Article

What a day is worth

A portfolio cost of delay dashboard for products headed into AI data centers

FTTM best practices series.

How fast teams price a day of schedule, use the number to make tradeoff decisions, and keep the work urgent every day. Illustrated with a client portfolio; identity withheld, figures changed.

Prepared by
lateralworks
Fast Time To Market (FTTM)

Date
November 2026
Client example, illustrative figures

Online
lateralworks.com
lateralworks.com/ideas

© 2026 lateralworks

Table of contents

What a day is worth

Abstract	**03**
01 Why a day has a price	**04**
02 Five products for AI data centers	**07**
03 Tradeoffs and urgency, every day	**12**
04 Six steps to a portfolio number	**16**
References	**17**

Core thesis. Delay has a price, and the team can know it. When every program carries a current cost of delay in dollars per day, the team settles tradeoffs on economics in the meeting where they come up, shared resources go where a day is worth the most, and urgency no longer waits for the deadline.

© 2026 lateralworks

Overview

Abstract

Most product teams know their budget to the dollar and cannot say what a day of delay costs. Fast teams can. They state, in dollars of lost profit, what happens when their launch slips by one day. They use the number to settle tradeoffs, and they use it to keep the work urgent long before a deadline is in sight. We found this habit in our original research into fast product teams and have built it into client programs since [1, 2]. It is still rare. Reinertsen reports that about 85 percent of product managers do not know the cost of delay for their own projects [3].

This article shows the practice at portfolio scale. The example is a client developing five products used in AI data centers, a market with very large unit volumes and customers who do not wait. Each project team owns a cost of delay model, developed with lateralworks, and refreshes it every month as the market, the customers and the team's own fastProject schedule change. We synthesize the five models into one executive dashboard.

The client's identity is withheld. Program names and every figure shown here are changed, so nothing in the example traces to a real product. Treat the numbers as an illustration. The structure of the dashboard and the way the teams maintain it are real.

Three findings carry over to any portfolio. First, a day of delay is worth far more than the development money that usually gets argued over. In the illustration, about nine days of delay cost as much as the R&D budgets of all five programs combined as the teams modeled them, and about 35 days on a common basis. Second, programs differ widely in what a day is worth: more than 100 to 1 as modeled and about 15 to 1 on a common basis. That spread settles priority calls that otherwise go to the loudest voice. Third, the number changes behavior only when the team that owns the schedule also owns the model and refreshes both.

The complete nine-page illustrative dashboard accompanies this article [4].

© 2026 lateralworks

01

Economics

Why a day has a price

Time to market is a development period followed by a market window. The window opens when customers are ready to buy and closes when the technology stops being competitive. A team that arrives at the front of the window sells at the highest prices for the longest time. A team that arrives late sells fewer units at lower prices, and the window still closes on the same date [2].

Figure 1 shows the two models we use to explain this to teams. Late entry loses revenue across the whole sales life, because the late product never reaches the peak the on-time product would have reached. It loses profit faster still, because the first units into a new window carry the best prices and margins [2, 5].

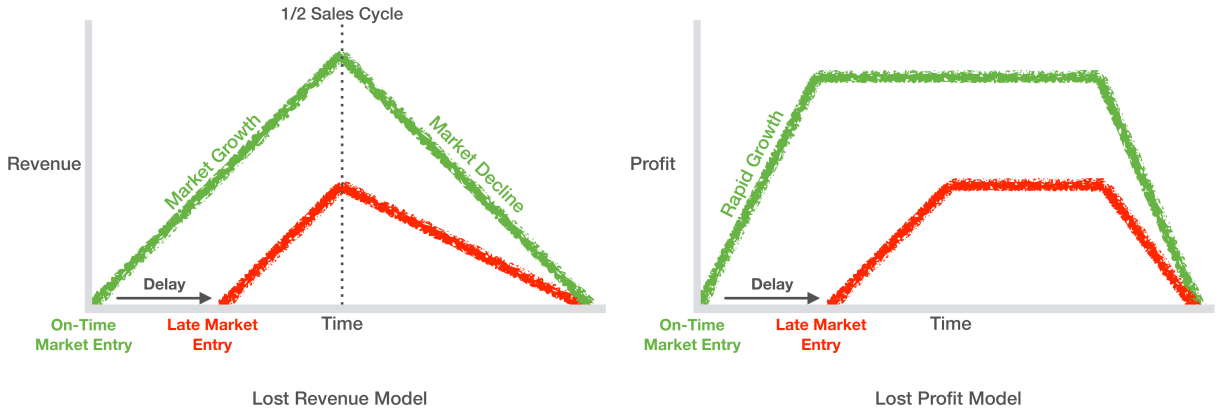


Figure 1. The lost revenue and lost profit models. The late entrant (red) peaks lower and never recovers the area between the two curves. Source: lateralworks host best practices [5].

© 2026 lateralworks

What the research found

Fast teams connect time and money

Our research into fast product teams found a consistent habit. The fast teams had calculated the economics of delay and worked on a “pay to save a day” basis. They used the number to create urgency early in the program and to speed up decisions throughout it [1, 5]. The practice is written at the level of the individual contributor. Every engineer on the team knows what a day costs [5].

The idea is not new. Neal Mitchell of lateralworks first met it at Digital Equipment Corporation in 1980, where the MicroVAX program used Don Reinertsen’s work on the economics of delay to decide which features to keep and which to defer. Decisions got faster as a result [1]. Smith and Reinertsen later gave a chapter of *Developing Products in Half the Time* to putting a price tag on time [6], and Reinertsen’s later book treats cost of delay as the one number a development organization must quantify [3].

The published evidence points the same way. Cohen, Eliashberg and Ho report a McKinsey study in which shipping six months late cost a product 33 percent of its after-tax profit, while overspending the development budget by 50 percent cost 3.5 percent [7]. House and Price built Hewlett-Packard’s return map on the same logic and tracked product teams by time to market and break-even time [8]. Arnold and Yüce describe how Maersk Line applied cost of delay across a $100 million portfolio to decide what to build first [9]. On most of the programs we work with, the cost of delay runs between $1 million and $2 million of lost profit a day [10].

Why most organizations never see the number

In a normal organization no one owns the cost of delay. Finance owns the budget and engineering owns the schedule, so time and money are managed by two functions that seldom meet over the same decision [5]. The budget is visible to the dollar. The cost of being late is invisible, so it loses every argument. A team is refused one engineer who would save 90 days because the headcount was not in the annual budget. No one is accountable for the profit that refusal gave away [5]. Engineering assumes that marketing or finance must know what lateness costs. Usually neither does [1].

Normal organization	Best organization
No one owns the cost of being late. Engineering manages time and finance manages money.	Each team owns a cost of delay model and can state its figure per day.
Tradeoffs are argued on opinion and settled by seniority, often weeks later.	Tradeoffs are priced in the model and settled in the meeting where they come up.
Spending to go faster needs an exception to the annual budget.	The team pays to save a day whenever the day is worth more than the spend.
Urgency arrives with the deadline.	Urgency is constant, because every day of the program carries the same price.

© 2026 lateralworks

Economics of delay

The one number

If you only quantify one thing, quantify the cost of delay.

Donald G. Reinertsen

The Principles of Product Development Flow, 2009

© 2026 lateralworks

02

Client example

Five products for AI data centers

The client develops products used in AI data centers. The largest buyers of computing in the world set the demand. The five largest hyperscalers (the companies that build and run the biggest cloud and AI data centers) spent about $383 billion on capital expenditures in 2025. Estimates made in May 2026 put their 2026 spending at about $768 billion [11]. The equipment they buy turns over quickly. By mid-2024 Nvidia and AMD had both committed to a new data center accelerator every year instead of every two [12].

For a supplier whose products go into that equipment, each generation is a new market window. Arriving on time earns a place in the design for the life of that generation. Arriving late leaves whatever share is still open. This is the right-time problem in its sharpest form: the financial return comes from being in the market as demand starts to climb [13]. In the illustration, unit volumes run from tens of millions to hundreds of millions per program, so a late day is expensive on every one of them.

© 2026 lateralworks

How it is built

One model per team, refreshed monthly

Each of the five project teams owns a cost of delay model, developed with lateralworks in the Cost-of-delayAI worksheet [14]. The model is a compact business case: unit sales, average selling price and cost of goods by year, the R&D budget, and selling and administrative cost (SG&A). A second case applies the delay. The difference in net profit between the two cases, divided by the months of delay, is the cost of delay per month. Dividing by 30 gives the cost per day [4].

The teams refresh their models every month. Three kinds of news drive the changes:

- **The market.** Volume forecasts, price erosion and the timing of the window move as the data center buildout moves.
- **The customers.** A design win or a moved qualification date alters the units a program can expect and what a late arrival would forfeit.
- **The schedule.** Each team runs its own fastProject schedule, a critical-path schedule the team refreshes every week [15]. The current forecast against the target date tells the model how much delay is at stake.

The people who refresh the schedule are the people who refresh the model. When the weekly refresh shows days lost on the critical path (the chain of tasks that sets the finish date), the team can price those days the same week. Figure 2 shows the loop.

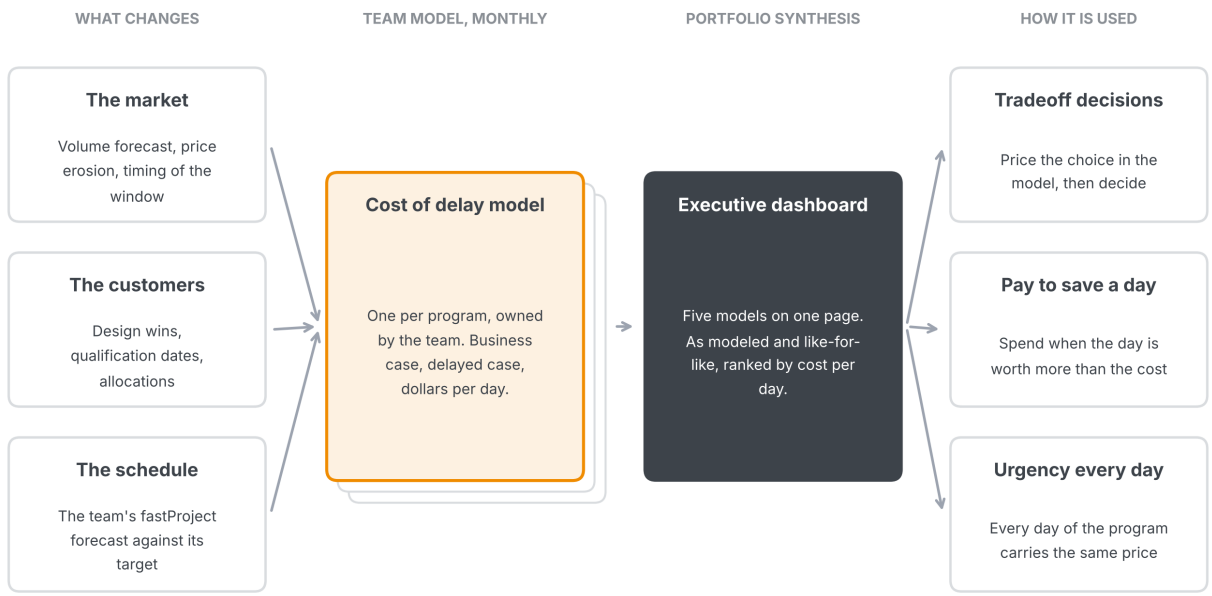


Figure 2. The monthly refresh. Market, customer and schedule news update each team's model; the models roll up to one dashboard; the dashboard feeds decisions.

© 2026 lateralworks

Reading the dashboard

What the portfolio page says

We read the five team workbooks and synthesize them into one executive dashboard: a portfolio page, a roll-up of every program on one calendar, and a page for each program. No figure on it is typed by hand. Each one is read from a team's model and recomputed from the model's inputs before it prints [4]. Figure 3 is the portfolio page from the illustrative version, and four readings stand out.

A day is worth $8.5 million

Across the five programs as the teams modeled them, one day of delay costs $8.5 million in net profit and one month costs $255 million. The R&D budgets of all five programs total $78.1 million. About nine days of delay cost as much as all the development money in the portfolio. On a common basis, described below, the same budget buys about 35 days.

One program carries most of it

Program A accounts for 85 percent of the daily figure, and its model prices delay differently from the others. The other four use a sales shift: each month of delay moves unit sales a month later, and units pushed past the end of the window are lost. In the illustration, Program A's model assumes a three-month slip misses the customer's qualification window, the period in which the customer approves parts for a new design, and forfeits 60 percent of unit sales. Both models are legitimate. They describe different markets.

Rank on one basis

A ranking built on mixed assumptions ranks the assumptions. So the dashboard also applies one delay model, a three-month sales shift, to every business case. On that like-for-like basis Program A costs $962,000 a day instead of $7.2 million, and the portfolio costs $2.3 million a day. The team's own figure stays the headline because the team owns it. The like-for-like figure sits beside it so an executive can see how much of the ranking comes from the market and how much from the modeling choice.

The spread is wide

Program E costs $66,000 a day. On the common basis a day on Program A is worth about 15 times a day on Program E, and as modeled more than 100 times. When both programs need the same test engineer or the same lab slot, the dashboard says which one goes first.

© 2026 lateralworks

AI DATA CENTER PORTFOLIO · COST OF DELAY · ILLUSTRATIVE EXAMPLE

Cost of delay: executive dashboard

What a day and a month of schedule delay cost in net profit, program by program, from the AI data center portfolio cost of delay models.

Models as of Nov 18, 2026

5 workbooks read, 5 with a cost of delay

$8.5M
Cost of delay per day, as modeled
5 programs; Program A is 85% of it

$2.3M
Cost of delay per day, like-for-like
3-month sales shift on all 5 business cases

$7.24B
Revenue in the business cases
net profit $2.37B; R&D $78.1M

5 of 5
Programs with a cost of delay in the workbook
every model carries a delay

Cost of delay by program

Ranked by cost per day as stored in each workbook. Like-for-like bars apply one 3-month sales shift to every business case. Net profit and ROI: business case, then delayed case.

Program and delay model	Per day	Per month	Per day, to scale	Net profit and ROI
Program A -60% unit sales, 3-month delay	$7.2M	$216M	$7.2M / $962K	$1.06B to $410M ROI 50x to 20x
Program B Sales shift, 4-month delay	$782K	$23.5M	$782K / $782K	$752M to $658M ROI 29x to 25x
Program C Sales shift, 3-month delay	$376K	$11.3M	$376K / $376K	$351M to $317M ROI 20x to 16x
Program D Sales shift, 3-month delay	$75K	$2.2M	$75K / $75K	$98.4M to $91.7M ROI 14x to 13x
Program E Sales shift, 3-month delay	$66K	$2.0M	$66K / $66K	$114M to $108M ROI 16x to 17x
Portfolio, as modeled	$8.5M	$255M	as modeled in the workbook / like-for-like: 3-month sales shift. A broken bar runs off the scale; its label carries the value.	

What the models say

1. A day of delay across the 5 modeled programs costs $8.5M in net profit. Program A carries 85% of it.
2. Program A is priced on a different delay model (-60% unit sales, 3-month delay). On the same 3-month sales shift as the others it costs $962K a day, not $7.2M.
3. One month of delay on Program B costs $23.5M, 0.9 times the $26.0M R&D budget entered in the model. About 33 days of delay cost as much as that budget.
4. Illustrative example. Client identity withheld; program names and all figures are changed.

How to read: cost of delay is the net profit lost when the launch slips, divided by the months of delay; per day is per month ÷ 30. When two programs need the same resource, the higher cost per day goes first. Method and data notes follow the program pages.

lateralworks

Sources [1] to [5]: AI data center portfolio cost of delay workbooks, saved Nov 18, 2026. Figures are the workbooks' own unless marked like-for-like.

Page 1 of 9

© 2026 lateralworks

Figure 3. Portfolio page of the cost of delay dashboard, rotated. Illustrative figures; client identity withheld [4].

© 2026 lateralworks

The roll-up

Every program on one calendar

Figure 4 puts the five programs on one calendar. The business cases add up to $2.37 billion in net profit through 2032. The delays as the teams modeled them, three months on four programs and four months on one, would take $787 million of it. A three-month slip on every program, priced as a sales shift, would take $203 million. The gap at the right-hand end of the cumulative chart is the portfolio's cost of delay. In these models it does not close, because each window ends on the same date whether the product arrived on time or late.

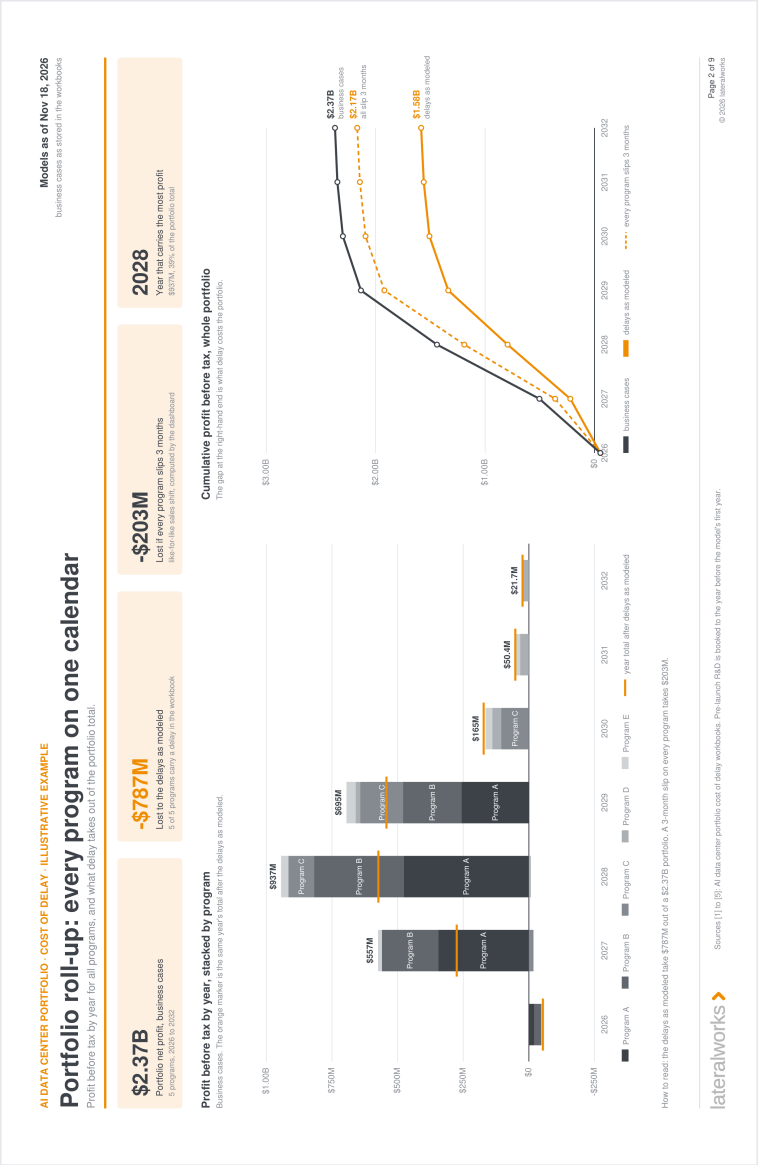


Figure 4. Portfolio roll-up, rotated. Profit before tax by year, and the cumulative gap that delay opens. Illustrative figures [4].

© 2026 lateralworks

03

Practice

Tradeoffs and urgency, every day

A cost of delay that appears only on an executive dashboard is a statistic. The teams in this portfolio put theirs to work in two ways. They settle tradeoff decisions on economics, and they treat every day as urgent.

Both uses depend on the figure being current and being the team’s own. A number handed down from finance once a year gets quoted in reviews. A number the team rebuilt last month, from its own schedule and its own customer conversations, gets used.

© 2026 lateralworks

Tradeoff decisions

Price the choice, then decide

Every program page carries a sensitivity chart: what a 1 percent change in each input does to net profit, set against one month of schedule (Figure 5). For Program B the bars read as follows.

- One month of schedule: $23.5 million
- A 1 percent drop in selling price: $21.4 million
- A 1 percent rise in cost of goods: $13.6 million
- A 1 percent drop in unit sales: $7.8 million
- A 1 percent overrun in R&D: $260,000

A month of delay costs Program B more than a 1 percent price cut on every unit it will ever sell, and 90 times what a 1 percent budget overrun costs. The example model in our best practices shows the same pattern: schedule is the most sensitive factor in the profit projection [5]. It does not hold for every program. On Program E a 1 percent price change outweighs a month of schedule, which is why each team needs its own model.

The bars turn a debate into arithmetic. Take the kind of decision our trade-off model was built for [1]. Midway through Program B a customer asks for added functionality. Marketing believes it supports a 3 percent higher price. Engineering estimates 2 percent more cost of goods, $2 million more R&D and six weeks of schedule. Read off the bars, the price is worth $64 million. The cost of goods takes back $27 million, the R&D $2 million and the six weeks $35 million. A feature that looked like a $35 million gain nets to about zero once the delay is priced. The useful question becomes whether the team can deliver it without the six weeks, and what that would be worth paying for. These are first-order readings; the team reruns the model to confirm before it commits.

Pay to save a day

The same number sets the price a team should pay for speed. A day on Program B is worth $782,000. An expedite fee of $250,000 that saves five days returns $3.9 million. In an organization run on annual functional budgets that request stalls, because the fee is visible and the five days are not [5]. With the model in hand the approval takes one conversation. The best organizations make the offer to pay for a day explicit, and the offer changes the mindset more than the money does [5]. We treat a day pulled in the same way: it is worth about what a day lost costs.

Shared resources go where a day is worth most

Programs in a portfolio compete for the same specialists and the same lab time. The dashboard states the rule on its first page: when two programs need the same resource, the higher cost per day goes first. Reinertsen refines the rule for jobs of different lengths by dividing cost of delay by duration [3].

© 2026 lateralworks

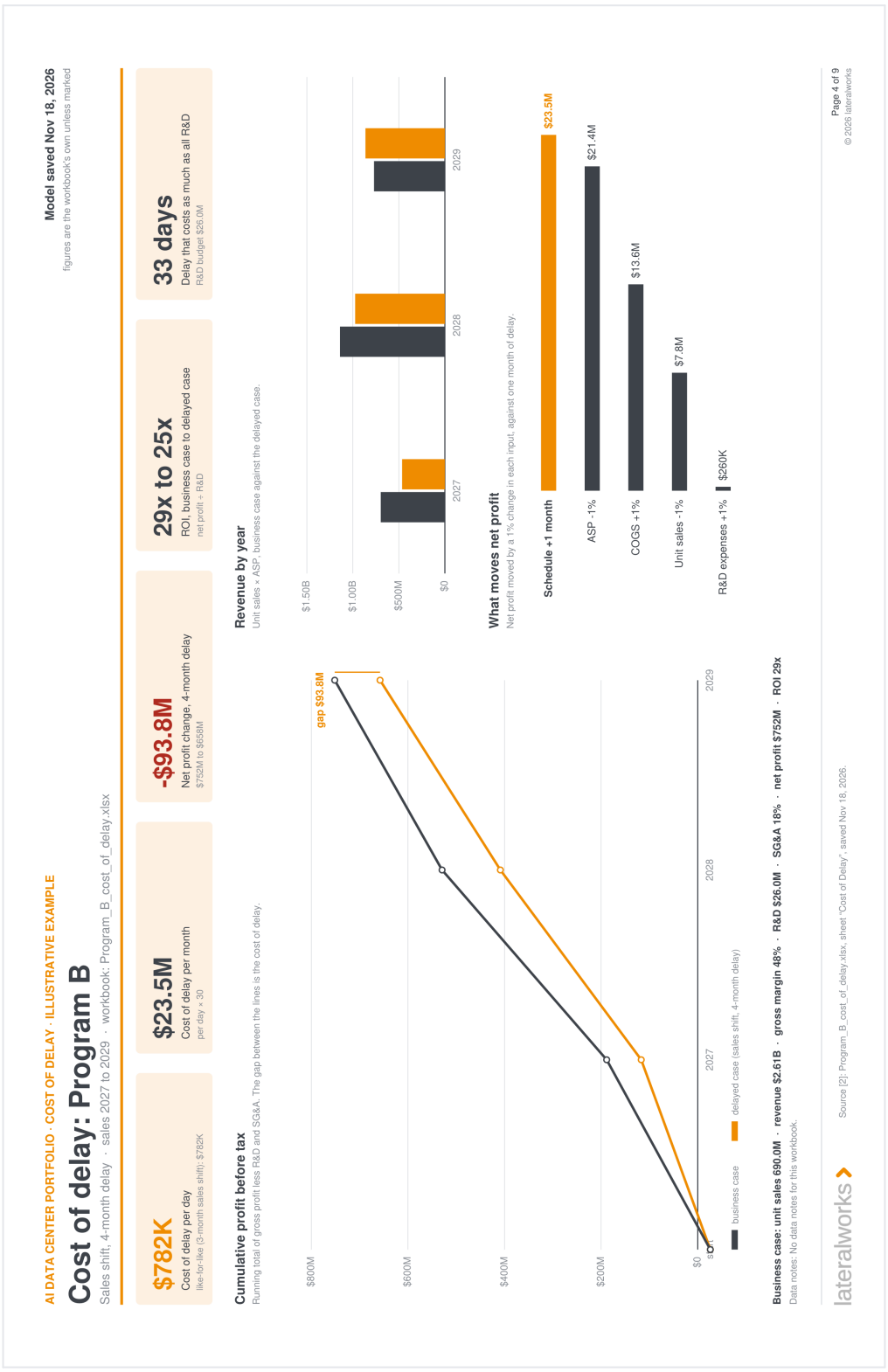


Figure 5. Program page for Program B, rotated: headline tiles, cumulative profit for the business case and the delayed case, revenue by year and the sensitivity bars. Illustrative figures [4].

© 2026 lateralworks

Urgency

Every day carries the same price

Deadlines create urgency late. A team eighteen months from its target feels little pressure in month two. A day lost on the critical path in month two still moves the launch as far as a day lost in the final week. The cost of delay makes that visible. Program A loses an average of $7.2 million for every day it slips, about $300,000 an hour, and the price is the same in the first month of the program as in the last. Three days of delay cost Program A as much as its entire $21 million R&D budget. A decision on Program B that waits a week for the next meeting costs $5.5 million.

This is why the practice is written at the level of the individual contributor [5]. An engineer who knows the figure treats a two-day wait for a license or a review differently, and so does the manager who can remove the wait. The fastest teams go a step further and pull in before they slip: they accelerate work early to bank time, because the schedule will need it later [16]. The weekly schedule refresh supplies the days and the model supplies the dollars [15].

What the number is not

A cost of delay is an estimate built on a forecast. The five models here use different delay assumptions, and a figure can move several-fold when a team changes its view of the customer’s window. The per-day figure is also an average over the delay the team modeled. Under a sales shift each day costs about the same. Where the model assumes a cliff, as Program A’s does, the days nearest the cliff cost more than the average, which argues for more urgency early.

The answer is to refresh the model, print the assumption next to the figure and compare programs on a common basis. The dashboard does all three. Judgment without a model does far worse. Reinertsen reports that intuitive estimates of cost of delay can differ by 50 to 1, even among members of the same team [17].

© 2026 lateralworks

04

Getting started

Six steps to a portfolio number

Start with one program. A first model is small: one business case and one delayed case. The portfolio page comes from repeating the exercise on each program and holding the teams to a monthly refresh. Six steps cover it.

1. **Build one model per program, with the team.** Put the product boss (the single leader accountable for the product), marketing and engineering in the same session so everyone works through the logic of the market window. If lateness cannot be quantified, the target date may not be the real need date [10].
2. **Choose the delay model on purpose.** A sales shift suits a market that will still buy late. A percentage loss suits a design win that can be missed. Write the choice on the page.
3. **Tie the model to a live schedule.** The weekly forecast against target supplies the days. Without a current schedule the model prices a guess.
4. **Refresh monthly.** Record what changed in the market, with the customers and in the schedule. Let the figure move.
5. **Roll up to one page.** Show each team's own figure and a like-for-like figure, ranked by cost per day. The executive who chairs the portfolio review owns the page.
6. **Use it.** Bring the figure to every tradeoff decision and every resource conflict, and post the per-day number where the team works.

The schedule tells a team how many days are at stake. The model tells the team what those days are worth. A team that knows both does not wait for the deadline to feel urgent.

For more on the method, see the cost of delay notes and the Cost-of-delayAI overview at lateralworks.com [1, 2, 14].

© 2026 lateralworks

Sources

References

1 Mitchell, N. "Cost-of-Delay (1)." Cost-of-delayAI concept note. lateralworks, January 24, 2019.
https://lateralworks.com/docs/cost-of-delay-1
Accessed November 18, 2026.

2 Mitchell, N. "Cost-of-Delay." lateralworks Ideas, January 24, 2019.
https://lateralworks.com/ideas/cost-of-delay-2
Accessed November 18, 2026.

3 Reinertsen, D. G. *The Principles of Product Development Flow: Second Generation Lean Product Development.* Celeritas Publishing, 2009. The 85 percent finding is also reported in "Getting flow into your product development," Lean Magazine, April 11, 2011.
https://leanmagazine.net/lean/getting-flow-into-your-product-development/
Accessed November 18, 2026.

4 lateralworks. Client program records and cost of delay executive dashboard, 2026. Client identity withheld; program names and all figures shown are changed. Companion file: Cost of delay dashboard - generic example.pdf.

5 lateralworks. *FTTM New Product Development Best Practices,* Revision 2018.002, practices 1.6 and 3.2; and the host best practice slides, lost revenue and lost profit models. Supplied files: fttm best practices_rev 2018_002.pdf; host best practice slides_summaries.pdf.

6 Smith, P. G., and Reinertsen, D. G. *Developing Products in Half the Time: New Rules, New Tools,* 2nd ed. John Wiley & Sons, 1998. Chapter 2, "Putting a Price Tag on Time."
https://www.wiley-vch.de/en/areas-interest/finance-economics-law/developing-products-in-half-the-time-978-0-471-29252-4
Accessed November 18, 2026.

7 Cohen, M. A., Eliashberg, J., and Ho, T.-H. "New Product Development: The Performance and Time-to-Market Tradeoff." *Management Science* 42(2), February 1996, pp. 173-186.
https://faculty.wharton.upenn.edu/wp-content/uploads/2013/04/New-Product-Development_1.pdf
Accessed November 18, 2026.

8 House, C. H., and Price, R. L. "The Return Map: Tracking Product Teams." *Harvard Business Review,* January-February 1991, pp. 92-101. Summarized by the University of Cambridge Institute for Manufacturing.
https://www.ifm.eng.cam.ac.uk/research/dstools/hewlett-packard-return-map/
Accessed November 18, 2026.

9 Arnold, J. J., and Yüce, Ö. "Black Swan Farming Using Cost of Delay: Discover, Nurture and Speed Up Delivery of Value." *Agile Conference (AGILE) 2013,* IEEE, pp. 101-116.
https://blackswanfarming.com/black-swan-farming-using-cost-of-delay
Accessed November 18, 2026.

10 Mitchell, N. "Targets and Trends." lateralworks Ideas, January 16, 2019.
https://lateralworks.com/ideas/targets-and-trends
Accessed November 18, 2026.

11 SB Energy, Inc. Form S-1 registration statement. U.S. Securities and Exchange Commission, filed August 31, 2026. Hyperscaler capital expenditure estimates as of May 2026.
https://www.sec.gov/Archives/edgar/data/0002133037/000162828026059639/sbenergy-sx1.htm
Accessed November 18, 2026.

© 2026 lateralworks

12 Martin, D. "Analysis: As Nvidia Takes AI Victory Lap, AMD Doubles The Trouble For Intel." CRN, June 17, 2024.
https://www.crnasia.com/news/analysis-as-nvidia-takes-ai-victory-lap-amd-doubles-the-trouble-for-intel
Accessed November 18, 2026.

13 Mitchell, N. "Right-time ensures financial return from new products." lateralworks Ideas, June 20, 2019.
https://lateralworks.com/ideas/right-time-ensures-financial-return-from-new-products
Accessed November 18, 2026.

14 lateralworks. "Cost-of-delayAI." Software overview, fastProjectAI suite.
https://lateralworks.com/software
Accessed November 18, 2026.

15 Mitchell, N. "The weekly schedule refresh." lateralworks Ideas, May 8, 2026.
https://lateralworks.com/ideas/the-weekly-schedule-refresh
Accessed November 18, 2026.

16 Mitchell, N. "Pull-in before you slip." lateralworks Ideas, June 22, 2019.
https://lateralworks.com/ideas/pull-in-before-you-slip
Accessed November 18, 2026.

17 Sixtensson, A. "Cost of Delay – interview with Don Reinertsen." Lean Magazine, February 20, 2012.
https://leanmagazine.net/lean/cost-of-delay-don-reinertsen/
Accessed November 18, 2026.

© 2026 lateralworks