Whitepaper

Change control

How empowered core teams govern the frozen specification and protect the launch date

FTTM practice series.

Why the fastest product organizations put every specification change through one board, one schedule, and one economic model, and why that board is the core team itself.

Prepared by	Date	Online
lateralworks	July 2026	lateralworks.com
FTTM methodology	Practice paper	FTTM practice series

Table of contents

Change control

Core thesis. Time to market and change control are the same subject. Every change to a frozen specification carries a schedule cost and a financial cost, and an organization that cannot price those costs will accept every change and miss every date. The fastest organizations give one small board authority over the specification, arm it with a live schedule and a cost-of-delay model, and make that board the core team itself.

Overview

Abstract

Late products rarely die from one big decision. They bleed out through small ones. A feature added in a hallway conversation, a spec parameter an executive adjusts after a customer visit, a "quick" rewrite following an early study. Each change looks cheap on its own. None of them is priced. The schedule absorbs them silently until the committed date is fiction, and by then nobody can say which decisions spent the margin.

This paper treats change control as a time-to-market discipline. The mechanism is old and simple: a frozen specification under revision control, a change control board (CCB) that is the only route by which the baseline moves, and a decision rule that weighs the benefit of each change against the cost of the delay it creates. What makes the mechanism fast rather than slow is who sits on the board and how much freedom they have.

In the FTTM model the CCB is the core team itself: product, engineering, and operations, joined by procurement or legal when a change touches contracts or supply. The team owns the specification, meets only when there is a change to decide, and operates at freedom level 1 or 2 on the Oncken scale: it acts, then informs the executive staff of the decision. Executives who reserve spec decisions for themselves become the de facto change board, and a team waiting for permission cannot hit an aggressive date.

The paper walks through the cost-of-delay model that converts schedule days into dollars, then shows the mechanism in practice at Boeing, Maersk Line, and Hewlett-Packard. It closes with a practical charter: membership, cadence, decision rules, and the freedom-level agreement that keeps executives coaching instead of deciding.

01

The problem

What uncontrolled change costs

Every product organization believes it controls change. Almost none measures what change costs. The gap between those two statements is where schedules go to die: industry surveys put uncontrolled scope change in half of all projects, and the financial penalty lands not in the development budget but in the market window.

Section 01

Schedule creep is a pricing failure

The Project Management Institute found that 52 percent of projects completed in the preceding year experienced scope creep or uncontrolled change, up from 43 percent five years earlier [1]. The direction of that trend matters as much as the level: markets now demand faster delivery and more customer input, which means more change pressure on every baseline, not less. PMI's case files show where the pressure ends up: the Royal Adelaide Hospital program absorbed feature requests until it ran AU$640 million over its AU$2.3 billion budget [2].

Development cost is the wrong place to look for the damage, though. The classic McKinsey analysis by Donald Reinertsen made the point thirty years ago and it still holds: under typical market conditions, a product that ships six months late earns about 33 percent less profit over five years, while a product that ships on time but 50 percent over its development budget gives up only around 4 percent [3, 4]. Time is the expensive axis. Every uncontrolled change spends it.

Uncontrolled does not mean large. The changes that kill schedules are individually reasonable: a screen added to a probe, or a feature pulled forward because a competitor announced something. Each arrives without a price tag. Nobody computes the schedule impact, so nobody can weigh it, so the default answer is yes. The organization discovers the cumulative cost months later as a slipped date it cannot explain.

The freeze-thaw-replan cycle

The usual response is the opposite failure. Burned by creep, management freezes the specification and forbids further change. The world keeps moving anyway. Customer needs shift, a competitor ships, a technical assumption breaks, and reality eventually forces a major replan to a new "really frozen" spec. lateralworks field research identifies this freeze, thaw, replan cycle as one of the primary causes of time-to-market overrun: the change volume arrives anyway, but late, in a lump, when the rework penalty is highest [5]. Best-practice organizations run the opposite profile. They front-load change while it is cheap, then taper it as development proceeds, using a flexible containment envelope rather than a hard freeze: the spec is under version control, changes are allowed, and every change passes through a gate that prices it [5].

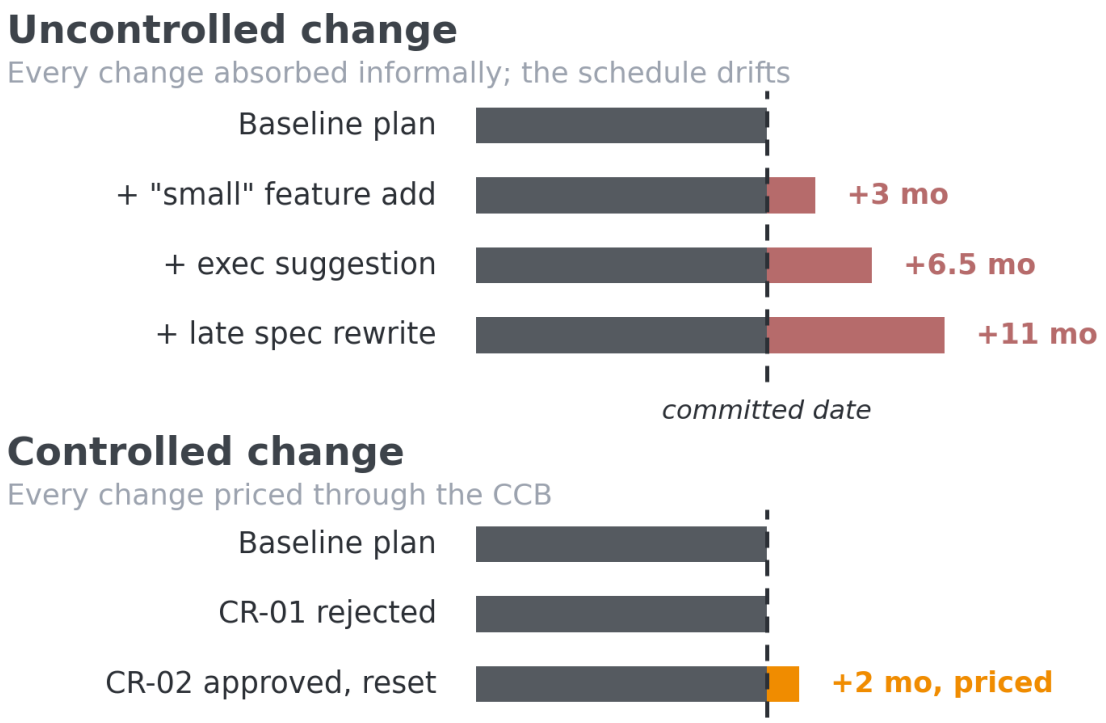

Figure 1. The same change pressure, two outcomes. Absorbed informally, changes accumulate into unexplained schedule creep. Priced through a CCB, the baseline holds or moves deliberately.

02

The model

Pricing change with the cost of delay

A change control board without an economic model is just a slower way to argue. The board can collect opinions about whether a feature matters, but opinions never settle against each other; rank settles them, and the loudest sponsor wins. What converts argument into decision is a number: the cost of one day of delay on this program, multiplied by the number of days the change adds to the critical path.

That number comes from the market, not from the project. Products sell into a window. Enter early and the product commands the full ramp, the highest prices, and the strongest margins. Enter late and it fights for a shrinking market at falling prices. This section draws the window, builds the delay model from it, and shows how the board plays each change request against the live schedule to reach a decision.

Section 02

Every change has a schedule price

Cost of delay is the profit a program forgoes for each unit of time its output arrives late. Reinertsen, who spent a career quantifying product development economics, called it the one number an organization should compute if it computes nothing else [6]. The model does not need to be elaborate. A first-order version takes the product's expected peak revenue, the market growth rate, the price erosion rate, and the window length, and produces a dollar figure per week of delay. Programs that build the model are routinely surprised by its size: when analysts at Maersk Line priced delay on their development portfolio, one capability came out above $200,000 per week. It had been sitting in queues for 38 weeks, an invisible $8 million loss no one had authorized [8, 9].

The shape behind the number is the market window. lateralworks has used the same illustration in its cost-of-delay work for years: the potential revenue cycle drawn against time, once for on-time entry and once for late entry, under a margin line that erodes across the whole window [7]. The late entrant starts after the delay, meets a market already peaking, and captures a fraction of the cycle. The damage runs past the revenue gap alone: the earliest units carry the highest prices and margins, late units sell into price erosion, and the late entrant is still paying development expense while the on-time entrant is already shipping. A product that misses the front of the window does not earn proportionally less; it earns disproportionately less.

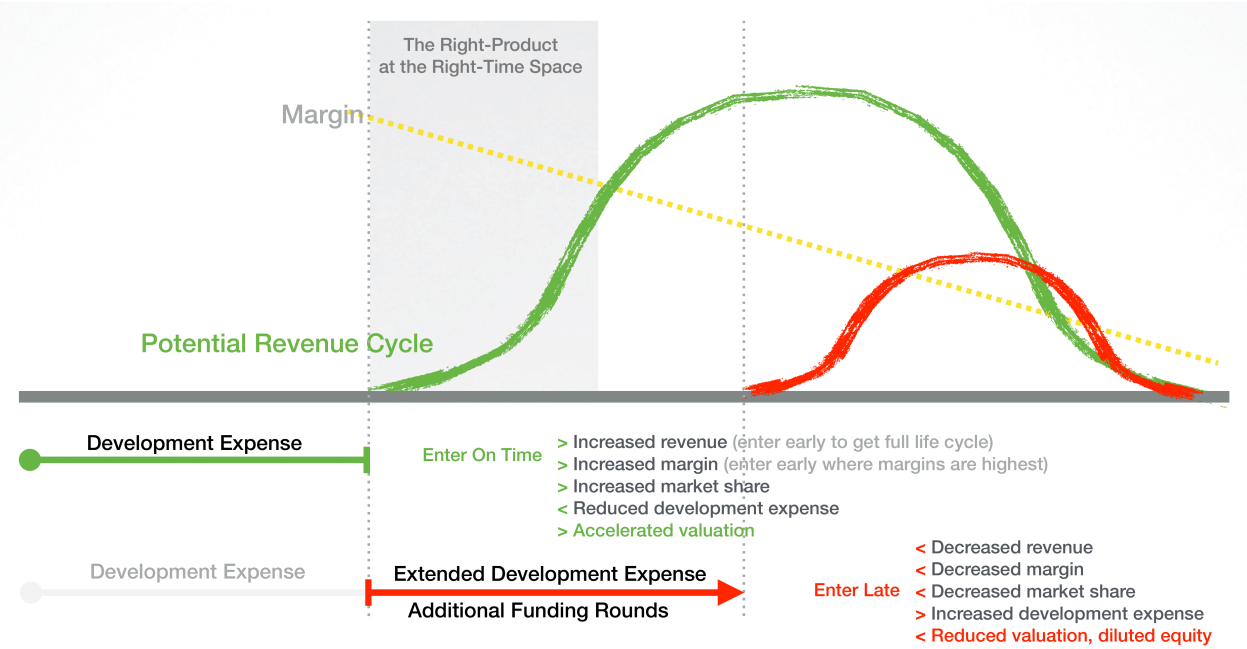

Figure 2. The lateralworks cost-of-delay illustration [7]. On-time entry captures the full potential revenue cycle at the highest margins; late entry buys a compressed cycle, eroded margins, and extended development expense.

With the model in hand, the CCB decision becomes arithmetic plus judgment instead of rank plus volume. A change request arrives. The team prices its schedule impact against the live critical path: four months, say. The cost-of-delay model converts four months into dollars. The sponsor of the change states the benefit: 20 percent more unit sales, say, or a contract retained. If the benefit clearly exceeds the delay cost, the board accepts the change, accepts the slip, resets the target, and records the revision. The new date is legitimate because it was bought knowingly. If the delay cost dominates, the change goes to the next version and the baseline holds.

The schedule is the instrument

Everything in this comparison hangs on one piece of data: the schedule. A change request lands somewhere in the network of tasks, and only the schedule can convert it into time, the +x days it adds to the critical path. That number in days is what feeds the cost-of-delay model, and the model converts the days into dollars for the cost-benefit comparison. Nothing else in the program can produce x. Opinion cannot produce it. The requester cannot produce it. Only a schedule that models the real dependency network can say whether a change costs three days of float or four months of critical path.

This is also where change control quietly dies in most organizations: the schedule is not current, so x is fiction, and every number downstream inherits the fiction. A plan that was last leveled at the quarterly review cannot price a change submitted this week. The board needs a schedule that accurately represents the future as the team now understands it, refreshed weekly, with the critical path recomputed as work completes and estimates move [10]. A change decision therefore has exactly two elements: the schedule impact in days, and the cost-of-delay analysis those days feed. Remove either one and the CCB is reduced to voting on opinions.

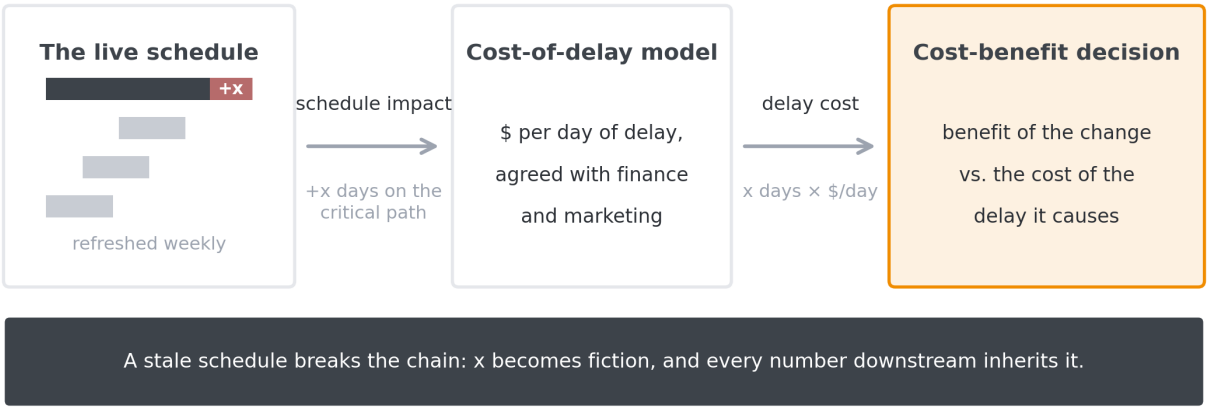

Figure 3. The data chain behind every change decision. The live schedule prices the change in time; the cost-of-delay model prices the time in dollars; the board weighs the dollars against the benefit.

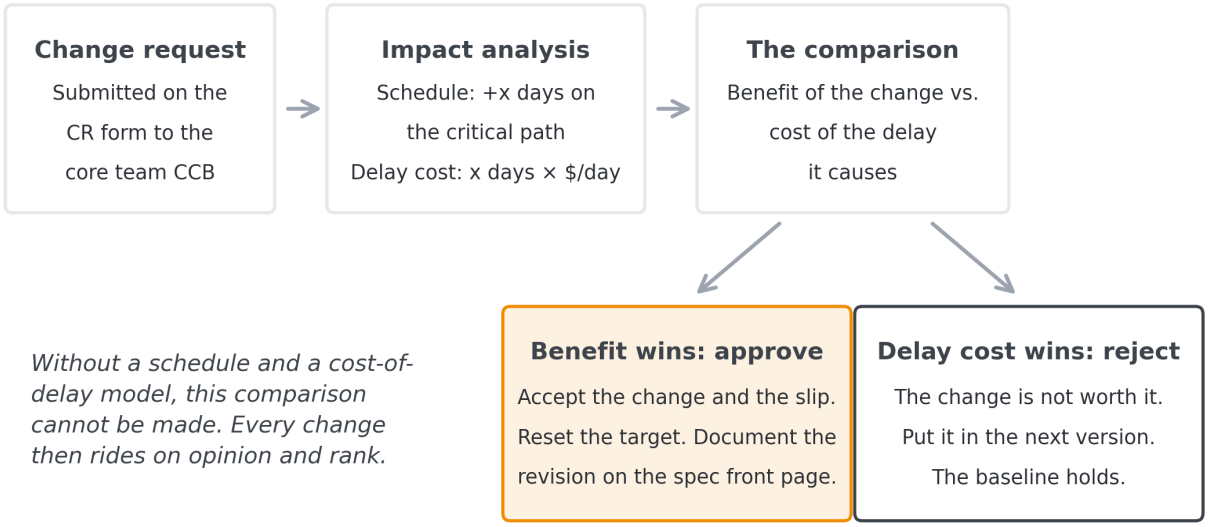

Figure 4. The CCB decision rule. The comparison in the third box is only possible when a live schedule and a cost-of-delay model both exist.

When people see the price, they reconsider

The most reliable effect of pricing changes is that many of them evaporate. A request that sounded urgent as a hallway suggestion looks different framed as "four months and $3.4 million of delay cost against $1.1 million of incremental margin." In lateralworks engagements, sponsors withdraw a large share of change requests once the schedule impact is on the table. The pitch meeting never happens because the pitch no longer makes sense. The board earns its keep as much through the requests it deters as through the ones it decides.

03

The mechanism

The change control board

The mechanism itself is deliberately unglamorous: a controlled document, a small board, a request form, four possible outcomes, and a meeting that only happens when there is something to decide. This section specifies each piece.

Section 03

A small board with a frozen document

The controlled document

Change control starts with something worth controlling: a written product specification, under revision control, owned by the core team. The front page carries the revision history (date, change, approver) so any reader can see what moved and when. Once the spec is approved it becomes a frozen baseline, and the CCB is the only route by which that baseline moves. A specification that anyone can edit is not a baseline; it is a wiki. lateralworks best-practice research is blunt on this point: best teams install version control on external and internal specifications, and they treat the change-authorization trail as part of the product [5].

Membership

The board is small: three to five voting members, and an odd number breaks ties cleanly. Product leadership chairs it and owns the controlled document. Engineering brings feasibility and schedule truth. Operations or manufacturing brings supply, cost, and ramp reality. Procurement and legal join when a change touches contracts, compliance, or supplier commitments: members in companies where those functions gate the product, invited reviewers elsewhere. A secretary, typically the program or technical PM, logs requests, prepares impact analyses before the meeting, and keeps the decision record. An executive sponsor holds the escalation route and resource authority but does not run the board.

Seat	Brings	Role on the board
Product (chair)	Market and customer case	Owns the controlled document; convenes; breaks ties
Engineering	Feasibility, critical-path impact	Prices the schedule cost of each request
Operations / manufacturing	Cost, supply, ramp	Prices unit cost and production impact
Procurement / legal (as needed)	Contracts, compliance, suppliers	Joins when a change touches their domain
Secretary (program PM)	The record	Logs CRs, prepares impact analyses, keeps minutes
Executive sponsor	Resource authority	Escalation route; informed of decisions; does not chair

Table 1. CCB composition. Three to five voting seats; an odd number is better.

Process and cadence

A change becomes CCB business only when someone writes it down on a change request form. That single requirement filters most noise: ideas that survive being written down with a sponsor and a rationale are the ones worth an impact analysis. The secretary triages each request and prepares the schedule, cost, and market impacts before the meeting, so the board decides rather than researches. Each request closes with one of four outcomes: approve, reject, defer, or return for more information. The decision is recorded with the vote and the rationale [11].

Cadence follows demand. A standing monthly slot works for most programs, with two rules: if there is nothing to decide, cancel the meeting; if a request is urgent, decide it within days by an ad hoc session rather than letting it age. The board exists to move decisions fast, and whenever a best-practice team needs its CCB, it convenes the board instantly rather than waiting for the calendar [5]. Anyone who wants a change makes the pitch to the board — including executives. That is the point of the mechanism: change pays its way through the same gate regardless of the requester's rank.

Normal versus best

Normal organization	Best organization
No change control strategy, or an ad hoc one; the spec is hard-frozen until reality forces a full replan	Flexible containment: the spec is under version control and changes pass one priced gate
Changes enter through hallways, customer visits, and executive suggestions; none carries a price	Every change enters on a CR form with schedule, cost, and market impacts prepared in advance
Change volume rises as development proceeds, peaking where rework is most expensive	Change volume is front-loaded while rework is cheap, then tapers deliberately
The executive staff acts as an informal change board; the team waits to be told	The core team is the CCB and informs executives of decisions taken
Slips surface at milestones, unexplained and unowned	Every accepted slip was bought consciously against a cost-of-delay figure

Table 2. Change control in normal and best organizations, from lateralworks FTTM best-practice research [5].

A working charter

The mechanism fits on two pages. A 2026 semiconductor program lateralworks coaches chartered its board this way: the CCB governs all changes to the controlled product specification once approved; scope covers every specification parameter and the assumptions behind them; membership is the product owner as chair, the technical PM as secretary, marketing, an executive sponsor from program leadership, and two independent challengers; quorum is four including the chair or sponsor; decisions are by consensus where possible, majority otherwise, chair breaking ties; urgent requests close within five working days; and the board meets monthly or on demand, canceling when the log is empty [11]. Nothing in that paragraph requires a process department. It requires a team that owns its spec and a leadership willing to route its own ideas through the same gate. The full charter, anonymized, is reproduced in Appendix A.

04

Empowerment

The CCB is the core team

Everything to this point describes machinery. The machinery runs at the speed of its authority: a board that must check its decisions upstairs is not a board, it is a staffing function for the real board. This section is about who actually decides.

Section 04

Freedom levels and the speed of deciding

Bill Oncken's freedom scale describes five levels of initiative in any working relationship [12]. At level 1 a team acts and reports routinely; at level 2 it acts and advises at once; at level 3 it recommends, then acts. Levels 4 and 5 — wait until told, ask what to do — are training states. Experienced professionals cannot operate there, and Oncken's point was that the lower levels are not granted so much as taken: a team earns freedom by exercising it well, and a boss learns to delegate by watching it exercised well.

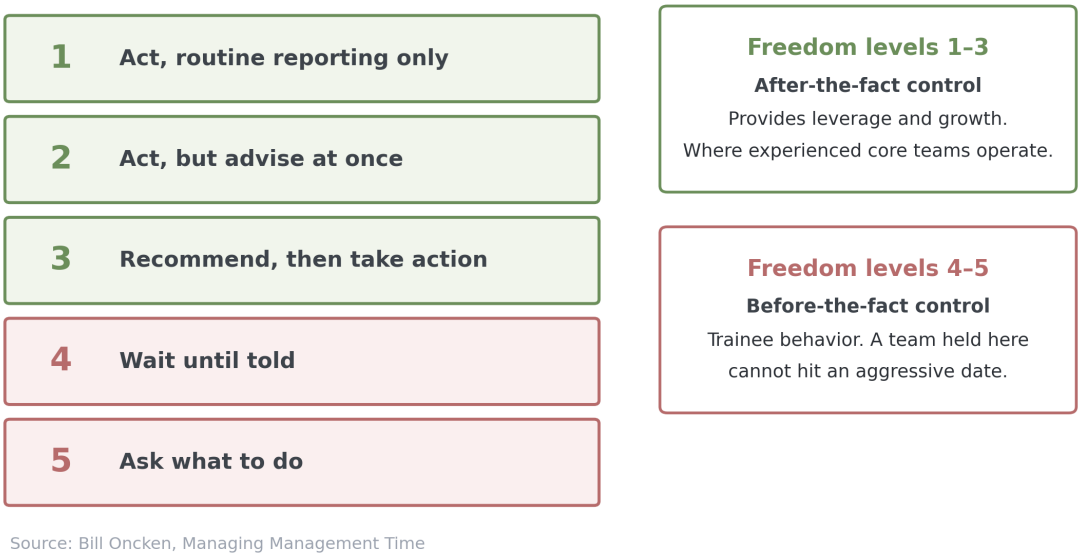

Figure 5. The Oncken freedom scale [12]. Fast core teams run their CCB at level 1 or 2: decide, then inform.

Applied to change control, the scale gives a precise answer to the question every transition stumbles on: how much can the team decide without the executives? The FTTM answer is that the core team, the small cross-functional group that owns the program [13], runs product-definition changes at freedom level 1 or 2. It decides, and it informs the executive staff of the decision, immediately for consequential changes and in routine reporting for the rest. The scope-locked spec circulates to executives for information, not for review. This is the fastest arrangement available, because decision latency, not decision quality, is what separates team-level from executive-level change control. The people closest to the spec, the schedule, and the customer already hold the information; a trip up the organization adds weeks and adds nothing else.

Glenn Parker's empowerment continuum makes the same case from team research: on fully empowered teams, final decisions rest with the team, and the leader's role shifts from checking schedules to coaching [14]. In one benchmark organization from the lateralworks client history — Tektronix in its fastest era — the change control board simply was the core team, and the war rooms were named speed rooms to make the

intent unmistakable. That is what level-1 autonomy looks like in furniture.

What executives do instead

Executives do not disappear from the picture; they move to the places where they add speed. They sit on the board only when they hold a seat — the product executive chairing, for instance. They sponsor: resources, escalation, air cover. They coach the team toward higher freedom levels instead of slapping hands when the team takes one. And they submit their own change ideas on the same form everyone else uses. An executive team that reserves spec decisions for itself has quietly become the CCB: the slowest possible one, meeting at the tempo of executive calendars with the least current information in the building. The visible symptom is a team that thinks it decided something and later learns it had no authority to; after two of those, the team stops deciding anything, and the date is gone.

Field note

Taking empowerment

You can wait for empowerment, or you can take it. If you wait to get it, you will never get it.

lateralworks

Core team coaching session, 2026

05

Evidence

Three companies that did it

The mechanism is not a lateralworks invention. Companies that took change discipline seriously have documented what it bought them. Three examples show the same pattern from three directions: an airframe, a shipping line, and an instrument company.

Section 05

Boeing, Maersk Line, Hewlett-Packard

Boeing: design the change out before it is expensive

The 777 program made configuration discipline a design principle. Every part lived in a single digital product definition, and cross-functional design-build teams resolved interferences on the screen, where a change costs hours, instead of on the factory floor, where it costs months. Boeing's stated target was half the change, error, and rework of preceding programs, and Boeing retrospectives credit the approach with far fewer assembly and systems problems than any previous airplane program [15, 16]. The lesson is the front-loading principle from Section 01 at industrial scale. The 777 moved change to the phase where it was cheap, then controlled the baseline ruthlessly.

Maersk Line: put a price on every week

Maersk Line's product development group quantified cost of delay across a development portfolio of roughly $100 million and used it to prioritize work and judge changes [8, 9]. The numbers reshaped behavior on contact. Features assumed to be equivalent differed in delay cost by orders of magnitude; the $200,000-per-week capability that had waited 38 weeks in queues became the emblem of what unpriced time was costing. Delay economics gave the organization a shared language for trade-offs that had previously been settled by seniority, and the team reported clear gains in value delivered and end-to-end speed once the metric was in routine use [9].

Hewlett-Packard: one chart, one economic language

HP institutionalized the schedule-money connection a generation earlier with the return map, published by House and Price in 1991 [17]. The map plots cumulative investment and return against time and yields break-even time, how long until the product pays back its development, as the company's standard product development metric. Its power was exactly the power a CCB needs: when someone proposed a change, the team could redraw the map and see the cost of the added weeks in the same units as the benefit. House and Price show teams redrawing the map as decisions moved it, weighing added weeks against added return — the reject branch of Figure 4, drawn by hand, decades before the vocabulary existed.

> **The common thread.** None of the three companies made change harder to request. Each made its cost visible, in a digital baseline, a delay model, or a break-even chart, and let the economics deter the changes that could not pay.

06

Implementation

Putting change control in place

A program can stand up working change control in two weeks. The sequence below is the one lateralworks coaches through, in the order the pieces depend on each other.

Section 06

Two weeks to a working gate

First, the document

Write the specification down if it is not written, and put it under revision control the day it is approved. Revision history on the front page: date, what changed, who approved. The core team owns the document. This step costs almost nothing and can happen this week; no board is needed to start it.

Second, the schedule and the model

Change control runs on two instruments: a schedule current enough to show the real critical path, and a cost-of-delay model for the program. Build the first-order model in a spreadsheet (peak revenue, market growth, price erosion, window length) and agree on the dollars-per-week figure with finance and marketing so the number carries authority when the board cites it. A weekly schedule refresh keeps the critical path honest between board meetings [9].

Third, the board

Charter the CCB on the core team: product chairs and owns the spec, engineering and operations hold seats, procurement or legal join when their domain is touched, the program PM keeps the log, and an executive sponsors without chairing. Adopt the four outcomes, the CR form, and the cancel-if-empty cadence from Section 03. Then set the freedom-level agreement explicitly with the executive staff: the team decides product-definition changes at level 1 or 2 and informs; executives coach, sponsor, and pitch their own changes through the same gate. Writing that sentence into the charter and having the executive sponsor sign it is what prevents the respin where a team discovers its decisions were provisional.

The board’s working rules compress to three lines that lateralworks best-practice research has carried for two decades: stop the unnecessary changes; sort the early commits from the late ones; implement the changes you accept, quickly [5].

Where this connects

Change control governs what enters a program that has already started. Its portfolio-level twin, starts control, governs what enters the pipeline in the first place and fails the same way when unpriced additions are absorbed silently. The two disciplines share the zero-sum logic that adding anything visibly costs something, and readers implementing one should read the companion paper on the other at lateralworks.com/papers [18]. Both rest on the same foundation: a team empowered to decide, a schedule that tells the truth, and an economic model that makes time visible in dollars. Install those three and the board meetings get short. Most changes stop making sense before anyone has to say no.

A

Appendix

A working charter, anonymized

The charter below is the working document from the semiconductor memory program referenced in Section 03, reproduced with names, the client, and product identifiers removed [11]. It is here because of what it demonstrates: the entire mechanism, from scope to cadence, fits on two pages and was drafted in an afternoon.

Appendix A

Product definition CCB charter

1. Purpose and scope

The product definition change control board (the CCB) governs all changes to the product specification (the controlled document) once it is initially approved. Its goal is that changes are adopted only after a deliberate review of the pros, the cons, and the business need driving them. The controlled document anchors the entire downstream design and qualification effort; it is treated as a frozen baseline, and the CCB is the only route by which that baseline moves. Scope covers all specification parameters — densities, configurations, interface speed, supply voltages, temperature range, reliability and endurance targets, timing parameters, and the internal test-mode set — together with the assumptions stated in the document.

2. Membership

Member	Function	CCB role and responsibility
Product owner	Product management	Chair — owns the controlled document; convenes meetings; breaks ties
Technical program manager	Program office	Secretary — maintains the CR log, prepares impact analyses, owns minutes
Marketing lead	Marketing	Member — represents the customer and market view; owns the business rationale
Executive sponsor	Program leadership	Resource authority; escalation route
Two external advisors	Consultants	Members — independent technical and process challenge

Table 3. Charter membership, anonymized.

3. Process and decision rules

A change is a CCB matter only when submitted on the change request (CR) form. The secretary triages and logs each CR; schedule, technical, market, and cost impacts are prepared before the meeting. The CCB reviews at the next standing meeting; urgent CRs are decided within five working days by an urgent session or an asynchronous round. Each CR closes with one of four outcomes — approve, reject, defer, or return for more information — recorded with rationale and vote. Quorum is four members including the chair or the executive sponsor. Decisions are by consensus where possible, otherwise by simple majority, with the chair breaking ties. The executive sponsor may escalate any decision for reconsideration.

4. Evaluation criteria

Each request is weighed against business need, customer and market impact, schedule impact (including the current critical path), cost, technical risk, reversibility, and alternatives considered. These criteria structure the discussion; they are not a checklist. The CCB's job is judgment, not arithmetic — a strong business need does not automatically carry a damaging change, and a low-impact change does not automatically clear a weak rationale.

5. Cadence, records, and effective date

The CCB meets monthly or upon demand through the first design spin, with cadence reviewed at first samples. The secretary maintains the controlled-document register, the CR log, the decision record, and the meeting minutes. The charter takes effect on approval of the initial product specification, aligned with the program's product-definition milestone on the critical path.

Sources

References

[1] Project Management Institute. *Pulse of the Profession 2018: Success in Disruptive Times.* PMI, 2018. https://www.pmi.org/learning/thought-leadership/pulse

[2] Alderton, M. “Scope Patrol.” *PM Network*, Project Management Institute, 2017. https://www.pmi.org/learning/library/scope-creep-rising-11308

[3] Reinertsen, D. G. “Whodunit? The Search for the New-Product Killers.” *Electronic Business*, July 1983. The McKinsey analysis behind the six-months-late figure.

[4] Smith, P. G., and Reinertsen, D. G. *Developing Products in Half the Time: New Rules, New Tools.* 2nd ed., Wiley, 1998.

[5] lateralworks. “FTTM New Product Development Best Practices.” Revision 2018.002. Sections on change volume, flexible containment of specifications, and CCB guidelines.

[6] Reinertsen, D. G. *The Principles of Product Development Flow: Second Generation Lean Product Development.* Celeritas Publishing, 2009.

[7] lateralworks. “Cost-of-delay.” lateralworks.com/ideas, January 2019. https://lateralworks.com/ideas/cost-of-delay-2

[8] Arnold, J. J., and Yüce, Ö. “Black Swan Farming Using Cost of Delay: Discover, Nurture and Speed Up Delivery of Value.” *Proceedings of the Agile 2013 Conference*, IEEE, 2013.

[9] Arnold, J. J. “Experience Report — Maersk Line” and “Cost of Delay.” Black Swan Farming, 2013–2015. https://blackswanfarming.com/experience-report-maersk-line/

[10] lateralworks. “Make your bed: the weekly refresh” and “The long pole.” lateralworks.com/papers, 2026.

[11] lateralworks. Internal engagement records: product-definition CCB charter, semiconductor memory program, 2026. Client identity withheld.

[12] Oncken, W. *Managing Management Time: Who’s Got the Monkey?* Prentice-Hall, 1984.

[13] lateralworks. “The FTTM core team.” lateralworks.com/papers, 2026.

[14] Parker, G. M. *Cross-Functional Teams: Working with Allies, Enemies, and Other Strangers.* Jossey-Bass, 2003.

[15] Glende, W. L. “The Boeing 777: A Look Back.” *AGARD Conference Proceedings CP-602*, NATO Advisory Group for Aerospace Research and Development, 1997.

[16] “Boeing 777: Design Philosophies — CATIA Paperless Design.” ME39C case study, University of California, Berkeley, 1997. Reports Boeing’s 50 percent rework-reduction target for digital preassembly.

[17] House, C. H., and Price, R. L. “The Return Map: Tracking Product Teams.” *Harvard Business Review*, January–February 1991, pp. 92–100. https://hbr.org/1991/01/the-return-map-tracking-product-teams

[18] lateralworks. “Starts control: balancing the portfolio for speed.” lateralworks.com/papers, 2026.