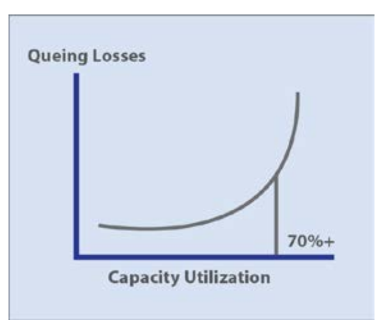
SYNCHRONIZED PRODUCT DEVELOPMENT

How to get the most from your development capacity

Businesses have to increase the rate at which they introduce new products, but without increasing engineering capacity. The implications of finite capacity, however, are missing from traditional strategies for new product development. In this paper, we present techniques for increasing the rate of product development with finite capacity.

Skilled resources are the number-one limitation for many product development projects. These capacity limitations affect all aspects of product development—from deciding which projects to undertake to coordinating and controlling execution. For example:

- **Portfolio Selection.** Which current projects will have to be sacrificed in order to free capacity for the new project being considered? What is the net effect on business goals of sacrificing those projects? Capacity limitations increasingly play a central role in determining which new products get the green light for development.
- **Pipeline Sequencing.** If resources are overloaded by even 10%, the entire pipeline gets clogged because of queuing losses (contention for resources, wait times, multitasking, etc.). On the other hand, staying 10% under capacity means sacrificing opportunities. When multiple new products are being developed with limited capacity, they need to be properly sequenced to maximize the flow.



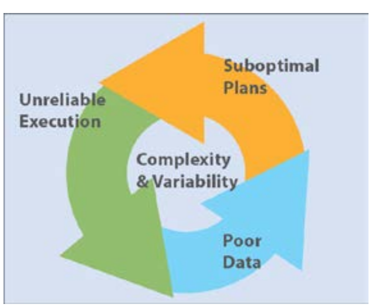
- **Execution Efficiency.** The most common complaint of project managers is that they don't get resources when needed. Somehow, required people are always working on other projects. Resources, on the other hand, complain about conflicting priorities and being forced to multitask. Again, when multiple projects are being pursued with limited capacity, assigning resources to the right tasks at the right time is the key to shorter lead times and higher throughput.

Obstacles to managing product development capacity

High uncertainty and poor data limit the use of planning and control systems in product development.

Planning and control systems have long been available for manufacturing. Two main factors, however, have traditionally blocked organizations from systematically managing product development capacity.

- **High uncertainties.** Development tasks cannot be perfectly estimated, and delays are inevitable. These delays multiply through activity dependencies and shared resources, preventing management from establishing stable plans and priorities.



- **Poor data.** Effects of uncertainties are compounded by unreliable data. Not having managed their product development capacity systematically, most organizations know little about it. Time and effort required for collecting and refining data become additional obstacles to undertaking capacity management.

The result is a vicious cycle that organizations find difficult to escape. Uncertainties and lack of data prevent the use of planning and control systems, and the lack of planning and control systems means there are no feedback loops to improve data and learn lessons.

Can these obstacles be overcome? In the following sections, we will show how product development can be managed, how real-life uncertainties can be accommodated and how data can be acquired to guide decision-making and execution—all with the objective of getting the most out of limited product development resources.

Product Development Planning: Portfolio Selection and Project Sequencing

No matter how many activities need to be completed, how complex their interrelationship is, or how many different resources are required, the flow of projects through the new product development pipeline is governed by pipeline *constraints*, i.e., resources that have the least capacity compared with the demand placed on them. As corollaries:

- Throughput of the pipeline cannot exceed throughput at the constraints.
- Releasing projects in violation of the constraint's capacity creates unnecessary work in progress (WIP).
- Capacity at non-constraints should support throughput at the constraints.

Synchronized Portfolio Selection

Without capacity limitations, the rule for selecting projects is clear and simple. Projects are accepted if they show a positive net present value (NPV). But let's assume the business has finite product development capacity, and also needs to create multiple products.[1] How should the NPV rule be modified to consider projects?

As an illustration, Table 1 shows five projects. With infinite development capacity, the decision is easy. We accept all five because they all have a positive NPV.

For most businesses, however, resources are limited. Suppose, for example, that the projects require a test lab for a total of 85 weeks, but only 50 weeks are available. In other words, the test lab is the bottleneck. How do we use the test lab's capacity to decide which projects to pursue?

We could start by prioritizing projects according to their individual NPVs. In that scenario, we would accept project 1, skip project 2 (because it needs more than the remaining test-lab capacity), and accept project 3. Portfolio throughput would be $75 million.

Can we do better? A careful look shows we can increase total throughput to $105 million by accepting projects 2, 3, 4 and 5. They have the greatest combined NPV among all projects that use the test lab for no more than 50 weeks.

Now, let's set priorities. Again, we are guided by our constraint, the test lab. In this case, we can calculate the NPV per unit of the test lab's capacity available for each project. (The unit of capacity is one week of test lab time.)

This technique is known as constraint-indexed NPV.[2] Constraint-indexed NPV gives us a rational way to determine the sequence of product development projects when capacity is limited. By calculating the constraint-indexed NPV, we decide to start project 5 first, followed by projects 2, 4 and 3, in that order.

Table 1: Illustration of constraints-based portfolio optimization

PROJECT	NET PRESENT VALUE (RISK-ADJUSTED)	CAPACITY REQUIRED AT TEST LAB	DECISION WITH SIMPLE NPD	NPV PER UNIT OF TEST LAB CAPACITY	DECISION WITH CONSTRAINT-INDEX NPV
1	$50,000,000	35 Weeks	Select	$1.43 M/Week	Discard
2	$45,000,000	20 Weeks	Discard	$2.35 M/Week	2nd Choice
3	$25,000,000	15 Weeks	Select	$1.67 M/Week	4th Choice
4	$10,000,000	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**

REALIZATION[SM]

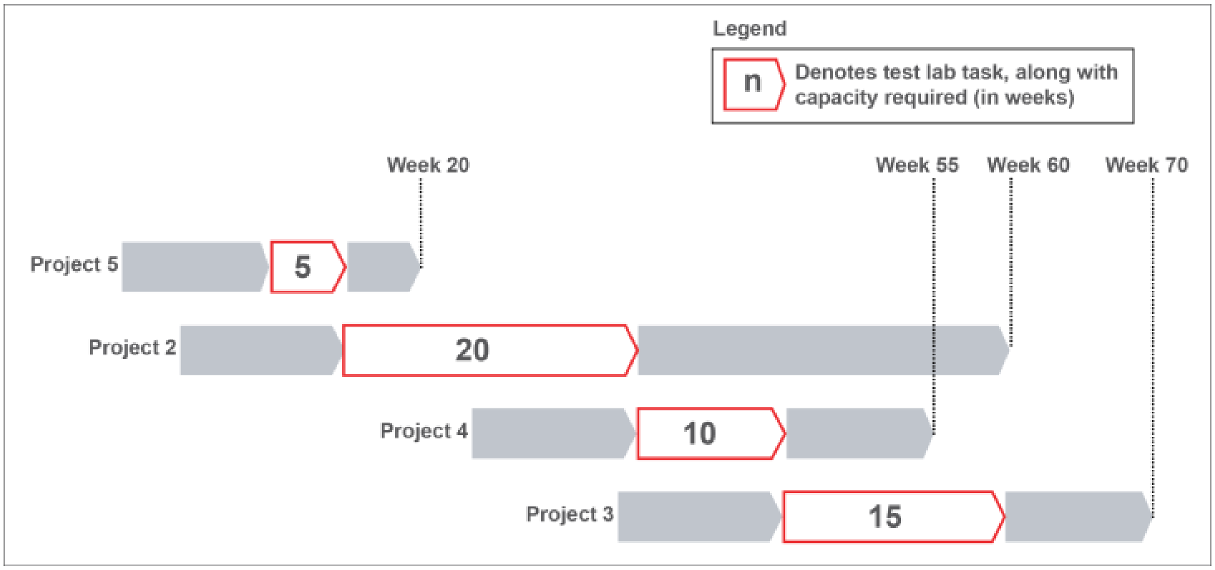
Synchronized Pipelining

When capacity is infinite, the widely used critical path method dictates a project's lead time. When capacity is limited, however, project schedules and lead times depend on when capacity is available at the constraints.

Let's continue with our previous example. Once we select projects 5, 2, 4 and 3, what should be their due dates? For simplicity, assume that testing lies on the critical path of each project. Table 2 and the accompanying figure show how the test lab is loaded, and, then, how project deadlines are established.[3]

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

PROJECT	PROJECT LENGTH PRIOR TO TEST	TESTING CAPACITY REQUIRED	TESTING SCHEDULE	PROJECT LENGTH AFTER TEST	PROJECT LEAD TIME
5	10 Weeks	5 Weeks	Week 11-15	5 Weeks	20 Weeks
2	10 Weeks	20 Weeks	Week 16-35	25 Weeks	60 Weeks
4	10 Weeks	10 Weeks	Week 36-45	10 Weeks	55 Weeks
3	10 Weeks	15 Weeks	Week 46-60	10 Weeks	70 Weeks



Why is timing important? If projects are released before the test lab is available, the flow will be disrupted. Throughput will be lost, and due dates extended.

What happens, though, if we increase the test lab's capacity? If the company doubles capacity, it can finish projects 4 and 3 in almost half the time. Thus, understanding the constraints and their capacity provides a valuable piece of information—the tradeoff between lead times and capacity, with additional implications for time-to-market.

Product Development Execution: Making It Efficient

Multitasking may be the number-one contributor to organization-wide inefficiency in new product development.

Avoiding Multitasking

Multitasking is commonly perceived as being ultra-efficient. Recent research, however, shows that the opposite is true.

In a 2009 study, for example, investigators at Stanford University found multitaskers were far less efficient than other individuals who were allowed to finish a project before starting another.[4] Our experience at Realization shows it's true for organizations, as well. Recently, getting rid of multitasking helped one of our clients vastly increase its output of new products and hit every delivery deadline. Eliminating multitasking has also helped other clients with complex projects in a variety of industries.

Multitasking is most often caused by competing priorities. When many tasks are equally urgent, people and organizations are pressured to work on all of them at once. Resources are spread thin, work in progress rises sharply and few tasks are actually completed.

Multitasking is inefficient also because it involves a "switching cost." Knowledge workers, in particular, can't be turned on and off like light bulbs. They have to ramp up to new tasks, and reacquaint themselves with old tasks when they go back to them ("Now, where was I?").

The way to end multitasking is to synchronize priorities—between workers and supervisors, within teams or departments, and across departments. Once the organization has synchronized its priorities, it must prepare tasks carefully so that resources have the needed inputs to do their jobs from start to finish. Finally, individuals must be allowed to complete what they start before taking on any other task. It may seem counterintuitive, but growing research shows sticking to a single job is far more efficient than multitasking for both individuals and organizations.

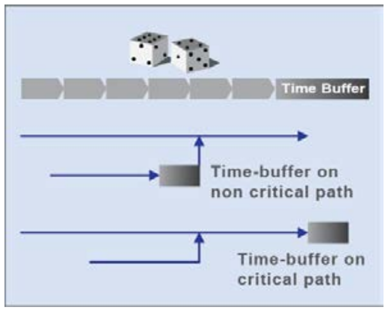
Creating Flexible Schedules

Traditionally, product development schedules are planned in great detail and have fixed deadlines for each element of the project. Once the project begins, however, reality intervenes and the meticulously created schedule quickly becomes obsolete. Even so, product development organizations typically create fixed schedules when they plan projects.

Here again, new thinking is in order. Instead of setting rigid schedules that rapidly become irrelevant, we need flexible schedules that absorb the shock of realities such as missed vendor deliveries, design delays and other factors that can throw new product development into chaos—especially when numerous products are being created at once.

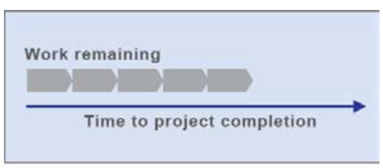
Flexible schedules are based on two main principles—time-buffers and priority-setting.

- **Time-buffers:** These are blocks of time with no scheduled work. They're typically placed at the end of a set of activities to absorb variability.



  - On non-critical paths, time-buffers protect integration points, without increasing project length.
  - On the critical path, time-buffers protect the project's due date.

- **Priority-setting:** Even with adequate average capacity, task-time variability during execution can cause peak loads. In turn, peak loads can create queuing losses in the form of delayed projects and expediting costs. "Just-in-time" resource assignment can control such losses, as follows:

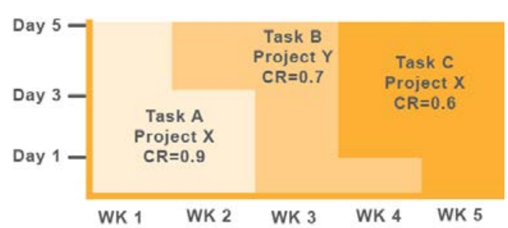


  - Since it's difficult to predict the actual timing of tasks, they are scheduled when they can be worked on.
  - "Critical Ratio" is calculated for various tasks in the queue. Critical ratio equals the *work* remaining until project completion, divided by the *time* to completion.
  - Tasks with the highest Critical Ratio are the ones most essential to the due dates of their respective projects. Thus, they get first priority.

As we observed in our discussion of multitasking, the most important point to remember about priorities is that they must be synchronized across the board for the benefit of the entire project. When priorities are allowed to be localized, they cause resources needed for the good of the whole project to be tied up in other activities. The results are delays and multitasking, which quickly translate into missed deadlines and cost overruns.

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



REALIZATION℠

Getting started despite very poor data

When information misleads, intuition can be the guide.

As discussed earlier, data about capacity and projects are either poor or non-existent in most product development organizations. However, organizations do know which 20% of their resources are most overloaded (e.g., where there is a perpetual need to hire more people or outsource work).

Such intuition can focus data collection and cleanup efforts (capacity data for likely constraints, 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 progressively make the model accurate. Sophisticated analysis should be possible within a few months.

Here is a set of useful guidelines:

1. Use intuition to pinpoint probable constraints.
2. Eliminate or reduce multitasking at constraints.
3. Rationalize due dates for projects flowing through the bottlenecks.
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

Product development capacity is finite. A variety of techniques can help you get the most from your limited capacity.

The growing problem of capacity limitations in new product development means it's time for new thinking about the way projects are selected, pipelines are loaded and execution is managed. Our experience shows that substantial improvements are possible through synchronized priorities, flexible schedules, the elimination of multitasking, focusing on bottlenecks, and calculations such as constraint-indexed NPV and Critical Ratio. These principles and techniques clarify complex issues, guide rational decisions and create the highest possible output under finite new product capacity. Organizations that use them will find themselves more productive, more competitive and more profitable.

The methods we have discussed are straightforward to use and provide 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 tradeoffs between project lead times and finite global 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, ensuring high due-date performance
9. Break the vicious cycle of poor data, poor plans and unreliable execution
10. Create synchronized priorities and avoids multitasking

Please contact Realization Technologies with comments or requests for additional information.

Phone: 408.271.5100

Email: info@realization.com

[1] Such a situation is often called a constrained capital budgeting problem (see Copeland and Weston, *Financial Theory and Corporate Policy,* 3rd edition, p. 55). For example, the constraint could be budget or capacity.

[2] Constraint-indexed NPV yields the same results as linear programming optimization. Thus, constraint-indexed NPV is not only simple, but also optimal.

[3] At the planning stage, detailed project schedules are unnecessary. Because of high variability and strong interdependencies in product development, schedules will change daily.

[4] Eyal Ophir, Clifford Nass and Anthony D. Wagner, "Cognitive Control in Media Multitaskers," Proceedings of the National Academy of Sciences of the United States of America, August 24, 2009.

[5] Now that contention for resources among concurrent projects can be resolved, it's no longer necessary to maintain artificial silos of capacity.