



Case study

# The schedule mirrors the team

Why functional project managers build functional schedules — and what it takes to move them

## **FTTM coaching series.**

A representative case from a global semiconductor program serving AI data-center customers. Names and sector-identifying details have been removed.

---

**Prepared by**

lateralworks  
FTTM methodology

**Date**

April 2026  
Coaching case study

**Online**

lateralworks.com  
FTTM coaching series

## Table of contents

# The schedule mirrors the team

---

<b>The setup and the diagnostic</b>	<b>04</b>
<b>The team-levels lens</b>	<b>08</b>
<b>Seven philosophical splits</b>	<b>11</b>
<b>Why this is structural, not personal</b>	<b>14</b>
<b>A learning opportunity, not a critique</b>	<b>16</b>
<b>The coaching arc</b>	<b>19</b>
<b>Implications for technology executives</b>	<b>21</b>
<b>References</b>	<b>23</b>

**Core thesis.** A schedule is the visible artifact of an invisible operating model. When a functional organization adopts cross-functional tools, the organization reshapes the tools faster than the tools reshape the organization. Real change comes from moving the team from lightweight coordination to heavyweight ownership; the schedule will follow.

## Overview

# Abstract

---

**A schedule shows you how its author thinks about the work**, and how the organization above the author expects the work to be run. This case study covers a coaching engagement at a global semiconductor company on a new-product program for AI data-center customers. The senior project manager assigned to the program left a Fast Time To Market (FTTM) workshop with a cross-functional schedule and a heavyweight team structure. Three weeks later she had quietly rebuilt that schedule into the functional shape she had used for twenty years.

The pattern shows up in every senior project manager whose career has been spent inside a functional or lightweight organization. Lightweight PMs coordinate functional teams they do not own. Their schedules show it: tasks grouped by function, milestones stacked at the top for executive consumption, dependencies looping upward inside each functional cluster, and the plan read from a table instead of a Gantt. The schedule becomes a private database for the PM. The team has no way in.

Heavyweight project managers, the third level on the Wheelwright-Clark team continuum [1, 2], own the cross-functional flow of the work and structure their schedules around how the product matures from concept to qualified volume. Functional views come from filtering the master plan. The Gantt is the team's shared picture of the work; the table is one way to query that picture.

This paper covers seven philosophical splits that surfaced between the FTTM methodology and the senior PM's existing practice, shows why those splits are structural rather than personal, and outlines the coaching arc for moving mature PMs from lightweight to heavyweight ownership. It also clarifies a point that gets lost in technical argument. The methods being introduced come from years of work on products that reached market quickly. The team has not been asked to use them before. Every PM and core-team member who learns them now adds a capability that travels.

The conclusion reaches beyond one program. In a legacy operating model, the schedule does not change until the team does, and the team does not change until the host gives the PM authority over the work and tells her clearly that this new mode is what the company is asking her to learn.

# 01

## Diagnostic

# **The setup and the diagnostic**

---

A senior project manager at a global semiconductor company, running a complex new-product program for AI data-center customers, left an FTTM workshop with a cross-functional schedule and a heavyweight team structure. Within weeks the schedule had been quietly rebuilt into the functional shape she had used for two decades. The structural logic the team agreed to at the workshop was still visible underneath. The operating model she worked in every day had written over the top of it. This section covers what we observed and what the schedule revealed about the organization underneath.

## Context

# A complex program in a legacy operating environment

---

The company is a top-five global semiconductor manufacturer with tens of thousands of employees across design centers, fabs, and qualification labs in three regions. Its product portfolio includes advanced nodes for AI data-center accelerators sold to hyperscalers and AI-infrastructure customers. These are programs measured in hundreds of millions of units, with multi-year design-in cycles and revenue opportunities that anchor the company's growth trajectory. A single qualification slip on one of them can move the parent's quarterly guidance.

The program at the center of this case study is a new-product development with high-volume production targets, a tight competitive window against AI-accelerator alternatives from rival foundries and IDMs, and a schedule that touches every function in the company: design, foundry release, multi-site assembly and test, reliability qualification, system-level validation, and customer engagement. The project development team is staffed cross-functionally, but it is structured as a lightweight coordination layer. The PM has visibility into all functions and authority over none of them.

### Why FTTM was introduced

Senior leadership engaged lateralworks because the company's prior programs had developed a familiar pattern: schedule slips that compounded in the back half, late integration discoveries that forced multiple silicon spins, and customer commitments delivered six to twelve months later than the original target. The diagnosis was consistent across post-mortems. Each function ran its own plan. Integration risk surfaced late. The cross-functional project manager spent more time chasing status than driving decisions.

The leadership ask was specific: adopt a methodology that gets the company to market faster on the next AI data-center program, with predictable execution. FTTM is built for exactly this problem. It shifts the operating model from functional sequencing through coordination handoffs to cross-functional flow owned by an empowered core team. The methodology is grounded in heavyweight team principles articulated by Wheelwright and Clark in their study of fast-cycle product development [1, 2], extended through three decades of lateralworks engagement data across hardware, semiconductor, and complex-systems programs [3].

### The senior PM

The PM running the program is a 20-year veteran. She is technically fluent, organizationally credible, well-regarded by the functions, and trusted by the executive layer above her. She has also spent her career as a textbook lightweight PM: coordinating functional teams that report elsewhere, building schedules that answer to gates rather than drive flow, and answering executive questions with tabular data drawn from spreadsheets. She is good at the work she has been doing, and she has been promoted for it.

She attended the FTTM workshop. She participated. She left with a schedule built around process flow and product maturity, with a cross-functional dependency structure, and with a heavyweight orientation toward the work. Three weeks later, when we sat down with her to review the refined version of that schedule, the cross-functional spine had been overwritten. The structure was still recognizable in places, but the spine had been replaced with the functional groupings she had used her entire career.

## What the schedule revealed

When we ran the FTTM Analyzer against the refined plan, the diagnosis was immediate. The schedule had been functionally restructured. The process-flow architecture from the workshop survived only as fragments: section labels, a few cross-functional links. The spine had been replaced.

The most reliable diagnostic is the upward link. In a cross-functional schedule, dependencies flow forward and downward through the WBS as the product matures: design feeds engineering builds, builds feed qualification, qualification feeds customer release. The plan reads as a story of progressive product maturity. In a functional schedule, dependencies leap up and down inside each functional cluster — test tasks chain to test tasks, fab tasks chain to fab tasks — because the work has been grouped by who does it rather than by when it happens in the product's journey. The Analyzer flagged dozens of upward links across the refined plan.

Reinertsen reaches a parallel conclusion in his treatment of product-development flow [4]. Invisible queues and hand-off discontinuities are the root cause of slow programs, and they are easiest to spot when the work is mapped as a forward flow rather than as a static functional inventory. A functional schedule hides queues inside each cluster. A cross-functional schedule exposes them at the seams where one function is waiting on the previous one.

## The Gantt-as-table problem

A second pattern surfaced almost immediately. The PM does not use the Gantt view. She reads the schedule from the left-hand text rows and traces dependencies by referring back to the table. She reported, without irony, that she "can do that in the words." She has never used the Gantt as her primary view.

A schedule read from a table is a private database for one person. A schedule read from a Gantt is a communication tool for a team. The Gantt carries information the table cannot show concisely: concurrency, critical-path geometry, slack, the visual signature of a healthy versus a stressed plan, and where pull-in opportunities live. A team that cannot read the Gantt cannot engage with the plan, and a team that cannot engage with the plan cannot own its commitments. The lightweight PM, running the schedule alone, can work from the table. The heavyweight PM, running the schedule as a team artifact, depends on the Gantt.

## Functional versus cross-functional structure

The disagreement on schedule structure is really a disagreement about who owns the work. The lightweight PM organizes by function because, in her operating world, the functions are her counterparties: separate organizations she must coordinate, each with its own leader and its own priorities. Grouping the schedule by function mirrors how she actually moves through her week — a meeting with test, a meeting with fabs, a meeting with QA, a meeting with the customer. The schedule is an accounting of those conversations.

The heavyweight PM organizes by product flow because, in her operating world, the product is the unit of work and the team is the unit of execution. Functions participate in the flow rather than running parallel lanes that the PM has to keep aligned. Each task carries a functional code, so a function-specific view is one filter away. But the master plan is structured around how the product matures from concept to qualified volume, because that is how the team experiences progress.

**The mechanical truth.** A cross-functional schedule yields a function-grouped view at any time through filtering. A function-grouped schedule cannot yield a clean cross-functional view without rebuilding. The asymmetry argues for the cross-functional spine.

# 02

## Framework

### The team-levels lens

---

The most useful frame for understanding what we observed is Wheelwright and Clark's four-level continuum of project team configurations [1, 2]. The continuum runs from a pure functional hierarchy, where there is no project ownership at all, to fully autonomous tiger teams that are extracted from the functional organization entirely. Most companies live somewhere in the middle two levels. lateralworks engagement data places legacy semiconductor companies almost invariably between Level 1 and Level 2.

## Framework

# Four configurations on the Wheelwright-Clark continuum

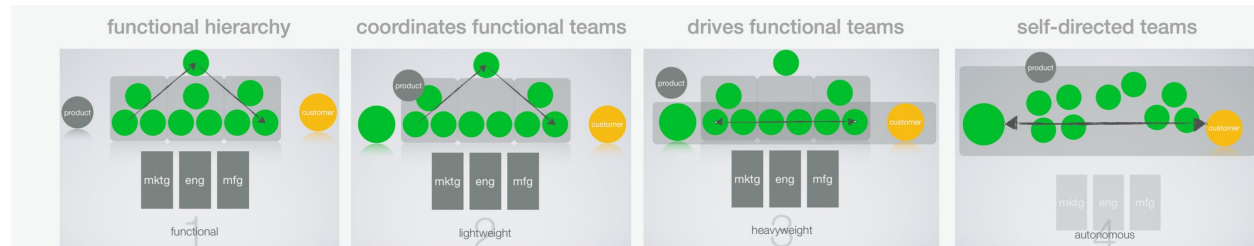


Figure 1. The four team configurations on the Wheelwright-Clark continuum. Functional hierarchy (1): no project accountability; the product flows through functional gates. Lightweight (2): a project manager coordinates functional teams without authority. This is where most legacy semiconductor PMs operate. Heavyweight (3): a strong project manager drives the functional teams; the core team is empowered and accountable. Autonomous (4): the team is extracted entirely from the functional organization.

### Where the PM in this case study lives

The senior PM in this engagement is operating squarely at Level 2, the lightweight configuration. She is a credible, senior coordinator who does not own the functional teams she works with. The functional leaders own those teams; she influences. She runs status, escalates risk, coordinates dependencies, and defends the timeline upward to executives. The schedule she builds reflects that posture exactly: a coordination artifact rather than a team operating system.

Twenty years of operating at Level 2 have produced a settled view of what a schedule is for, who reads it, and how it should be structured. The view is correct for Level 2. It does not transfer to Level 3, and FTTM assumes Level 3.

### Where the program needs the PM to live

The AI data-center program cannot be run successfully at Level 2. The integration complexity is too high, the cross-functional coupling is too tight, the customer expectations are too aggressive, and the cost of late discovery is too large. Programs of this scope demand heavyweight ownership: a project manager who drives the functions toward integrated product maturity rather than coordinating their independent timelines, supported by a cross-functional core team whose members are accountable to the program rather than to their functional homes.

Moving from Level 2 to Level 3 is an operating-model change, not a tooling change. New tools (a different schedule structure, a Gantt-first reading discipline, the Analyzer's diagnostics) matter, but they are downstream of the structural shift. The host has to grant the PM authority over the work. The PM has to take it. The functions have to release it. None of that happens because a schedule was redrawn at a workshop.

**Implication for the host.** Tools migrate downward. If the organization above the PM continues to operate Level 2 (managing the program through functional reviews, holding functional leaders accountable for functional milestones), the PM cannot operate Level 3. The schedule structure is the visible symptom. The authority structure above the PM is the cause.

# 03

## Specifics

### **Seven philosophical splits**

---

A two-hour working session with the PM produced seven specific disagreements about how the schedule should be structured and read. Six are real philosophical splits, coherent and well-defended, rooted in the lightweight operating model. The seventh was a tactical proposal that we conceded after pushback. Together they map the lightweight and heavyweight scheduling worldviews against each other.

# The catalogue

## What surfaced in the session

---

#	Issue	Heavyweight / FTTM position	Lightweight / functional position
1	<b>Functional grouping vs. cross-functional flow</b>	Organize the schedule around how work flows through the product maturity arc: cross-functional, with flow as the spine. Code each task with its function so a function-specific view is one filter away. Functional grouping forces dependencies up and down, undermines Gantt readability, and turns the schedule into a coordination artifact rather than a team communication tool.	Group tasks by function: all test together, all fabs together, all qualification together. Easier to read by function, matches how the organization actually works, and tasks rarely run perfectly in sequence anyway. The cross-functional layout feels chaotic.
2	<b>Milestones embedded vs. stacked at the top</b>	Milestones live where they naturally exist in the flow, marked as internal dependencies and pulled together via filters when a reporting view is needed. Embedding plus filtering yields both views from one structure.	Keep gates, customer-readiness milestones, and qualification milestones grouped at the top of the file. Quick to screenshot for executives; easy to track from the timeline view.
3	<b>Predecessors going up (uplinks)</b>	Allowed but exceptional. Uplinks hide errors. Once there are many of them, a bad one created accidentally becomes invisible. When something flows up, bring it down into its true grouping; package tasks where they actually flow together.	Uplinks are unavoidable inside any sensible functional grouping (e.g., test tasks naturally chain to other test tasks). Breaking the function apart to avoid them is worse than the problem.
4	<b>Blank rows vs. summary milestone pairs</b>	Use summary milestone pairs to separate phases. Same visual separation; gives proper structure with automatic dependency roll-up; lets the schedule collapse to a handful of high-level activities for executive views.	Blank rows separate phases. Personal preference. Cleaner to view. The summary-milestone logic is acceptable but not compelling.
5	<b>Reading the schedule: Gantt vs. text</b>	The Gantt is the primary view. Critical path is followed there, pull-in opportunities are found there, and the team sees the integrated plan there. The text view is a secondary tool for editing detail.	Reads the text view exclusively. Traces dependencies through the rows. Does not use the Gantt; reports being able to "do that in the words." Has never used the Gantt as her working view.
6	<b>Calendar: elapsed days vs. standard</b>	Days only, on a standard calendar. Add non-working time as standard-calendar exceptions. Elapsed-day units misrepresent true duration and break tools that expect calendar-aware math.	Used elapsed days, weeks, and a custom calendar with non-working time blocked out. The conversion to standard days appeared to pull the program in by months and forced a rollback to the custom calendar.

#	Issue	Heavyweight / FTTM position	Lightweight / functional position
7	<b>Product-line grouping (high-spec / low-spec)</b>	Reorganize qualification lots so each product variant forms its own product group with its fab and qualification tasks together, rather than splitting all fabs from all qualifications across variants. Add a single end milestone gathering all lots.	A consolidated end milestone is not actionable; only individual lot completions feeding into post-process matter. The PM pushed back; we conceded the specific reorganization.

*Figure 2. The seven splits surfaced in the schedule review. Items 1–6 are real philosophical disagreements rooted in lightweight versus heavyweight operating models. Item 7 was tactical and conceded after pushback.*

# 04

## Interpretation

### **Why this is structural, not personal**

A reflexive reading of the case is that the PM is rigid, defensive, or unwilling to learn. That reading is wrong, and following it would lead the host into a counterproductive coaching posture. The PM is doing exactly what the operating model she has worked in for two decades trains a senior coordinator to do. Her schedule is internally consistent with that model. The model is mismatched to the program.

## Diagnosis

# The pattern is structural, not personal

---

### The defensive-reasoning trap

Argyris's study of why expert professionals struggle to learn applies directly here [5]. Senior professionals build their careers on competent execution. They rarely experience public learning-related failure. When their core practice is challenged, the natural response is defensive reasoning: rationalize the existing practice, shift the frame to one in which the existing practice is correct, or acknowledge the new framework in principle while continuing the old practice in detail. Argyris shows this is a predictable cognitive pattern in highly successful professionals, not a character flaw.

We observed exactly this pattern in the session. The PM did not argue against FTTM in principle. She agreed with the workshop output. She rebuilt the schedule into the functional shape anyway. Asked directly whether her uplinks contradicted the methodology, she defended them on functional-grouping grounds. Asked about the Gantt, she reported being able to do everything from the text. Each response is internally coherent within her operating model, and each is a small act of conserving the model against the new framework.

### The pattern across mature PMs

lateralworks engagement data, collected across more than a hundred programs at semiconductor, hardware, and complex-systems companies over the past decade, shows that this is the rule rather than the exception in mature PMs working in lightweight organizations [3]. The schedule structure tracks the operating model closely. PMs from heavyweight organizations build heavyweight schedules without coaching. PMs from lightweight organizations rebuild any schedule, however well structured at hand-off, into the lightweight shape within weeks.

The implication is direct. A methodology rollout that targets the PM in isolation, while leaving the operating model around her unchanged, does not stick. The unit of intervention is the host, and the team is where you measure whether the intervention worked.

# 05

## Reframe

### **A learning opportunity, not a critique**

The way lateralworks plans and manages projects comes from decades of work with products that reached market quickly: fast-cycle programs in semiconductors, hardware, and complex systems where time-to-market gaps determined commercial outcomes. The host engaged us specifically to compress time-to-market on the AI data-center program. That is the mandate. The methods we bring exist because they have produced fast outcomes elsewhere; the company is asking the team to learn them.

## Reframe

# The team has not failed. They are being asked to learn.

---

Surfacing seven specific disagreements about how a schedule should be structured can drift into something it should never become: an evaluation of competence. The PMs at this company are highly capable in their existing operating context. The senior PM in this case study has earned a twenty-year career by being good at her job. Our diagnosis is not that she is a poor project manager. It is that she is a strong lightweight project manager working in a program that needs heavyweight ownership, and that nothing in her operating environment has previously required her to develop the methods being introduced.

That distinction matters for engagement. Argyris's defensive-reasoning trap [5] fires most reliably when accomplished professionals perceive that their existing practice is being judged. It fires least reliably when those same professionals understand that they are being asked to add a capability they have not previously been asked to develop. Reframing the engagement around what the team is learning, rather than around what the team has been doing wrong, disarms the trap at the front end and makes every subsequent coaching conversation easier.

It also matters for accuracy about what the methods are. FTTM is a body of practice developed and refined across hundreds of fast-cycle programs in industries where speed is the primary competitive lever. The specific techniques the senior PM is being asked to adopt (cross-functional flow as the schedule spine, Gantt-first reading, standard calendars, summary milestone pairs, embedded gates) are the operational signatures of teams that ship faster than their competitors. They are taught because they work, and because the host is paying for the company to learn them.

### A career advantage, not a compliance task

There is a message here for the PMs and core-team members on this program that risks being lost in the technical argument. The methods being introduced are durable, transferable career capabilities. A project manager who has internalized fast-cycle methodology (who knows how to structure a cross-functional schedule, how to drive a heavyweight team, how to read a program from a Gantt) carries that capability forward to every program, every role, and every company they work at thereafter. The attention required to learn it now compounds for the rest of a career.

For the PDT leads, PMs, and core teams currently inside this engagement: this is an opportunity to add a body of practice to your toolkit that most program managers in legacy semiconductor environments never get exposure to. The companies that compete on time-to-market in AI data-center products will be staffed, at every level, with people who run programs heavyweight. Being one of those people, starting now, is worth more than any specific outcome on this one program.

### What the host owes the team

The reframe is genuine only if the host carries it through. If senior leadership engaged lateralworks to bring fast-cycle methods into the company, that intent has to reach the PDT leads, the PMs, and the core teams as an invitation to learn, with the structural changes the methodology presumes behind it. Asking the team to develop heavyweight capabilities while the operating model continues to hold them in lightweight roles is an unfunded mandate, not a reframe.

**The message that needs to land.** The host engaged lateralworks to help the company get to market faster. The PMs and core teams on this program have been asked to operate in the mode the company has historically used; adopting FTTM is a developmental step the company is investing in. Every PDT lead, PM, and core-team member who takes it on emerges with a capability that travels with them.

# 06

## Path forward **The coaching arc**

---

Moving a mature PM from lightweight Level 2 to heavyweight Level 3 is rarely a straight line. The literature on professional learning and our own engagement experience converge on a multi-step progression that respects the PM's expertise while reframing the context in which that expertise operates. This section outlines the five steps we use, in the order they tend to matter.

## Sequence

# Five steps, in the order they matter

---

### Step one: reframe, do not replace

The PM's lightweight competence is real and valuable. The coaching frame is not "what you have been doing is wrong"; it is "what you have been doing is right for the operating model you have been in, and we are now changing the operating model." This reframing matters more than any specific tooling change. It preserves the PM's standing, brings her expertise into the new model, and avoids triggering the defensive reasoning that derails most methodology rollouts.

### Step two: change the host signals

The PM cannot operate heavyweight if her counterparties are still operating lightweight. Senior leadership has to send unambiguous signals through review cadences, escalation paths, and which milestones get attention in operating reviews. The signal is that the program is now run cross-functionally and that the PM owns the integrated outcome. Without it, the PM's adoption of cross-functional structure looks like cosmetic compliance and gets reversed at the first real conflict.

### Step three: small structural wins

Once the host signals are aligned, structural changes become possible without theological argument. Move milestones into the flow rather than stacking them at the top. Convert blank rows to summary milestone pairs. Code tasks with functional tags so function-specific filters become trivial. These are small, reversible changes individually. Cumulatively they reshape the schedule from a coordination database into a team operating system.

### Step four: switch the primary view

The hardest behavioral change is moving the PM's primary working view from the text rows to the Gantt. This rarely happens through argument. It happens when the team begins running working sessions in front of the Gantt, finding pull-in opportunities, identifying critical-path risks, debating sequencing, and the PM discovers that the Gantt is where the conversation actually happens. The text view becomes the editing interface it should always have been.

### Step five: openness as a behavior

A finding from decades of FTTM work: willingness to learn is the single best predictor of which mature PMs make the transition to heavyweight ownership. PMs who can hold their existing practice lightly enough to test alternatives, including alternatives they initially disbelieve, make the move. PMs who cannot do not, regardless of credentials, tenure, or technical skill. Openness is a behavior the host can reinforce through how it responds to experiments, errors, and reversals.

# 07

## Takeaways

# **Implications for technology executives**

Three implications follow for executives running large complex programs in legacy operating environments, particularly semiconductor and hardware leaders pursuing AI data-center opportunities where time-to-market gaps translate directly into design-win outcomes.

## Takeaways

# Three implications for senior leaders

---

### Read your schedules

Schedules are diagnostic. A well-built schedule reveals the operating model of the team that built it. Cross-functional spine plus a Gantt-driven team session implies heavyweight ownership. Functional clustering, top-stacked milestones, and a PM who reads from the table imply lightweight coordination. Senior leaders who inspect schedules with this lens read the operating model directly off the document.

### Decide which level you are operating at, and act accordingly

Most legacy semiconductor companies sit between Level 1 and Level 2. Moving even one level (to a true Level 2 or, more ambitiously, to Level 3) produces a measurable improvement in time-to-market and integration quality. The move requires concrete changes in authority, accountability, review cadence, and incentive alignment. A methodology rollout without those structural changes is theatre.

### Invest in the host, not just the team

The most expensive misallocation in methodology rollouts is heavy investment at the team layer with little or no investment at the host layer above. The team can be trained, equipped, and given new tooling. None of it survives contact with a host that continues to reward lightweight behavior. Intervene at the host layer; measure at the team layer.

### Transmit the learning frame

The message that needs to reach every PDT lead, PM, and core-team member is that the company is investing in giving them fast-cycle methodology, and the rollout is a development opportunity rather than an audit of past performance. Executives who transmit that frame clearly, and back it with the operating-model changes the methodology presumes, get adoption. Executives who leave the team to interpret the rollout on its own get cosmetic compliance.

**The bottom line.** For high-stakes AI data-center programs where every quarter of slip costs hundreds of millions in design-win position, the right question is whether the operating model gives the PM the authority to drive the program cross-functionally, and whether she has been told clearly that the company is asking her to learn something new. The schedule will show you the answer.

## Sources

# References

---

- [1] Wheelwright, S. C., and Clark, K. B. *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*. New York: Free Press, 1992.
- [2] Clark, K. B., and Wheelwright, S. C. "Organizing and Leading 'Heavyweight' Development Teams." *California Management Review* 34, no. 3 (Spring 1992): 9–28.
- [3] lateralworks. "Internal engagement assessment database." Aggregate observations across semiconductor, hardware, and complex-systems programs, 2015–2026.
- [4] Reinertsen, D. G. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Redondo Beach, CA: Celeritas Publishing, 2009.
- [5] Argyris, C. "Teaching Smart People How to Learn." *Harvard Business Review*, May–June 1991, pp. 99–109. <https://hbr.org/1991/05/teaching-smart-people-how-to-learn>
- [6] Smith, P. G., and Reinertsen, D. G. *Developing Products in Half the Time: New Rules, New Tools*. 2nd edition. New York: Wiley, 1998.
- [7] Pisano, G. P., and Wheelwright, S. C. "The New Logic of High-Tech R&D." *Harvard Business Review*, September–October 1995, pp. 93–105.
- [8] House, C. H., and Price, R. L. "The Return Map: Tracking Product Teams." *Harvard Business Review*, January–February 1991, pp. 92–101.
- [9] Ericsson, K. A., Krampe, R. T., and Tesch-Römer, C. "The Role of Deliberate Practice in the Acquisition of Expert Performance." *Psychological Review* 100, no. 3 (1993): 363–406.
- [10] Dweck, C. S. *Mindset: The New Psychology of Success*. New York: Random House, 2006.