



Whitepaper

# Starts control

## Balancing the portfolio for speed

### **FTTM Best Practices series.**

Speed is decided at the portfolio, before any project team is formed. This paper examines how portfolio management and time to market are connected, why uncontrolled project starts produce too many projects, the wrong projects, and not enough resources, and how best-practice companies put a clock on the fuzzy front end, rank every candidate against weighted business objectives, and gate each start against real resource capacity. A follow-on to the published lateralworks briefing Portfolio prioritization & starts control, drawn from FTTM research and field deployments and cross-referenced against four decades of portfolio management literature.

---

**Prepared by**

lateralworks  
FTTM methodology

**Date**

July 2026  
Research white paper

**Online**

lateralworks.com  
FTTM Best Practices series

# Table of contents

## Starts control

---

Abstract	03
01 Anyone can start a project	04
02 Why every project ends up late	07
03 Put a clock on the fuzzy front end	11
04 Balance the portfolio: a zero-sum game	14
05 Model, rank, then gate the starts	17
06 Who owns the portfolio?	22
07 The host mindset	26
08 Three companies that balanced for speed	30
09 Putting starts control in place	34
References	37

**Core thesis.** The active project portfolio is a zero-sum game. Best-practice companies balance it deliberately: they manage the fuzzy front end on a clock, rank every candidate against weighted business objectives, and gate each start against real resource capacity. When one project is added, another is killed or deprioritized. Organizations that skip this discipline start projects whenever anyone with clout wants one, overload the same engineers who are already carrying the committed portfolio, and deliver everything late.

**Companion paper.** This white paper is the full-length follow-on to the published lateralworks briefing *Portfolio prioritization & starts control*. For the short-form version of the rank-first, gate-second discipline, see [lateralworks.com/papers/portfolio-prioritization-and-starts-control.pdf](https://lateralworks.com/papers/portfolio-prioritization-and-starts-control.pdf) [14].

## Overview

# Abstract

---

**Speed is a portfolio property before it is a project property.** An organization can build excellent teams, plan well, and execute with discipline, and still ship everything late, because the layer above the projects is overloaded. How a company manages its portfolio determines the three conditions every project inherits: whether there are too many projects for the pipeline, whether they are the right projects, and whether the projects that matter have the resources to move. One discipline governs all three, and it operates at a single point: the moment a project is allowed to start. Manage that moment and the portfolio stays balanced, cycle time compresses, and the right products reach the market at the right time. Leave it unmanaged and no amount of downstream project heroics recovers the loss. This paper is the full-length follow-on to the published lateralworks briefing Portfolio prioritization & starts control [14], expanding its two-step discipline, rank first, gate second, into an end-to-end operating method.

In most product organizations, nothing controls when a project starts. A customer asks, an executive sponsors, marketing commits, and a new project enters the pipeline. No one models what the new start does to the twenty projects already competing for the same engineers. The result is predictable and universal: every project is under-resourced, every project queues behind every other project, and every project ships late. Robert Cooper calls the condition pipeline gridlock, and his benchmarking research identifies it as the most frequently cited failure in new product development [1]. Wheelwright and Clark documented the same pathology at a scientific instruments firm that had quietly overcommitted its development capacity by a factor of three [3].

The damage begins upstream of execution, in the fuzzy front end. Because the period between idea and staffed team is unowned and unmanaged in most companies, projects drift into existence rather than start on a decision. lateralworks field research across 200+ programs finds that up to 100 percent of total development time can be lost in this unmanaged gestation period [9]. Best-practice organizations treat the front end as a formal phase with a clock on it: a small multidisciplinary team, a fixed feasibility window, and a go/no-go decision at the end.

What separates the fastest product companies is the briefing's two-step discipline, run at full depth. First they rank: every candidate project is scored against weighted strategic and economic objectives, using a managed consensus process, before any constraint is applied. Then they gate: projects load in priority order until cumulative demand crosses the resource ceiling, and the line between funded and deferred is published. Projects below the line are deferred, descoped, or killed. The portfolio becomes a zero-sum game, which is precisely what makes it fast [14]. The chapters that follow cover the failure pattern, the queueing arithmetic of overload, the front-end clock, the ranking and gating mechanics from lateralworks zero-based portfolio deployments, the governance roles that make decisions stick, the fast-time-to-market (FTTM) portfolio practices behind the method, and three case studies, Apple, LEGO, and Corning, that show what the discipline returns [8, 9].

# 01

## The problem

### **Anyone can start a project**

---

Walk into almost any product organization and ask a simple question: what has to be true before a new project can start? The honest answer is usually nothing. A senior sponsor, a high-profile customer, or a business unit with budget authority is sufficient. The development pipeline has no gate at its entrance, and the organization treats project starts as free.

They are not free. Every start spreads a fixed pool of engineers across one more claim on their time. The new project does not bring its own staff; it borrows from the projects already underway. Starting a project creates cost immediately and value only when it finishes. An organization that celebrates starts is celebrating the acquisition of cost.

## Failure pattern

# How uncontrolled starts play out

The pattern repeats across industries and company sizes. Marketing and engineering run separate selection systems: marketing commits to revenue plans that assume new products, engineering funds what it can with the people it has, and no one reconciles the two [9]. Projects enter the pipeline informally, so funding is erratic in the early stages when stability matters most. Projects never get killed, and new ones keep arriving without regard to the load on resources already committed. When executives complain they lack people, they are told to get creative and be more efficient. The pipeline expands every planning cycle while the resource pool stays flat [9].

Cooper's benchmarking work quantifies the end state. In one portfolio review he documents, sixteen active projects had a combined resource demand of 60.7 full-time equivalents against a supply of 26: the pipeline was overcommitted by a factor of 2.3, nearly every project was under-staffed, and two executive-sponsored projects alone were consuming 66 percent of the resources [1]. Wheelwright and Clark's PreQuip case is the same story at larger scale: 30 active projects, capacity for roughly a third of the work, engineers reshuffled to whichever project was in crisis, and delays cascading through the whole portfolio [3].

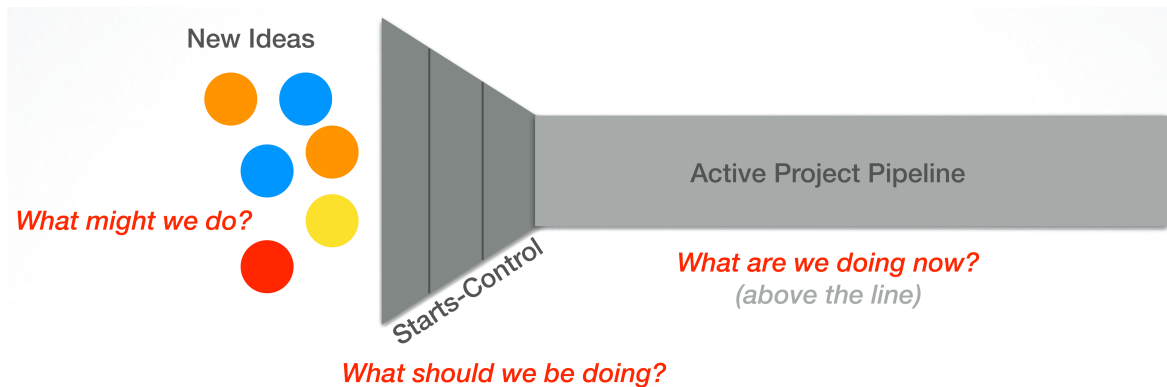


Figure 1. The starts-control question. New ideas (what might we do?) must pass through a deliberate starts-control gate before entering the active project pipeline (what are we doing now?). Source: lateralworks ZBB starts control process, client deployment materials.

lateralworks engagement diagnostics collect the causes of late projects at the start of every portfolio engagement, and the same list surfaces every time: the project started late; the resources it needed were still tied up on the last late project; no priorities exist, or they change mid-flight; nobody manages the fuzzy front end; the team is under-resourced in specific skills and over-committed in quantity; execution is unmanaged; and no single person owns the portfolio [20]. Every item on that list is upstream of the project team. The team inherits lateness before it writes a single requirement.

**The definition.** Starts control is the formal discipline that determines when, whether, and in what order new product development projects enter the active pipeline. It sits between the fuzzy front end and project execution, and it is the only mechanism that protects committed projects from the constant pull of new ideas.

The external evidence lines up with the field data. PMI's Pulse of the Profession benchmarking finds that champion organizations, those that complete 80 percent or more of projects on time and on budget, achieve a 92 percent project success rate against 32 percent for underperformers, and the differentiator is directed portfolio discipline rather than execution heroics [2]. Cooper, Edgett, and Kleinschmidt's study of portfolio practices reaches the same conclusion from the other direction: the businesses with the poorest-performing portfolios are the ones that select projects with financial models alone and end up with too many projects for the resources available [4].

# 02

## The arithmetic **Why every project ends up late**

Overloading a development organization does not slow it down proportionally. It slows it down catastrophically, because two nonlinear mechanisms compound: the context-switching tax on each individual engineer, and queue growth across the shared pipeline. Both are well quantified in the literature, and both are routinely ignored by the planning processes that add projects to the pipeline.

## Mechanism one

# The context-switching tax

Gerald Weinberg's model of context-switching loss remains the most cited quantification of what happens when engineers carry multiple projects [5]. The relationship is nonlinear: the second project does not cost ten percent of an engineer's time, it costs twenty. By five projects, three quarters of the engineer's time is consumed by switching overhead.

Simultaneous projects	Time per project	Lost to switching	Total productive time
1	100%	0%	100%
2	40%	20%	80%
3	20%	40%	60%
4	10%	60%	40%
5	5%	75%	20%

*Table 1. Weinberg's context-switching model [5]. Productive capacity collapses nonlinearly as concurrent project count rises.*

The companion lateralworks paper *The Optimal Project Load* traces this evidence in depth and establishes 1–2 projects per engineer as the research-backed maximum for complex development work [8]. The point for portfolio management is simpler: every project added above the capacity line converts productive engineering time into switching overhead across the whole organization. The new project does not just progress slowly itself; it makes every other project slower.

## Mechanism two

# Queues explode near full utilization

---

Donald Reinertsen's application of queueing mathematics to product development explains why the damage is so much larger than intuition suggests [6]. When a process with variability runs at high capacity utilization, queue time grows exponentially, not linearly. At 60 percent utilization queues are short and predictable. By 90 percent they have multiplied roughly nine-fold, and past 95 percent they spiral. Development work is highly variable, so an organization that loads its engineers to full utilization has chosen, mathematically, to make queues its dominant cycle-time component. Thomke and Reinertsen list the fallacy of high utilization as the first myth of product development: their executive surveys found the average product development manager keeps capacity utilization above 98 percent, and at those levels adding as little as 5 percent more work can double a project's duration. Most firms treat idle capacity as waste rather than as the thing that keeps the pipeline flowing [7].

Starts control is the portfolio-level answer to both mechanisms. It is the only lever that sets utilization deliberately rather than by accident. That is why the lateralworks briefing on this subject calls starts control the highest-leverage cycle-time lever available to a product organization: small overcommitment produces large schedule damage, so starting fewer projects is how more projects finish on time [14].

## The evidence

# What happens when the load comes down

---

The PreQuip transformation documented by Wheelwright and Clark is the cleanest before-and-after in the literature. The company mapped its 30 active projects, matched them against strategy and capacity, and canceled more than two-thirds of them, including senior managers' pet projects. Eleven projects survived. Between 1989 and 1991 commercial development productivity improved by a factor of three: fewer projects meant more work actually finished. Just as telling, PreQuip deliberately assigned only 75 of its 80 available engineers, holding a capacity cushion so that variability and unexpected opportunities could be absorbed without wrecking the plan [3].

Cooper reports the same result pattern from portfolio reviews: when the sixteen-project portfolio described in section 01 was force-ranked and cut to six funded projects, the five top-priority projects (excluding one executive-sponsored holdover) saw their resourcing triple, and acceleration followed [1]. The throughput paradox is real and repeatable: starting fewer projects at any given time increases the number of projects completed per year, because each one moves through the pipeline faster [1, 6].

Normal organization	Best organization
Projects start when a sponsor with clout wants one. The pipeline has no entrance gate.	Projects start when the portfolio model shows capacity. The entrance gate is explicit and enforced.
Engineers carry 3–5 concurrent projects; utilization runs at or past 100 percent.	Engineers carry 1–2 projects; utilization is held near 75–85 percent to absorb variability.
Projects never get killed. Weak projects consume resources to the end.	The portfolio refresh kills or defers weak projects and reallocates their people to winners.
Every project is a little bit late, then very late. Dates are managed by escalation.	The funded list is short enough that committed dates hold. Escalation is the exception.

*Table 2. The load discipline contrast, from lateralworks FTTM Best Practices field observations [9].*

# 03

## FTTM practice 2.1

### **Put a clock on the fuzzy front end**

---

The period between the first serious discussion of a product concept and the assignment of a staffed team is the fuzzy front end. It is called fuzzy because in most organizations it is unowned and unmanaged: it happens when it happens. Nothing about the front end requires it to be fuzzy. Best-practice organizations run it as a formal development phase with target dates, a small multidisciplinary team, and a decision at the end. The difference in elapsed time is enormous.

## The unmanaged default

# Ideas drift until they become late projects

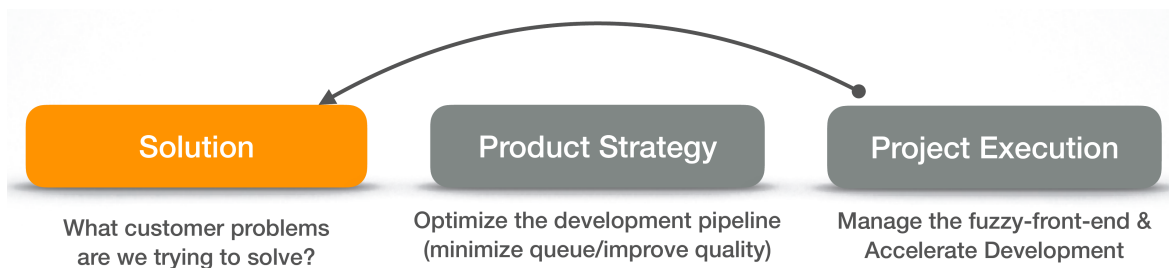
In the normal organization, a new idea is championed by one or two marketing people, or by a skunkworks of engineers quietly drawing time from funded projects to run feasibility studies. The idea lacks the resources for real technical and business analysis, so it gets cursory attention while management focuses on the projects already in trouble downstream. The gestation period can run as long as the entire design and development phase that follows, because nobody is on the clock [9]. Khurana and Rosenthal's study of front ends reached the same conclusion: companies that lack explicit front-end ownership and process see their projects enter development with unstable definitions and unrealistic expectations [10].

The worst damage is inherited. Goals set in the unmanaged front end, performance targets, schedule expectations, cost assumptions, were set without enough information, yet they attach to the formal team and never seem to change [9]. Smith and Reinertsen made the economic version of the argument decades ago: the front end is where cycle time is cheapest to buy, because a week saved before the team is staffed costs almost nothing compared to a week recovered during development [11].

## The best practice

# A fixed window and a real team

Best product organizations assign a small multidisciplinary team of engineering and marketing people to every serious concept, with a narrow fixed window, typically under two months, to determine technical and business feasibility and reach a go/no-go decision [9]. The feasibility work gets immediate attention because these assessments feed the portfolio process: they are one of the two major sources of ranked candidate projects. Senior management treats gestation time as development time, because when the front end is included in total time to market, cutting it produces some of the largest acceleration available anywhere in the process [9, 11].



Goal: Balance portfolio, efficient use of resources... 20-80

Figure 2. The FTTM portfolio scope. Product strategy optimizes the development pipeline between solution definition and project execution; managing the fuzzy front end is an explicit part of the scope. Source: lateralworks FTTM portfolio framework.

Putting a clock on the front end also changes the quality of what enters the portfolio funnel. A concept that has passed a time-boxed feasibility study arrives at the ranking step with a validated problem statement, a rough business case, and an effort estimate. A concept that drifted in through a sponsor arrives with enthusiasm. Ranking works on the first kind and is gamed by the second.

**Practice 2.1, fuzzy-front-end is managed.** Treat the front end as a formal phase: small cross-functional team, fixed feasibility window, go/no-go at the end. Up to 100 percent of total development time can be lost in an unmanaged front end [9].

# 04

## The discipline

### **Balance the portfolio: a zero-sum game**

A portfolio is balanced when the projects in flight match the resources available to execute them, in an order that serves the business strategy. Balance is not a report; it is a repeated decision. Because the resource pool is fixed on any planning horizon, the decision is zero-sum by definition: funding a new project means taking capacity from an existing one. Best-practice companies make that trade explicitly. Normal companies make it implicitly, by letting every project starve a little.

## The flow

# Strategic filter, economic filter, then the pipeline

The lateralworks decision-making flow runs every candidate through two filters before it can touch the active pipeline. The strategic filter asks: where should we focus? Projects that do not serve the business strategy are eliminated regardless of how attractive they look individually. The economic filter asks: what should we start? Projects that survive strategy are tested against economic criteria, revenue, return, growth. Only then does the pipeline question arise: what order should we do them in, and what do our constraints allow us to start now? A closing loop, success and failure analysis, feeds what the organization learned back into the next cycle [9, 20].

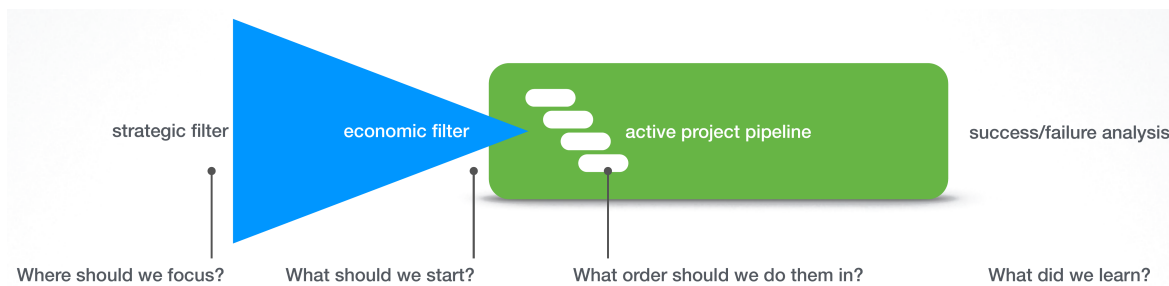


Figure 3. The decision-making flow. Candidates pass a strategic filter and an economic filter before entering the active project pipeline; failure analysis closes the loop. Source: lateralworks FTTM portfolio framework.

This sequencing, rank first, gate second, matters more than it looks. If stakeholders weight criteria while staring at the capacity constraint, the math accommodates the answer they already wanted. Building the unconstrained ranking first preserves what the strategic and economic evidence actually says, then forces the deferral conversation to happen against a transparent reference [14]. Wheelwright and Clark’s aggregate project plan embodies the same principle: classify and map the projects first, then match the set against capacity, then cut [3].

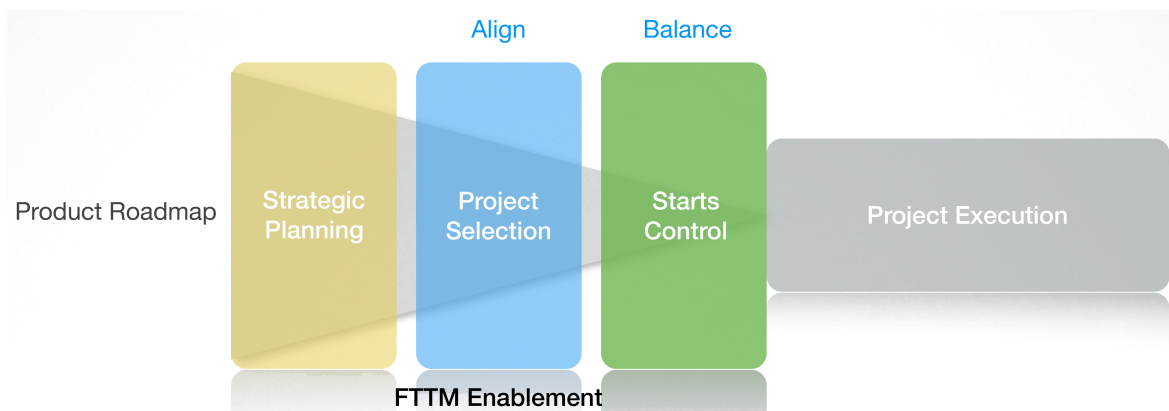


Figure 4. Align, then balance. Strategic planning and project selection align the portfolio with the product roadmap; starts control balances it against capacity before execution. FTTM enablement spans the chain. Source: lateralworks ZBB starts control process.

# The zero-sum rule

## Adding means killing

The prioritization that emerges from the filters is a zero-sum ranking: when one project rises in importance, others fall. This is deliberate. It forces real prioritization instead of the comfortable fiction that everything is priority one [9]. The same rule governs entry to the pipeline. New ideas are pushed through the model to see how they rank against the approved portfolio, and the refresh funds, reduces funding, or kills projects that no longer score. A new project that ranks above an existing one triggers a trade-off deliberation, not an automatic add: approve the newcomer and something else gets paused, descope, or killed to free its resources [9, 20].

Cooper’s portfolio reviews institutionalize the identical rule with a drawn line: projects are force-ranked one to N, resources are added up down the list, and where the resource limit is reached, the line is drawn. Projects below the line are killed or put on hold and their resources move up the list [1]. The lateralworks deployment experience adds one hard-won refinement: publish the line. A hidden line lets the active list quietly refill between quarterly reviews, because projects below the line keep consuming capacity informally [14].

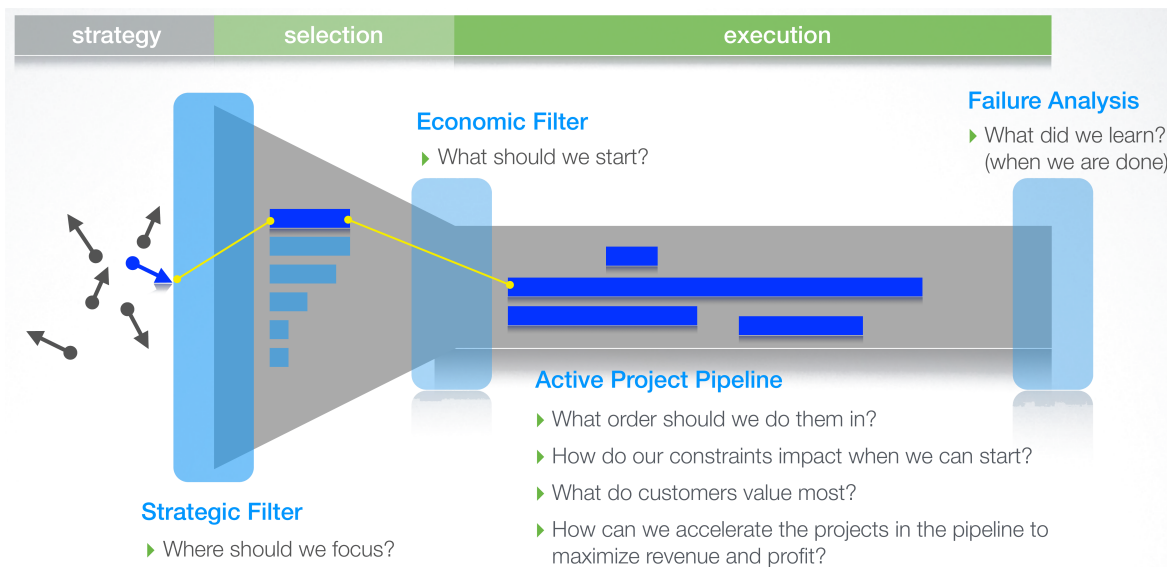


Figure 5. Strategy, selection, execution. The strategic and economic filters feed a constrained active pipeline; the pipeline questions are about order, constraints, customer value, and acceleration. Source: lateralworks ZBB starts control process.

Note what the zero-sum rule does to organizational behavior. When adding a project visibly costs another project, sponsors self-screen: weak ideas stop being proposed because their sponsors can see what the idea would have to beat. The rule converts portfolio management from an allocation argument into an investment comparison, which is what it always should have been [1, 4].

# 05

## The mechanics **Model, rank, then gate the starts**

The mechanics of starts control are a six-step zero-based budgeting loop, refined across lateralworks portfolio deployments in semiconductors, solar, and systems companies. Zero-based means exactly that: no project, existing or proposed, has a standing claim on resources. Every refresh cycle, the whole portfolio re-earns its funding against the business objectives. The loop produces an unconstrained ranking first, then applies the resource ceiling, and the result is an approved project list that the organization can actually execute.

## Rank without constraints

# Model the portfolio against objectives

The loop begins by gathering the complete project list, hidden skunkworks included, and defining the business objectives the portfolio must serve. Each stakeholder then weights the objectives through pairwise comparison, the mechanism at the heart of Saaty's Analytic Hierarchy Process [12]. Pairwise judgment matters because direct point assignment invites gaming: comparing objectives two at a time forces each stakeholder to reveal a consistent preference structure. The individual weightings are then consolidated in a stakeholder workshop, where the differences themselves are the agenda. Reviewing why sales weighted growth twice as heavily as engineering weighted risk is where the organization discovers it has never actually agreed on strategy [20].

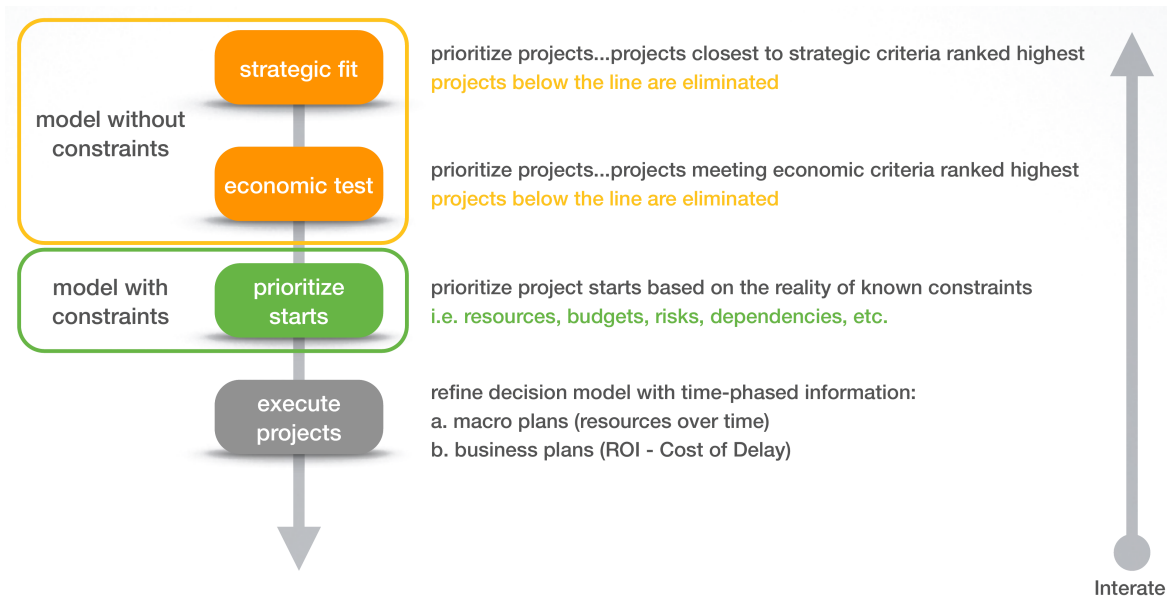


Figure 6. The modeling flow. The portfolio is modeled without constraints first (strategic fit, then economic test), and only then are starts prioritized against the reality of known constraints. Source: lateralworks FTTM portfolio framework.

Projects are rated against each weighted objective, and the model produces a zero-sum priority ranking. Typical criteria in lateralworks deployments: revenue impact over the planning horizon, strategic fit against market and technology strategy, market-maker customer engagement, value proposition strength, and execution risk [20]. The criteria are deliberately few, four to six, and each carries a defined rating scale so that scores mean the same thing across raters. Projects closest to the weighted criteria rank highest; projects below the line at each filter are eliminated before constraints ever enter the conversation.

# Apply the constraints

## Load priorities against real capacity

Only after the unconstrained ranking exists does the model take on constraints: the R&D budget per project in the simplest deployments, per-project headcount estimates by skill group in the more precise ones. Projects load in priority order until cumulative demand crosses available capacity. The output is not advice; it is an approved project list, the set of projects that get funded this cycle [20].

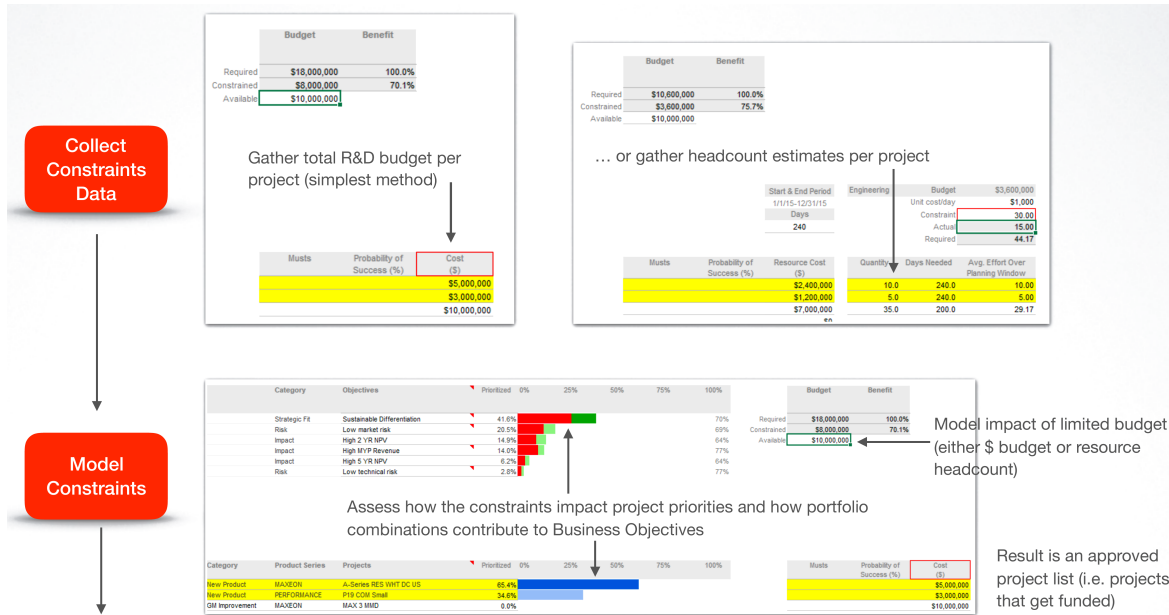


Figure 7. Constraint modeling in a live deployment. Budget or headcount data per project is loaded against the prioritized list to determine which portfolio combinations best serve the business objectives. Source: lateralworks ZBB starts control process, client deployment (identifying details removed).

The cost-benefit view completes the loop. Plotting cumulative benefit against cumulative investment down the ranked list exposes the portfolio sweet spot: the point where benefit is maximized for the lowest investment. lateralworks field data consistently shows the 20-80 pattern, roughly 20 percent of candidate projects generating about 80 percent of the modeled benefit, which is why the process is designed to locate that high-value fraction and give it maximum resources [9, 20].

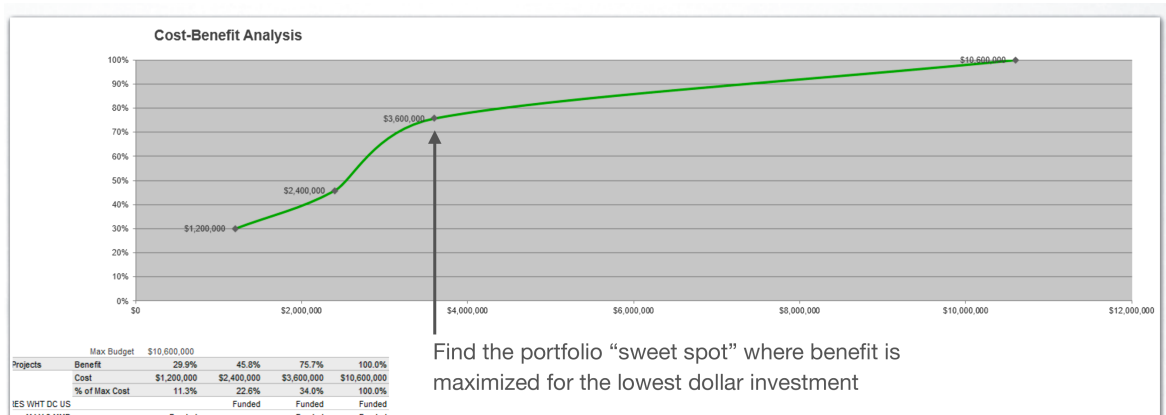


Figure 8. Cost-benefit analysis across the ranked portfolio. Cumulative benefit is plotted against cumulative cost in rank order; the curve flattens past the sweet spot, where additional projects add investment faster than they add benefit. Source: lateralworks ZBB starts control process.

## The ZBB line

### What the gate looks like in practice

The figure below is the artifact that makes starts control real in a working organization: a zero-based budget sheet from a division-level deployment. The upper table is demand, prioritized programs and their monthly engineering requirements. The lower table is what remains of a 23-engineer supply as each priority loads in order. The ZBB line is drawn where remaining capacity crosses zero. Programs above the line are staffed; the red cells below the line show demand the organization cannot meet without either adding capacity or removing a higher-priority program. The sheet's power is that it makes the overcommitment undeniable, monthly, by name [20].

CTD ZBB (Engineers)

Resource Requirement:

Priority	Resource Needs	Sum of Jan'19	Sum of Feb'19	Sum of Mar'19	Sum of Apr'19	Sum of May'19	Sum of Jun'19	Sum of Jul'19	Sum of Aug'19	Sum of Sep'19	Sum of Oct'19	Sum of Nov'19	Sum of Dec'19
1	NGT 1.0 Deployment	11.6	12.6	10.9	9.3	7.8	7.4	5.2	4.8	2.5	2.1	2.1	2
2	NGT 1.1 Deployment	1.2	1.3	1.6	2.5	4	4.3	4.8	4.8	4.4	3.9	2.4	2
3	Cost Reduction Fab4 - Athena	3.7	3.6	5.5	6.7	7	6.6	7.1	7.1	7.1	7.1	7.1	6
4	NGT 2.0 Development - BE	1.7	1.9	1.9	1.9	1.9	2.3	2.3	2.3	2.3	2.3	2.3	3
4	NGT 2.0 Development - FE	3.5	2.5	3.5	3.3	3.3	3.3	3.8	3.8	3.8	3.8	3.8	3
5	Cost Reduction Fab3 - Aurora	0.9	1.4	1.6	1.5	0.2							
6	Cost Reduction Fab3 - NGT 1				1.5	1.6	1.9	1.7	1.7	1.7	2.9	3	
	Grand Total	22.6	23.3	25	26.7	25.8	25.8	24.9	24.5	21.8	22.1	20.7	20

Remaining Resource:

Priority	Resources Remaining	Sum of Jan'19	Sum of Feb'19	Sum of Mar'19	Sum of Apr'19	Sum of May'19	Sum of Jun'19	Sum of Jul'19	Sum of Aug'19	Sum of Sep'19	Sum of Oct'19	Sum of Nov'19	Sum of Dec'19
1	NGT 1.0 Deployment	11.4	10.4	12.1	13.7	15.2	15.6	17.8	18.2	20.5	20.9	20.9	20
2	NGT 1.1 Deployment	10.2	9.1	10.5	11.2	11.2	11.3	13	13.4	16.1	17	18.5	18
3	Cost Reduction Fab4 - Athena	6.5	5.5	5	4.5	4.2	4.7	5.9	6.3	9	9.9	11.4	11
4	NGT 2.0 Development - BE	4.8	3.6	3.1	2.6	2.3	2.4	3.6	4	6.7	7.6	9.1	8
4	NGT 2.0 Development - FE	1.3	1.1	-0.4	-0.7	-1	-0.9	-0.2	0.2	2.9	3.8	5.3	5
5	Cost Reduction Fab3 - Aurora	0.4	-0.3	-2	-2.2	-1.2	-0.9	-0.2	0.2	2.9	3.8	5.3	5
6	Cost Reduction Fab3 - NGT 1	0.4	-0.3	-2	-3.7	-2.8	-2.8	-1.9	-1.5	1.2	0.9	2.3	2

Figure 9. A live ZBB sheet from a division deployment (client identity removed). Prioritized resource demand loads against a fixed 23-engineer supply; the ZBB line marks where capacity runs out. Source: lateralworks client engagement archive.

Cooper recommends precisely this instrument, a simple people-against-projects spreadsheet, as the antidote to hollow go decisions: if the people are not committed at the gate, the gate approved nothing [1]. lateralworks deployments refresh the model monthly to quarterly; every refresh re-scores changed projects, ranks new candidates, and re-draws the line [20].

# 06

## Governance

### **Who owns the portfolio?**

---

The most diagnostic governance failure in portfolio management is a portfolio nobody owns. When ownership is plural, every stakeholder protects their own projects, and the equilibrium is a bloated active list with no one accountable for cycle time. The model can be flawless; without named decision roles the rankings produce no decisions, and the discipline collapses back into committee within two quarters [14].

## The diagnostic

# The questions that expose an unowned portfolio

lateralworks portfolio engagements open with a structured diagnostic, and the questions are worth listing because most organizations cannot answer them: Who owns and manages the fuzzy front end? What are your economic metrics? How do you determine when projects start? How do you align the portfolio with resource constraints? How do you determine the impact of new projects on the projects already in flight? What is the average cycle time of the fuzzy front end, and of development? An organization that answers with silence has located its portfolio problem [20].

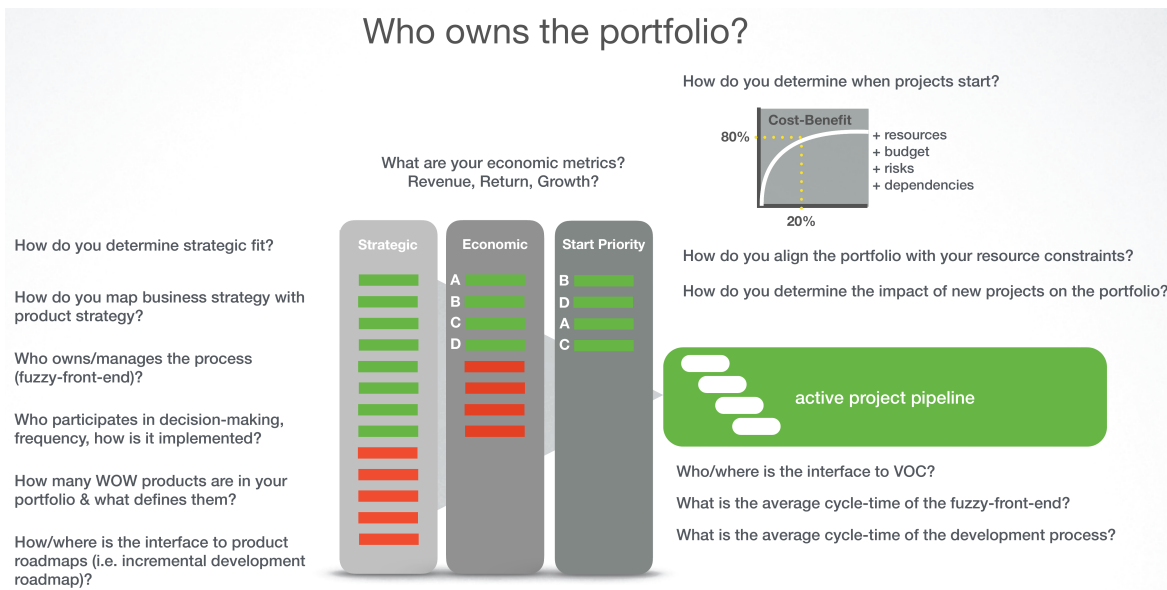


Figure 10. The portfolio ownership diagnostic. Strategic, economic, and start-priority rankings feed the active pipeline; the questions around the edges are the ones most organizations cannot answer. Source: lateralworks FTTM portfolio framework.

# Decision roles

## One decider, named in writing

lateralworks adapts Rogers and Blenko’s decision-role research [13] into the IRADP structure: who provides Input, who Recommends, who must Agree, who Decides, and who Performs. The design choice that matters is the singular Decider: one person, usually the general manager or product line leader who owns the P&L, resolves impasses and is accountable for the calendar outcome. Functional VPs and engineering leads sit in the Agree role with veto power on feasibility, not on priority. That split keeps the conversation honest: priority is a strategic call, feasibility is an engineering call, and confusing the two is how portfolios drift [13, 14].

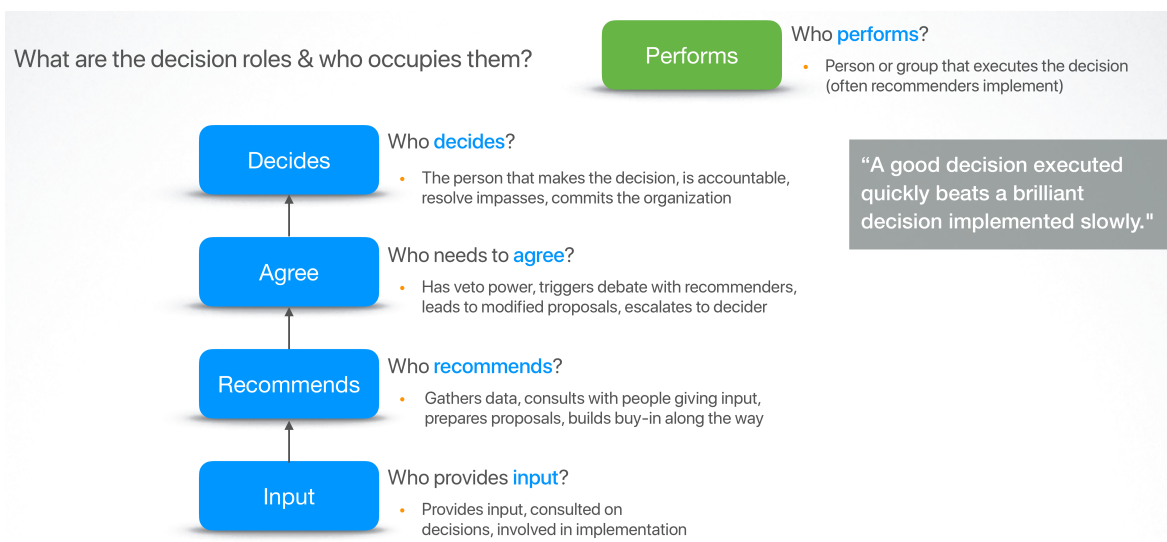


Figure 11. Decision roles for the portfolio. Input, Recommend, Agree, Decide, Perform, with a single accountable Decider. A good decision executed quickly beats a brilliant decision implemented slowly. Adapted from Rogers and Blenko [13]; source: lateralworks FTTM portfolio framework.

In deployed form the roles become a starts control board: a small standing body, chaired by the portfolio owner, that is the single organizational authority over project starts. Only board-approved projects receive staffing and budget. The board meets on a fixed monthly cadence with a published intake process, scores new proposals against the standing AHP model, and renders one of four verdicts: approved, conditionally approved pending a funded resource plan, deferred to the ranked backlog, or rejected with rationale [20]. Cooper’s prescription for gates with teeth is the same institution seen from the process side: a gate is an investment decision where resources are committed, not a milestone review [1].

**The resource gate.** Before any approval, the resource state is certified green (capacity confirmed without touching committed projects), yellow (gap exists with a funded acquisition plan), or red (unfunded gap). Red cannot be approved. A go decision without committed resources is a hollow go, and hollow gos are how pipelines refill [1, 20].

The zero-sum rule

**Balance is a decision, not a hope**

---

**When one project  
is added, another  
must be killed or  
deprioritized.**

FTTM New Product Development Best Practices  
lateralworks research, practice 2.2

# 07

## The FTTM connection

### **The host mindset**

---

Starts control is not a project practice. It belongs to the host: the organization outside the project team that provisions teams with resources, support, and guidance. In the FTTM Best Practices framework, the host either empowers teams by providing what they need when they need it, or interrupts them by filling the pipeline with more projects than the resources can carry. Portfolio balance is therefore a mindset before it is a mechanism: fast organizations believe that finishing creates value and starting creates cost, and they staff, fund, and govern accordingly [9].

# The framework

## The portfolio practices

The FTTM Best Practices framework classifies practices two ways: by who owns them (mindset, host, team) and by what they govern (portfolio, environment, execution). Reading down the portfolio column produces the complete prescription for portfolio management, and every mechanism in this paper implements one of its practices [9].

Best Practices	Portfolio	Environment		Execution	
	Prioritized Aligned Managed	Development Life-Cycle Organization Structure	Right-Product	Right-Team	Right-Time
<b>Mindset</b>	Grasp enormous financial impact late products have on profitability and growth Create fast culture through close-in oversight of product roadmap Systematically dismantle slow structures Accelerate the "threshold of pain," reward early warning behavior Fast decision-making system Cost-of-delay is understood at individual contributor level; used to drive urgency				
<b>Host</b>	The fuzzy front end is managed	Fast development framework		Cross-Functional Core Teams (CFCT); heavyweight / dedicated	Projects get killed that fail to meet objectives
	Projects are prioritized based on business objectives	Provision teams		Project-based performance measurement	
	Projects continuously aligned with resources and skills available			Freedom-level 1 empowerment	
	New products are tracked to break-even				
<b>Team</b>	Co-develop with tier 1 customers; continuously refine requirements	Schedule as driver versus reporter	Get customer requirements; translate to product definition; reconfirm through development life cycle	CFCT roles not functions; lateralized	Macro-micro planning; micro near, macro far...short interval scheduling
			Know what they value, when they want it, how much they will pay for it	Get the best people; mix of inside and external SMEs; the virtual team	Team planning, simulation, continuous scrubbing, breakdown, ownership
			Understand customer wants vs. hows; prioritize the drivers that maximize customer satisfaction		Market/Customer driven vs. resource constrained
			Start design while refining requirements; never freeze specs		Know the gap; use to drive before the fact urgency
			Fail fast; integrate / proto early, frequent & many do-it-try-it-fix-it learning cycles		Refresh Planning; continuous schedule pull-in

Figure 12. The FTTM best practice framework. Rows are mindset, host, and team; columns are portfolio, environment, and execution. The portfolio column carries the practices that starts control implements. Source: lateralworks FTTM Best Practices, revision 2018.002.

The mindset row applies across the whole column: fast organizations grasp the enormous financial impact late products have on profitability and growth, and they push cost-of-delay understanding down to the individual contributor level, where it drives urgency [9]. On that foundation sit five specific portfolio practices.

Portfolio practice	What it installs
<b>2.1 Fuzzy-front-end is managed</b>	A small multidisciplinary team and a fixed feasibility window on every concept, with a go/no-go decision at the end (section 03).
<b>2.2 Projects are prioritized based on business objectives</b>	The formal zero-sum ranking against weighted objectives, owned by a product planning function that belongs to neither engineering nor marketing (section 05).
<b>2.3 Projects continuously aligned with resources and skills available</b>	The constraint model and the ZBB line, refreshed monthly to quarterly with all stakeholders in the room (section 05).
<b>2.4 New products are tracked to break-even</b>	Teams see products through to break-even, not just to manufacturing handoff, so early design decisions carry visible economic consequences.
<b>2.5 Kill projects that fail to meet objectives, early</b>	Quarterly re-scoring against the objectives; projects that no longer rank are challenged and killed on data, freeing resources for projects that contribute.
<b>3.1 Co-develop with tier 1 customers</b>	The team-owned portfolio practice: requirements refined continuously with market-maker customers, feeding the ranking real commitment instead of forecasts.

*Table 3. The portfolio column of the FTTM Best Practices framework [9].*

## Practices 2.4 and 2.5

# Track to break-even, kill early

---

Practices 2.4 and 2.5 are the back end of starts control, and they are the two most often skipped. Tracking new products to break-even closes the economic loop that the business case opened: development teams that see a product through to profitability make different decisions early in the cycle than teams measured to the manufacturing handoff, because yield, cost, and market success stop being someone else's problem [9]. Killing early is the enforcement arm of the ranking. In normal organizations projects rarely die; when they do, they reappear under new names, and projects sponsored by senior management survive on life support long after their economic window has closed. Best organizations re-score the portfolio at least quarterly, challenge every project that falls below the bar, and treat the kill as a data-based decision rather than an emotional one. The freed resources move to the projects that still rank, which is where the throughput gain actually comes from [9].

**The throughput paradox.** Starting fewer projects at any given time increases the number of projects completed per year, because each one moves faster through the pipeline. Speed at the portfolio level is bought by restraint at the entrance [1, 6, 9].

The mindset distinction shows up most clearly in what each kind of organization considers normal. Normal organizations consider it normal that new projects appear mid-cycle, that priorities shift with the loudest voice, that everything is late, and that raising the resource question is career-limiting. Best organizations consider it normal that a new idea has to beat a funded project to enter the pipeline, that the funded list is public, and that killing a fading project is a routine portfolio event rather than an admission of failure [9]. The behavioral economics of the second culture are self-reinforcing: engineers who trust that the portfolio is protected stop hoarding, sandbagging, and quietly working pet projects, because the official pipeline is credible.

# 08

## Case studies

### **Three companies that balanced for speed**

The discipline is not theoretical. Three well-documented company histories, a consumer electronics turnaround, a toy company rescued from near-bankruptcy, and a materials science company that institutionalized reinvention, show why companies adopt portfolio balance, how they implement it, and what it returns.

## Case study one

# Apple: the 2x2 that saved the company

---

**Why.** When Steve Jobs returned in 1997, Apple was weeks from insolvency and producing over a dozen Macintosh variants plus printers, peripherals, the Newton, and the Pippin console. Overlapping teams competed for the same engineers, and no product received the concentration it needed to be excellent [15].

**How.** After weeks of product reviews, Jobs drew a two-by-two grid on a whiteboard: Consumer and Pro across the top, Desktop and Portable down the side. Apple would build four great products, one per quadrant, and cancel everything else, roughly 70 percent of the portfolio. He made the discipline permanent: at his annual top-100 retreat he would collect the ten most important priorities and strike seven, insisting Apple could only do three things well. His stated principle is the starts control mindset in one sentence: deciding what not to do is as important as deciding what to do [15, 16].

**Results.** Apple's best engineers, previously scattered across a dozen mediocre products, were concentrated on four. Development cadence compressed from roughly eighteen months to nine for major revisions, and the iMac, the first product of the focused portfolio, shipped within a year and redefined the company [15]. The mechanism is exactly the one sections 02 and 04 describe: a fixed engineering pool divided across fewer products means less switching, shorter queues, and faster cycles for everything that remains [8].

## Case study two

# LEGO: pruning the portfolio back to profit

---

**Why.** By 2003 LEGO had innovated itself to the edge of bankruptcy. Sales fell roughly 30 percent that year and another 10 percent in 2004 as the company posted the largest loss in its history. The diagnosis, documented in Robertson's *Brick by Brick*, was profitless proliferation: years of unchecked new products and ventures, theme parks, media, software, had produced a wealth of launches and almost no profitable ones, and the company could not even say which products made money [17].

**How.** Incoming CEO Jorgen Vig Knudstorp treated the crisis as a portfolio problem. LEGO cut the unique component count roughly in half, sold the Legoland parks, shed adjacent ventures, and installed financial discipline over every innovation stream, with clear criteria for what could enter development and executive governance over the innovation portfolio as a whole [17]. Growth was deliberately constrained until the core was profitable again.

**Results.** The turnaround became one of the most studied in modern business: profits multiplied through the late 2000s even through the financial crisis, and by 2015 Brand Finance ranked LEGO the most powerful brand in the world [17]. The lesson is the zero-sum rule at company scale: LEGO did not get faster and more profitable by starting more innovation, but by killing most of it and concentrating on the fraction that ranked [17].

## Case study three

# Corning: institutionalized portfolio balance

**Why.** Corning’s business model depends on repeatedly betting R&D on a small number of capital-intensive technology waves, optical fiber, LCD glass, Gorilla Glass, where a mistimed or overloaded portfolio can sink a decade. After the telecom crash nearly destroyed the company in 2001, Corning rebuilt its growth engine around explicit portfolio governance [18].

**How.** The Growth and Strategy Council, the CEO, COO, and CTO sitting as a standing portfolio board, reviews the innovation portfolio in disciplined monthly meetings, with prepared presentations from general managers, program managers, and technology leads, and written feedback after each session. Every September, the council re-ranks the full portfolio and finalizes budgets using zero-based budgeting: no program carries a standing claim, and resources are re-earned against the ranked opportunity set each year [18]. This is the starts control board of section 06 operating at the top of the enterprise, with the ZBB refresh of section 05 as its annual instrument.

**Results.** The council-governed portfolio let Corning keep funding long-horizon research while concentrating development capacity on the waves it chose, and the model produced repeated reinvention, including the rapid commercialization of Gorilla Glass into a category-defining product, at a company whose R&D runs near 10 percent of revenue [18]. Few organizations demonstrate as clearly that portfolio balance is a permanent governance function, not a one-time cleanup [18].

The crisis that forced the discipline	The starts control response
Apple 1997: a dozen-plus overlapping product lines, engineers scattered, no product excellent.	Four products on a 2x2; ~70 percent of the portfolio canceled; cadence halved; the iMac inside a year [15, 16].
LEGO 2003: unchecked innovation, unknown product profitability, record loss.	Component count halved, ventures sold, innovation gated and governed; most powerful brand by 2015 [17].
Corning 2001: near-death after betting the company on one overloaded wave.	Standing CEO-level portfolio council; monthly reviews; annual zero-based re-ranking of every program [18].

Table 4. Three portfolio turnarounds compared.

# 09

## Implementation

# Putting starts control in place

---

Starts control does not require a transformation program. lateralworks deployments stand up a working starts control system inside 90 days, because the components are few: a complete project list, a weighted objective model, a constraint sheet, a named decider, and a cadence. What requires commitment is the first refresh, the one where the organization draws the line and defends it.

# The operating system

## Starts control in the business flow

In steady state, starts control sits as a standing function between strategy and execution. Business strategy defines prioritized market segments; product strategy turns them into roadmaps and near-term prioritized products; starts control decides when projects actually enter execution, based on opportunity, lead customer, business case, required resources, a macro plan with its gap to target, cost, and risk. Performance management closes the loop with a portfolio dashboard that shows where help is needed and, when performance fails, whether to kill a project and recover its capacity [20].

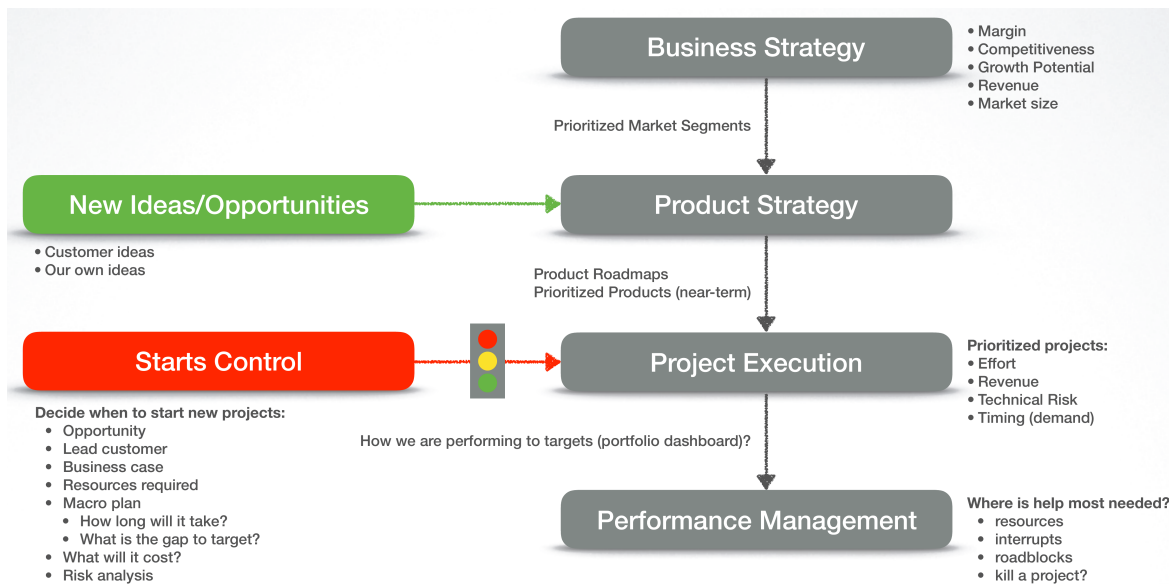


Figure 13. The portfolio operating system. Starts control gates the flow from product strategy into project execution; performance management feeds results back. Source: lateralworks ZBB starts control process.

## First 90 days

# Foundation, calibration, operation

---

The deployment sequence is deliberately unglamorous. Days 1–30, foundation: name the portfolio owner and the board roster in writing, catalog every in-flight project including the unofficial ones, define the AHP criteria and weights, and publish the intake process. Days 31–60, calibration: score the existing portfolio, build the resource allocation map, run a dry-run board meeting, and train stakeholders. Days 61–90, operation: execute the first full monthly cycle, publish the ranking and the funded line organization-wide, and enforce the rule that no budget flows without board approval [20].

Define the emergency path on day one, not mid-crisis. Time-critical opportunities get a compressed review, days instead of weeks, but the rigor holds: full scoring and a certified resource plan are still required, because the emergency lane is where pipelines refill when it is left ungoverned [14, 20]. Equally, define the triggers that force an out-of-cycle portfolio review, loss of a key customer, a capacity change, a schedule breach beyond a stated threshold, before any of them happen, while judgment is still neutral [14].

Measure the system by outcomes, not activity. Deployments typically target a 30–40 percent reduction in concurrent active projects against the pre-deployment baseline, effective engineering utilization held near 75–85 percent rather than past 100, a 20 percent reduction in average cycle time, and a measurable rise in the share of projects delivered on their original committed schedule [20]. Every one of those metrics is the queueing arithmetic of section 02 running in reverse: less WIP, shorter queues, faster finishes [6, 7].

Two failure modes deserve honest treatment, because starts control can be implemented badly. The first is bureaucratic drift: a board that reviews without deciding turns the gate into a delay of its own. The antidotes are the fixed two-hour board day, decisions published within 48 hours, and the singular Decider of section 06. The second is innovation starvation: a purely rank-ordered list favors near-term, well-quantified projects, and breakthrough work loses every head-to-head comparison against a safe line extension. Cooper's answer, strategic buckets, reserves capacity by project type so that bold projects compete only against each other, and it composes cleanly with the ranking and gating described here: buckets set the mix, the model ranks within each bucket, and the line is drawn per bucket [1].

**Start less. Finish more. Deliver better.** Controlling when projects start is how an organization guarantees that every project it does start gets the resources, focus, and attention it needs to finish fast [20].

The portfolio prioritization and constraint-loading workflow described in this paper, the AHP model, the ranked list, and the capacity line in one artifact, is a standard function of the lateralworks fastProjectAI suite, and the methodology is taught in the FTTM Academy portfolio management modules [14].

## Sources

# References

---

- [1] Cooper, R.G. “Unlocking Pipeline Gridlock: Effective Portfolio Management Is the Key.” ISBM, Penn State / InnovationManagement, 2021.  
<https://isbm.org/wp-content/uploads/2021/11/Cooper-2021-Unlocking-pipeline-gridlock.pdf>
- [2] Project Management Institute. *Success in Disruptive Times. Pulse of the Profession 2018*. PMI, 2018.  
<https://www.pmi.org/learning/thought-leadership/pulse/pulse-of-the-profession-2018>
- [3] Wheelwright, S.C., and Clark, K.B. “Creating Project Plans to Focus Product Development.” *Harvard Business Review*, March–April 1992, pp. 70–82.
- [4] Cooper, R.G., Edgett, S.J., and Kleinschmidt, E.J. “New Product Portfolio Management: Practices and Performance.” *Journal of Product Innovation Management*, 16(4), 1999, pp. 333–351.
- [5] Weinberg, G.M. *Quality Software Management: Systems Thinking*. Dorset House Publishing, 1992.
- [6] Reinertsen, D.G. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.
- [7] Thomke, S., and Reinertsen, D. “Six Myths of Product Development.” *Harvard Business Review*, 90(5), May 2012, pp. 84–94.
- [8] lateralworks. *The Optimal Project Load: How Dedicated Resources Accelerate New Product Development*. FTTM Best Practices white paper, March 2026.
- [9] lateralworks. *FTTM New Product Development Best Practices*, revision 2018.002. lateralworks research monograph, 2018.
- [10] Khurana, A., and Rosenthal, S.R. “Towards Holistic ‘Front Ends’ in New Product Development.” *Journal of Product Innovation Management*, 15(1), 1998, pp. 57–74.
- [11] Smith, P.G., and Reinertsen, D.G. *Developing Products in Half the Time*. Van Nostrand Reinhold, 1991.
- [12] Saaty, T.L. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill, 1980.
- [13] Rogers, P., and Blenko, M. “Who Has the D? How Clear Decision Roles Enhance Organizational Performance.” *Harvard Business Review*, January 2006.
- [14] lateralworks. *Portfolio prioritization & starts control*. lateralworks briefing paper, May 2026.  
<https://lateralworks.com/papers/portfolio-prioritization-and-starts-control.pdf>
- [15] Isaacson, W. *Steve Jobs*. Simon & Schuster, 2011, pp. 337–339.
- [16] Isaacson, W. “The Real Leadership Lessons of Steve Jobs.” *Harvard Business Review*, April 2012.
- [17] Robertson, D., and Breen, B. *Brick by Brick: How LEGO Rewrote the Rules of Innovation and Conquered the Global Toy Industry*. Crown Business, 2013.
- [18] Henderson, R.M., and Reavis, C. “Corning Incorporated: The Growth and Strategy Council.” MIT Sloan Teaching Innovation Resources, Case 08-056, 2008 (rev. 2009).  
<https://mitsloan.mit.edu/teaching-resources-library/corning-incorporated-growth-and-strategy-council>
- [19] Wheelwright, S.C., and Clark, K.B. *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*. Free Press, 1992.
- [20] lateralworks. Client engagement archive: division planning, project prioritization, and zero-based starts control deployments, 2012–2019. Internal deployment materials (client identities removed).