPD pipeline is governed by its constraints.

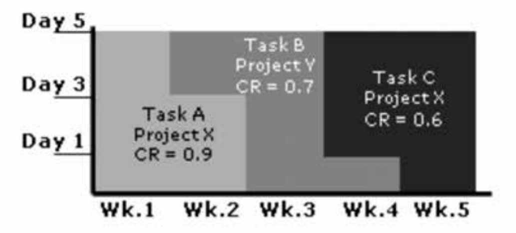
The logic leading to the correct decision is formalized by prioritizing projects on NPV per unit of constraint's capacity required for each project (constraint-indexed NPV). Using constraint-indexed NPV leads us to first accept project five, then two, four and finally three.

Now let us turn our attention to pipeline planning. In an unconstrained environment, its critical path dictates lead-time of a project. However, in

CRITICAL QUEUE CONTROL

Figure 3. Using critical ratio can help your enterprise develop just-in-time queue control for new product development.

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



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 five, two, four and three, we need to figure out the due dates of those projects. For simplicity, assume that testing lies on the critical path of each project. Figure 2 shows how the test lab is loaded and then how project due dates are established. At the planning stage, it is not necessary to create detailed project schedules. Because of high variability and strong interdependencies in product development, schedules will change daily anyway.

Releasing projects before the test lab is available will disrupt the existing flow of projects – causing loss of throughput and due-date performance.

Increasing the constraint's capacity cuts lead-times. In our example, if the company doubled the capacity of the test lab, they can finish projects four and three in almost half the time, assuming no other resources emerge as bottlenecks. Thus, an understanding of the constraints and their capacity also forms the basis of a uniquely valuable piece of information: the trade-off between lead-times and capacity.

Coordinating work and resources

Uncertainties limit us from achieving the full throughput potential of constraints. Uncertainties make local delays inevitable, and due to strong dependencies among tasks those delays tend to propagate rapidly.

When delays mount, even nonconstraints (resources that have sufficient capacity on average) start experiencing persistent peak loads that interrupt overall pipeline flow. As a result, WIP goes up and project due dates start slipping, jeopardizing throughput. Priorities also become unstable, as people are shuttled randomly between tasks, reducing their productivity by 20 to 80 percent. After all, 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 organizations want high project due-date performance and the highest possible productivity, they need to contain the effects of uncertainties. Good planning is not enough, and execution tools like time buffers and priority setting are required to assure the highest level of performance.

Time buffers are blocks of time with no scheduled work. Typically, these are placed at the end of a set of activities to absorb variability in those activities. On noncritical paths, time buffers protect integration points without increasing project length. Meanwhile, on the critical path, time buffers protect project due date but add to project lead-time. Therefore, they should be decided by those responsible for the overall performance of new product development.

With priority setting, note that 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" resource assignment can be used to contain those losses. Since it is difficult to predict the actual timing of tasks, they are scheduled when they are available to be worked on.

Critical ratio is calculated for various tasks in the queue. Critical ratio equals work remaining through to project completion divided by time to project completion. Tasks with the highest critical ratio are the ones most critical to the due dates of their respective projects and get priority. Figure 3 illustrates how to create a just-in-time schedule.

Instituting a process of ongoing improvement

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

Product development organizations are being tasked not just to make sporadic improvements but to progressively and rapidly improve their performance. This is reflected in the growing popularity of concepts such as Six Sigma, which focuses on process improvements. Since both the time and resources available to make improvements are finite, managers need to identify areas where local improvement will yield immediate and substantial gains in overall performance.

Managers need to identify areas where local improvement will yield immediate and substantial gains.

The question is, how do we identify high leverage areas?

As discussed earlier, constraints establish the upper limit on throughput, and uncertainties limit an organization from achieving that full potential by interrupting flow. Thus, improvement efforts can be directed at removing constraints and reducing uncertainties.

While constraints are easily identified, much difficulty lies in selecting areas for reducing uncertainties. However, using buffer performance history can help organizations reduce uncertainties

When time buffers are used to protect schedules, we could also keep a history of which activities use 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 interruptions to flow.

Whichever areas consistently consume the most buffer time should be targeted for process improvements, things that could include tightening technical processes, improving task estimates and deploying computer-aided engineering tools. All these potential improvements can be prioritized.

Obviously, at the beginning, your organization won't have the data to help decide what process improvements to tackle first. But improvement can start despite poor data, because when data do not tell the truth (or, as in this case, are nonexistent), intuition can be the guide.

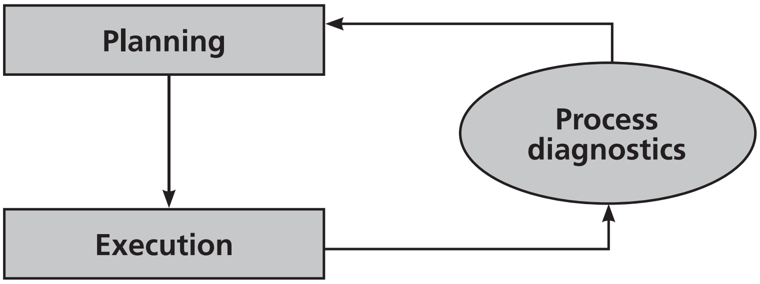
Although capacity and projects data are either poor or nonexistent in most product development environments, organizations do know which 20 percent of their resources are most overloaded. These are the areas where your enterprise has a perpetual need to hire more people or outsource work.

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. A seven-step process can help.

IT'S ALL RELATED

Figure 4. Adding process diagnostics to your plans for new product development can go a long way toward optimizing execution.



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 one through six for a few cycles.

Organizations do know which 20 percent of their resources are most overloaded.

More products, quicker

Developing more products faster is the NPD key to increased competitiveness and greater profitability. Finite NPD capacity is what stands in the way. This pragmatic solution to finite capacity planning and execution provides a number of benefits.

Adding the proper process diagnostics to your planning process boosts execution, as illustrated in Figure 4. It boosts performance through high-leverage managerial decisions, not cultural change. It allocates resources to the most profitable opportunities. It achieves higher productivity by creating central resource pools.

It determines the trade-off between project lead-time and global finite capacity. It accurately estimates how much money to spend to achieve desired throughput. It contains queuing losses while providing high levels of capacity utilization. It quotes feasible project due dates.

It sets stable priorities for all project participants – which assures high due-date performance. It focuses local improvement efforts on areas that cause the biggest disruptions to throughput.

And last but not least, it breaks the vicious cycle of poor data, poor plans and unreliable execution. ❖