Whitepaper

fastProjectAI Project Portfolio Dashboard

Aggregating live project schedules into a single leadership view

FTTM toolkit series.

How leadership teams running multiple new-product programs can replace status-report theater with live, exception-based portfolio visibility — drawn from data their PMs already maintain as part of FTTM planning.

Prepared by

lateralworks
FTTM toolkit

Date

April 2026
Synthesis paper

Online

lateralworks.com
Portfolio dashboard series

© 2026 lateralworks

Table of contents

fastProjectAI Project Portfolio Dashboard

Core thesis. Most leadership teams running multiple programs cannot see their portfolio in time to act. fastProjectAI fixes this by aggregating live wigglechart data the PMs already maintain — surfacing the small number of projects that need attention this week, not the long tail that does not.

© 2026 lateralworks

Overview

Abstract

Multi-project portfolio reviews fail in predictable ways. Status reports arrive weeks after the data was collected, get filtered through layers of optimism on the way up, and surface problems only after they have grown beyond easy recovery. Each project reports in its own format, against its own baselines, with its own definition of "on track." By the time the leadership team meets, the data is stale, distorted, and impossible to compare across the portfolio.

fastProjectAI replaces this pattern. The dashboard automatically aggregates wigglecharts — a standard FTTM artifact PMs maintain as part of normal weekly planning — from every project in the portfolio. It computes a common gap-to-target metric and a six-week trend rate for every milestone, ranks them by risk, and presents the result through a small set of one-click filters.

The economics matter. Because the data already exists in the PMs' live schedules, the dashboard adds zero data-entry overhead. The aggregator reads the same files the PMs are already maintaining for their own planning. This is why the data stays honest: there is no separate dashboard to fluff up before the executive review.

The dashboard alone replaces meeting-by-meeting status theater with a working list of projects that genuinely need leadership attention this week. A second layer goes further: an LLM (in this paper, Claude) reads the dashboard each week and surfaces cross-portfolio patterns no single project view can reveal — concentration of failure modes, structural under-triage, recovery-trend asymmetries, and lessons not transferring between programs.

The paper walks through the architecture, the filtering model, the free-data argument, a worked example from a 196-milestone site deployment portfolio, and the LLM analysis layer. It closes with practical guidance for adopting the dashboard inside an organization already running on FTTM-style planning.

© 2026 lateralworks

01

The problem

Why portfolio status reports fail

Every leadership team running multiple new product development programs at once eventually arrives at the same observation: the portfolio reviews aren't telling them what they need to know in time to act. The numbers look reasonable on the slide, the projects all report progress, the executive sponsor nods, and three months later the program that was "on track" misses its launch by a quarter. This is a problem in how status flows through the layers of an organization, and in what happens to information at each layer.

Reinertsen has documented the cost-of-delay framework for product development at length [1]: the financial cost of late projects is almost always far larger than the cost of accelerating them. Yet the gap between what leadership knows and what is actually happening on the ground is the rate-limiting step on every accelerator they could apply. Visibility precedes action.

© 2026 lateralworks

Four predictable failure modes

How portfolio status gets distorted

The same four failure modes appear across organization after organization in lateralworks engagement data, regardless of industry or program type. They are structural.

Information lag.

A typical monthly status review uses data that was collected two to four weeks before the meeting. Project status snapshots get rolled up the chain, formatted into slides, scheduled into the leadership calendar, and presented well after the underlying situation has changed. By the time leadership sees that a milestone slipped a week, the project has been slipping for a month.

Aggregation distortion.

Each handoff between layers adds a layer of optimistic rounding. Engineers report "almost done" to PMs; PMs report "minor risk" to program leads; program leads report "on track with watch items" to executives. By the top of the chain, the variance has been compressed out of the data. The phenomenon is well-documented: status reports tend toward green even as the underlying schedules turn red — the so-called "watermelon report" pattern observed across government and private-sector program management studies [2, 3]. The mechanism is the same one Lee, Padmanabhan, and Whang documented in supply chains as the bullwhip effect [19]: each layer of aggregation amplifies the variance it sees while smoothing what it reports. Kahneman has shown the same optimistic-rounding bias operates at the individual-decision level [20]; what the watermelon report adds is the organizational amplifier.

No common metric.

Each project reports in its own format. One uses percent complete; another uses task burn-down; another tracks milestone slippage in days; another reports critical path float. None of them is wrong, but none of them is comparable across the portfolio. Leadership cannot ask "which of my fifteen programs is in worst shape this week" because there is no shared answer to "in worst shape compared to what." This is the comparability problem identified in Cooper, Edgett, and Kleinschmidt's work on portfolio management for new products [4]: without a common dimension, portfolios cannot be managed as portfolios.

Recovery fiction.

A point-in-time status reading does not distinguish a project that is two weeks late and recovering from one that is two weeks late and getting worse. Both look the same in the cell of a spreadsheet. Without a trend signal, leadership cannot tell which projects need intervention now and which are already self-correcting. The data shape that matters — the slope, not just the level — is invisible. Goldratt argued the same point in *Critical Chain* [25]: a status field that ignores the rate of change is structurally misleading. Pinto and Mantel's study of project failure causes reaches a related conclusion — the ability to detect derailment early is the strongest single predictor of recovery success [23].

© 2026 lateralworks

What this means. Status theater is the predictable consequence of these four failure modes acting together. The cure is to remove the layers between leadership and the live project-level data, and to express that data in a single common metric with a visible trend. Better-formatted slides will not get there.

© 2026 lateralworks

02

The solution

How fastProjectAI works

fastProjectAI takes a different approach. Rather than introducing a separate reporting system, it reads from the planning artifacts PMs already maintain in the course of their normal FTTM work — specifically, the wigglechart at the milestone level — and aggregates them into a single live portfolio view. This section walks through the underlying data structure, the aggregation logic, and what the resulting dashboard looks like in practice.

© 2026 lateralworks

The underlying artifact

What a wigglechart captures

Every project milestone in an FTTM-managed program has a *wigglechart*: a week-by-week plot of the milestone's forecast finish date against its committed target. Each week, the PM updates the forecast based on the latest schedule data; a new point is added; the line moves. This single chart captures three things a status table cannot:

- The **committed target** — the date leadership and the team agreed the milestone would land.
- The **current gap** — how far the forecast has drifted from that target, in days.
- The **trend** — whether that gap is widening, holding, or closing, and at what rate.

The wigglechart is not a lateralworks invention. The concept of plotting forecast schedule against committed schedule over time goes back at least to Smith and Reinertsen's work on fast product development cycles in the 1990s [1, 5]. Wheelwright and Clark argued in *Revolutionizing Product Development* that visible time-based metrics enable speed [16]; House and Price's "Return Map" framing pushed the same idea into the executive review by plotting committed dates against actuals on a single figure [17]. What FTTM adds is the discipline of maintaining one for every milestone, every week, as a non-negotiable PM practice. This discipline is what makes the aggregation possible.

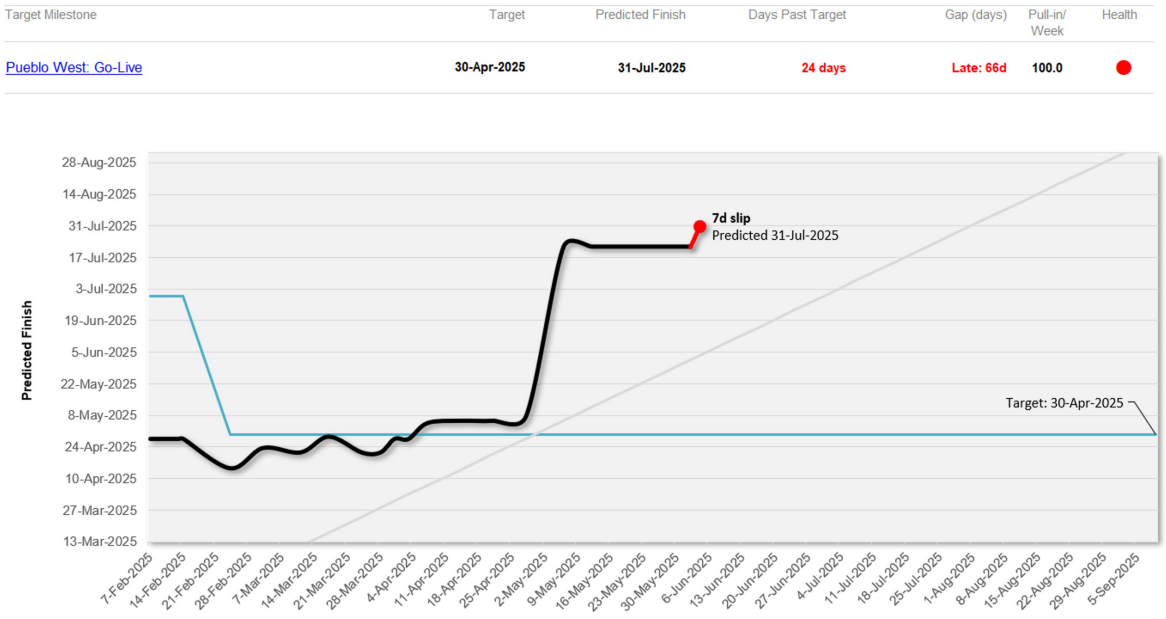

Figure 1. A real wigglechart from the worked-example portfolio: Pueblo West Go-Live, target 30-Apr-2025. The blue line is the committed target date. The black line is the weekly forecast date. The gray diagonal is the today line — where forecast meets today, time has run out. The vertical distance between the black and blue lines is the gap; the slope of the recent black line is the trend. Read together, gap and trend tell leadership which projects need attention this week.

The wigglechart makes Section 01's failure modes visible at the project level. Information lag collapses to one week (the update cadence). Aggregation distortion is removed because the data is the live forecast. The common metric is gap-in-days. Recovery fiction is impossible because the trend is plotted directly.

© 2026 lateralworks

From single charts

to portfolio aggregation

fastProjectAI takes the next step. If every project in a portfolio maintains wigglecharts at the milestone level, then a single piece of software can read all of those wigglecharts, extract the gap and trend for every milestone, and render the result as a portfolio dashboard. The aggregator is structurally simple: it ingests, normalizes, ranks, and presents.

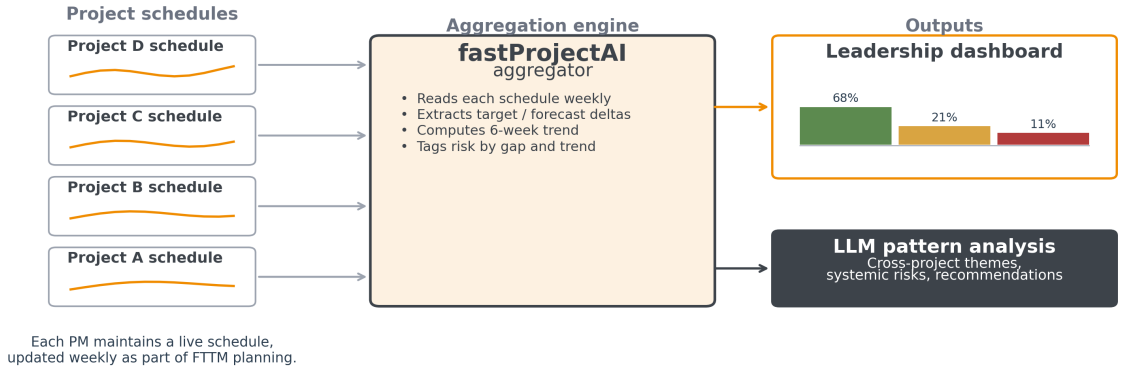

Figure 2. Data flow. Each PM maintains a live schedule as part of their normal FTTM work. The aggregator reads each schedule weekly, extracts the per-milestone target/forecast deltas, computes the six-week trend, and tags risk. Outputs feed both the leadership dashboard and the LLM pattern-analysis layer.

What the leadership team sees is a single screen: a count of milestones, distributions of risk, a sortable table of every active milestone with its gap, its trend, and its risk flag. The screen is live. The data behind it was generated by the PMs in the course of their planning work this week. There is no separate filing or reporting step.

© 2026 lateralworks

What leadership sees

The dashboard at a glance

Figure 3 shows the dashboard as it appears to leadership in a recent site-deployment portfolio. The top of the screen summarizes the health of 111 active milestones across 15 programs: 68% are projected to land within two weeks of their target, 21% within four weeks, and 11% are more than four weeks behind. The completed milestones panel on the right shows the historical record — 79% of completed milestones landed within two weeks of their committed date, suggesting the planning discipline is working in aggregate even as the active set carries pockets of trouble.

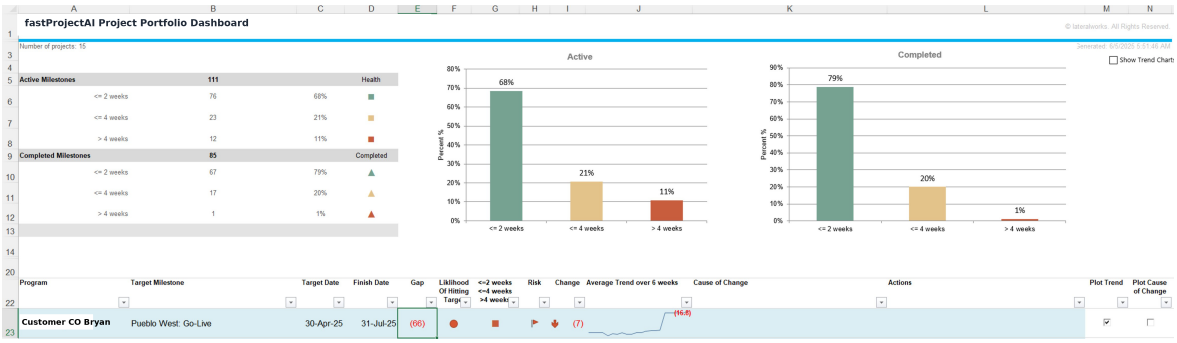

Figure 3. The dashboard's summary panel. Active milestones (left) and completed milestones (right) are bucketed by gap-to-target. A single screen lets leadership read both the current state and the planning track record at the same time.

Below the summary, the dashboard exposes a per-milestone detail table with the columns leadership actually needs: program, target milestone, target date, finish date, gap, likelihood of hitting target, risk flag, week-over-week change, six-week trend, cause of change, and current actions with named owners. Each row is one milestone, one wigglechart, one project's honest current state.

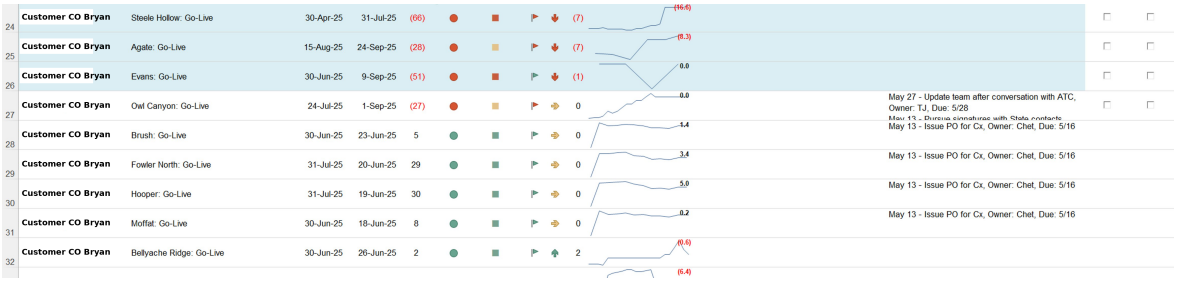

Figure 4. Per-milestone detail. Each row is a single milestone with its target date, current forecast, gap, trend rate, and the action items the PM has logged this week. The mini-charts in the "average trend over 6 weeks" column are sparkline wigglecharts — the whole shape, not just the latest number.

© 2026 lateralworks

03

Method

Reporting by exception

Aggregating the data is the first half of the problem. The second half is filtering it down to the small set of projects that actually need leadership attention this week. The principle is old — Drucker called it management by exception more than fifty years ago [6] — but its execution requires the right primitives. fastProjectAI provides them as one-click filters on the dashboard, each of which compresses the working set further.

© 2026 lateralworks

The filtering model

From 196 milestones to 10

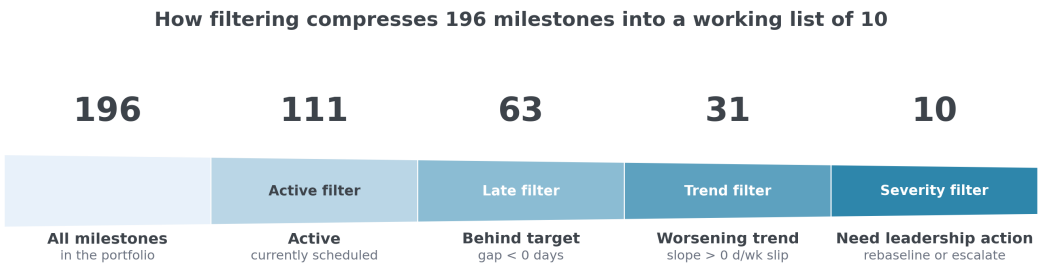

Figure 5. The filtering funnel. Each stage corresponds to a single filter button on the dashboard. Leadership reviews the right-most cohort — the projects that genuinely need attention this week — not the full 196.

In the worked example covered later in the paper, the portfolio contains 196 milestones. Of those, 111 are currently active. Of the active set, 63 are behind their target. Of the late ones, 31 are getting worse rather than recovering. Of the worsening ones, 10 are severely behind — more than four weeks — and need immediate intervention. That final cohort of 10 is what leadership should be reviewing, debating, and acting on. The other 186 either do not need attention this week or are already being managed at the project level.

Why each filter matters.

- The **active filter** removes finished and not-yet-started milestones from the working view. There is no point reviewing what cannot be influenced.
- The **late filter** removes the on-track majority. Time spent reviewing on-track projects is time not spent on the ones that need help.
- The **trend filter** removes recovering projects from the late set. A project that is late but pulling back toward target does not need leadership intervention; it needs to be left alone.
- The **severity filter** separates the projects that can still be recovered with a course correction from the runaways that need rebaselining or escalation.

The filters compose. Each one is a single click, and the dashboard updates the visible row set in place. There is no separate filing, no waiting for a refreshed report. Leadership can apply the filters live during the review and immediately see who is in trouble and who is not. The principle is older than the dashboard: Ohno made visual management the load-bearing element of the Toyota Production System on the same logic [24] — surface the abnormal, not the normal.

© 2026 lateralworks

The leader's control surface

One ribbon, seven filter groups

The four-stage funnel describes the discipline. The ribbon below is how leaders apply it. Every filter is a single button on a custom dashboard ribbon — no formulas, no pivot-table reconfiguration, no waiting for an analyst. A leader running a portfolio review opens the workbook, clicks two or three buttons, and the visible row set collapses to the projects that need attention this week.

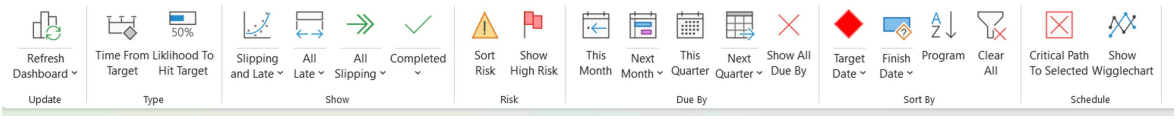

Figure 6. The fastProjectAI dashboard ribbon. Each group is one dimension of triage; combinations of buttons across groups produce the working set. Filters compose live; the row set updates in place.

What each group does.

- **Update.** *Refresh Dashboard* pulls the latest weekly forecasts from every connected project schedule. This is the only data-entry step in the entire review process — one click, before the meeting starts.
- **Type.** *Time From Target* sorts by absolute gap (how late, in days). *Likelihood To Hit Target* sorts by the probability the project still lands on date. The two views answer different questions: how bad is the slippage, and how recoverable is it.
- **Show.** The trend filters — *Slipping and Late*, *All Late*, *All Slipping*, *Completed* — are the four-stage funnel made tactile. *Slipping and Late* is the leadership-action working set; *All Late* shows everything behind regardless of trend; *All Slipping* shows everything with a worsening trend regardless of current gap; *Completed* turns the dashboard into a planning-track-record review.
- **Risk.** *Sort Risk* orders the visible set by the PM's own risk classification. *Show High Risk* filters to Risk-3 milestones only — the projects the program team has already flagged as needing escalation.
- **Due By.** Time-window filters: *This Month*, *Next Month*, *This Quarter*, *Next Quarter*, *Show All Due By*. These are how a leader running the quarterly business review pulls just the milestones that hit inside the review window. Outside-window projects stay in the data but disappear from the working view.
- **Sort By.** *Target Date* orders the visible set by committed milestone date; *Finish Date* orders by the current forecast. *Program* groups all milestones by program for a program-by-program walk. *Clear All* resets every filter to its default state — one click to recover the full portfolio view.
- **Schedule.** *Critical Path To Selected* highlights the predecessor chain that drives the selected milestone. *Show Wigglechart* opens the per-milestone wigglechart (Figure 1) for the selected row — the leader can drill from the portfolio view to the live forecast curve in one click.

The combinatorial point is what matters. *Show: Slipping and Late* + *Risk: Show High Risk* + *Due By: This Quarter* is a three-click query that returns the projects that are behind, getting worse, flagged by the PM, and on the executive's near-term critical list. That is the working agenda for the review. The same query, asked of a traditional status-report stack, would take an analyst hours and produce a snapshot already a week stale.

© 2026 lateralworks

> *"Exception reporting asks what is going wrong, who owns fixing it, and whether the fix is on track."*

A second-order benefit emerges from this filtering pattern: leadership reviews become measurably shorter and more useful. In engagements where lateralworks has helped clients adopt the dashboard, weekly portfolio reviews have dropped from 90-minute walk-throughs of every program to 30-minute working sessions on the right-most cohort. The remaining time is reinvested in actually unblocking the troubled projects.

© 2026 lateralworks

Core principle

Visibility precedes action

"You can't accelerate a portfolio you can't see in time to act."

lateralworks engagement observation

across multi-project leadership reviews

© 2026 lateralworks

04

Economics

Live data, no overhead

A common objection to portfolio dashboards is the overhead they create: yet another reporting system the PMs must keep up to date, separate from their own planning artifacts, with predictable drift and predictable resentment. fastProjectAI sidesteps the objection because it does not introduce new data. It reads the data the PMs are already producing, in the format they are already producing it in, on the cadence they already produce it.

© 2026 lateralworks

Why the data is free

No parallel reporting system

In an FTTM-managed organization, every PM already maintains a live milestone schedule and updates the wigglechart for each milestone every week. The wigglechart is part of the planning discipline that makes FTTM work [7]; the PM keeps it to manage their own project, and the executive review is a downstream consumer. The wigglechart is how the PM knows whether the project is on track and which milestones need attention this week.

fastProjectAI reads the same source files. There is no separate data-entry burden, no parallel system that drifts out of sync with reality, and no incentive for the PM to present a different version of the data to leadership than they keep for themselves. The data is already in the PM's own working planning artifact; the dashboard aggregates it across the portfolio.

Three consequences.

- **The data stays honest.** There is no separate dashboard to fluff up before the executive review, because the dashboard is reading the same file the PM uses to manage the project. Inflating the dashboard would mean inflating the PM's own working data — against the PM's own interest.
- **The data stays current.** Updates to the project schedule flow through to the dashboard automatically on the next read cycle, typically weekly. There is no lag between the PM updating their plan and leadership seeing the updated plan.
- **The data stays cheap.** The marginal cost of adding a new project to the dashboard is the cost of pointing the aggregator at one more file. There is no per-project setup, configuration, or training needed beyond the FTTM planning practice itself.

The pattern echoes the broader observation in lean product development that adding measurement systems often degrades the thing being measured [8]. fastProjectAI sidesteps this by measuring the artifact PMs are already producing for their own planning rather than asking them to produce something extra for reporting. The economic argument also runs the other way: Hamel and Zanini have estimated the cost of excess management overhead in the U.S. economy at roughly three trillion dollars per year [15], much of it spent producing status data that is stale by the time it is read. A free-data dashboard recovers a sliver of that cost at the program level.

> **The free-data principle.** A dashboard that asks the PMs for nothing they aren't already doing has no resistance, no drift, and no reporting bias. The same dashboard that asks the PMs to maintain a separate status feed will be wrong within a month.

© 2026 lateralworks

05

Worked example

Site deployment portfolio

This section walks through one full leadership review using the dashboard, drawn from a recent lateralworks engagement with a utility-scale site deployment portfolio. The portfolio contains 196 milestones across 15 programs, ranging from 2-milestone programs to 19-milestone programs. The example illustrates how the dashboard surfaces the projects that need attention and how the patterns connect across programs.

© 2026 lateralworks

A 15-program portfolio

Where the trouble lives

The portfolio summary itself reads cleanly: 111 active milestones, 85 already completed, 68% of the active set within two weeks of target. Without the per-milestone detail, this would suggest a healthy portfolio. The detail tells a different story: 63 of the 111 active milestones (57%) are behind their target dates; 12 are severely behind; 31 are actively slipping rather than recovering. The trouble is concentrated.

Program	Active	Late	Risk 3	Avg gap (d)	Avg trend (d/wk)
Customer CO Brandon	10	7	3	-19	-0.6
Wasatch Peaks Ranch	2	2	0	-25	-7.6
APS 2025 Expansion	4	4	0	-18	-5.1
Customer CO Bryan	19	10	4	-12	-3.4
PSE 2025 Expansion	3	3	0	-13	-7.1
NW Mexico Lighthouse R1	11	11	0	-11	-1.4
Customer SPS Exp R1 R2	8	6	0	-9	-2.6
SoCal Lighthouse	4	3	0	-7	-4.0
Customer CO DN	19	9	4	-4	-1.7
All NextEra Stations	7	3	0	-2	-5.1
Utah Lighthouse R1	3	2	0	-8	-1.7
CO Excel-only BKP	5	1	0	+10	+0.4
Customer SPS R3 R4	10	1	0	+21	+4.5

Figure 7. Program-level health summary. Each row is one of the 15 programs, with active milestone count, count of late milestones, count of Risk-3 milestones, average gap, and average six-week trend. The bottom two rows are running ahead of schedule with improving trends; everything above is behind.

Five patterns become visible at this level of detail. First, the three Customer Colorado programs (Bryan, Brandon, DN) account for 48 of the 111 active milestones and contain all 11 of the portfolio's Risk-3 flagged projects. Second, three smaller new-market expansion programs — Wasatch Peaks Ranch, APS 2025, PSE 2025 — are 100% behind schedule with worsening trends. Third, the NW Mexico Lighthouse R1 program shows 11 milestones all behind schedule with zero Risk-3 flags — the risk assessment is clearly lagging the actual schedule. Fourth, the Customer SPS R3/R4 program is running 21 days ahead of schedule with an improving trend of +4.5 days per week. Fifth, the CO Excel-only BKP program is similarly ahead.

Each of these observations is one filter-and-sort on the dashboard, completed in seconds. None of them is visible from a traditional green-yellow-red status report.

© 2026 lateralworks

The most urgent projects

Ten that need action this week

Applying the severity filter narrows the working set further. Ten milestones need immediate leadership attention:

Site	Program	Gap	Trend	Risk
Pueblo West	Customer CO Bryan	-66 d	-16.8 d/wk	3
Steele Hollow	Customer CO Bryan	-66 d	-16.6 d/wk	3
Craig	Customer CO Bryan	-64 d	-10.6 d/wk	2
Vail	Customer CO DN	-49 d	-12.5 d/wk	3
Keystone	Customer CO Brandon	-46 d	-10.8 d/wk	3
Sugar Loaf	Customer CO DN	-40 d	0 d/wk	3
Aurora Reservoir	Customer CO Brandon	-36 d	-4.8 d/wk	3
Bellvue	Customer CO Brandon	-35 d	0 d/wk	3
Sunset Point	APS 2025 Expansion	-36 d	-9.0 d/wk	1
Agate	Customer CO Bryan	-28 d	-8.2 d/wk	3

Table 1. The ten most urgent milestones in the portfolio after applying active + late + worsening + severity filters. Six of the ten are in the same program family (Customer Colorado); three of those six have no documented corrective actions despite being the worst performers in the entire portfolio.

The shape of the data tells the leadership team where to focus. Pueblo West and Steele Hollow are the same project type, in the same program, with the same trend, and with no documented action plans — suggesting a structural failure in how that program is responding to slippage. Sugar Loaf and Bellvue have stopped getting worse but are stuck deeply behind — they need rebaselining. Sunset Point, in a newer expansion program, is replicating the same failure mode the Customer Colorado programs are stuck in.

© 2026 lateralworks

Trend distribution

The widening gap

Plotting the six-week trend rate for every active milestone reveals a pattern that is invisible in any single project view: the distribution of trends is bimodal. Projects that are on or ahead of schedule are pulling further ahead, while projects that are behind are falling further behind. The middle — projects recovering from moderate delays — is empty.

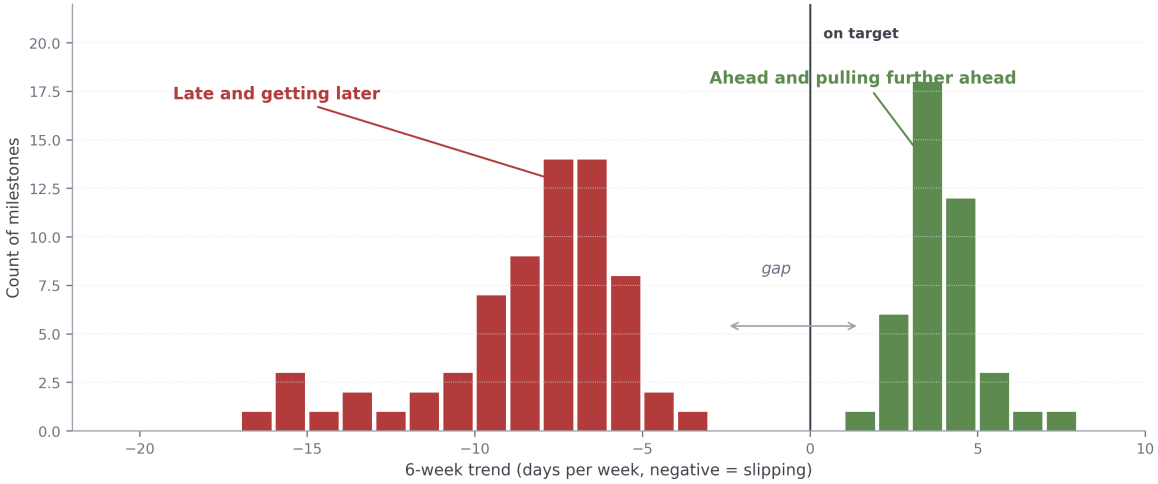

Figure 8. Six-week trend distribution across the active portfolio. Projects polarize into two camps; the recovery zone in the middle is empty. This means once a project falls behind by more than two or three weeks, the recovery mechanisms are not working.

This observation has direct implications for how the leadership team should allocate intervention resources. Adding more rigor to the projects already on track produces no return. Adding more rigor to projects in the worsening tail is the only place where leadership attention changes the outcome. The bimodal pattern recurs across most multi-project portfolios in lateralworks engagement data [26], and it has been observed in broader project management research as well: schedule slippage tends to compound rather than correct in the absence of explicit recovery mechanisms [9, 10].

© 2026 lateralworks

06

Pattern detection

LLM analysis across projects

The dashboard alone surfaces the projects that need attention. The next step is harder: identifying the structural patterns connecting them. This is where a large language model earns its keep. Given the structured weekly export of the dashboard, an LLM can read across all 196 projects at once, look for cross-project regularities, and surface findings no single-project view would reveal. The work in this paper used Claude (Anthropic's Claude Opus model) for the pattern analysis, but the approach is model-agnostic.

© 2026 lateralworks

What the LLM is doing

Pattern detection across 196 Projects

The workflow is simple. Each week, the dashboard exports its current state as a structured table: program, milestone, target date, current forecast, gap, trend, risk, and current actions. That table is fed to the LLM with a structured prompt asking it to identify cross-project patterns, structural risks, and recommended interventions. The LLM returns a written analysis that is reviewed by the program leadership.

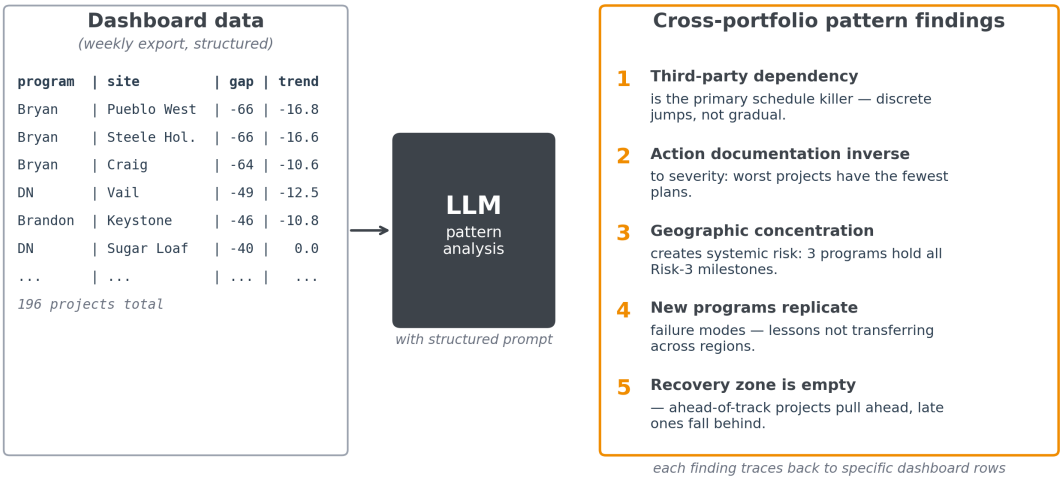

Figure 9. The LLM analysis layer. The dashboard's weekly export is fed to the LLM with a structured prompt. The LLM produces a narrative analysis identifying cross-portfolio patterns. Each finding traces back to specific rows in the dashboard so the analysis is verifiable.

What makes this work is the structure of the input. Because the dashboard data is already normalized into a common format — one row per milestone, with the same columns for every project — the LLM can apply the same analytical lens across the entire portfolio. There is no need to teach it the idiosyncrasies of each project's reporting format, because there are no idiosyncrasies. The aggregator already removed them.

The body of academic and industry research on LLM-assisted analytical work has expanded rapidly since the few-shot results reported in the GPT-3 paper [21] and the broader survey work that followed [11, 12]. The pattern that has consistently emerged is that LLMs do their best work when fed clean, structured data with a clear analytical prompt — not when asked to extract structure from messy inputs. fastProjectAI's aggregation step is the precondition that makes the LLM analysis work. The analytical mode is closer to what Tukey called exploratory data analysis [18] than to hypothesis testing — the LLM is asked to look at the shape of the data and report what stands out, with each finding traceable back to specific dashboard rows.

© 2026 lateralworks

Five findings

What the LLM surfaced

In the worked example portfolio, the LLM analysis identified five cross-portfolio patterns. Each is a property of the portfolio. The findings emerge in aggregate, not in any individual project view.

1. Third-party dependency is the primary schedule killer.

Across the portfolio, the largest single driver of schedule slippage is dependence on external parties — lease negotiations with landowners, carrier approvals, county permitting, state agency sign-offs. These dependencies do not erode schedules gradually; they slip in large discrete jumps as external approvals arrive late or stall. The current scheduling model treats them as predictable durations when they are high-variance events.

2. Action documentation is inversely correlated with severity.

The two worst projects in the portfolio — Pueblo West and Steele Hollow, both 66 days behind — have no documented corrective actions. Projects that are on or ahead of schedule have detailed action items with named owners and due dates. The projects that need the most intervention are getting the least. This is a known dysfunction in distressed-program management [13]: severely late projects tend to be implicitly written off and stop receiving the active management that less-distressed projects retain.

3. Geographic concentration creates systemic risk.

The three Customer Colorado programs account for 48 of 111 active milestones and contain all 11 Risk-3 flagged projects. Meanwhile the Customer SPS R3/R4 program is running 21 days ahead with improving trends. The schedule challenges are concentrated in a single regional environment. Possible contributing factors include regional regulatory complexity, higher density of private-landowner negotiations, and resource contention across three programs sharing the same field teams.

4. New programs are replicating old failure modes.

Three newer expansion programs — APS, PSE, Wasatch Peaks Ranch — are all 100% behind schedule with worsening trends. They are making the same third-party dependency estimation errors that have plagued the Customer Colorado programs. Lessons from the older programs are not being transferred to new market entries. This is a structural problem in how the organization captures and redistributes program-specific lessons learned [14].

5. The recovery zone is empty.

Confirming the bimodal trend pattern from Section 05: of the 63 milestones currently behind schedule, almost none show a positive recovery trend. Most are flat or worsening. The portfolio lacks an effective catch-up mechanism for distressed projects. Once a project falls behind by more than two or three weeks, structural recovery is rare.

© 2026 lateralworks

From findings to action

What the LLM recommends

The same LLM run produces a recommended action list grounded in the findings. Each item is tied to specific rows in the dashboard. Five recommendations follow from the five findings:

- **Rebaseline the runaway projects.** Pueblo West, Steele Hollow, Craig, Vail, and Keystone are 46-66 days behind and still worsening. Their current target dates are no longer meaningful. Rebaseline based on actual dependency-resolution timelines.
- **Mandate action plans for any project >2 weeks behind.** Make this a hard gate on the weekly dashboard review. The current pattern where the worst projects have the fewest plans must be reversed.
- **Add buffer multipliers for third-party dependencies.** Lease negotiations, carrier approvals, and permitting should be scheduled with 2-3x duration buffers based on actual historical variance, especially for new market entries.
- **Investigate what Customer SPS R3/R4 is doing differently.** A program running 21 days ahead with a +4.5 day-per-week trend is a replicable success pattern. Understanding the difference is a playbook for the rest of the portfolio.
- **Triage NW Mexico Lighthouse R1.** Eleven milestones all behind with zero Risk-3 flags means the risk assessment framework is not calibrating correctly for that program. A formal risk-recalibration review is needed before the program drifts further.

These recommendations are first drafts. The leadership team reviews them, accepts or modifies them, and assigns ownership. The LLM's job is to compress the time between dashboard data and a structured candidate analysis — from the days a human analyst would need to a few minutes — while preserving traceability back to the source rows.

What the LLM is and isn't. The LLM is reading the same dashboard the leadership team is reading, with the same data, on the same cadence. The difference is that it can hold all 196 projects in attention at once and look for patterns across all of them simultaneously, which a human reviewer cannot. The LLM complements human judgment, consistent with the human-oversight framing in current model-development practice [22].

© 2026 lateralworks

07

Implementation
Adopting the dashboard

Adopting fastProjectAI inside an organization is mostly a question of upstream prerequisites. The dashboard itself is straightforward to deploy; the precondition for it to add value is the FTTM planning discipline that produces the wigglecharts in the first place. This section covers the practical steps in order.

© 2026 lateralworks

Practical adoption

Three steps in order

1. Establish the planning discipline first.

fastProjectAI is only useful if the underlying wigglecharts are real. If the PMs are not maintaining live milestone schedules with weekly forecast updates, no dashboard will fix the problem — it will just expose the absence of data more visibly. The first step is to ensure FTTM-style planning practice is in place at the project level. Without it, the dashboard reports zeros or stale inputs.

2. Standardize the milestone schema.

Each milestone needs a target date, a current forecast date, an owner, and a risk flag — in a consistent column format across projects. The aggregator has to be able to read each project's schedule the same way. In practice, this is a one-time set-up cost: agree on the schema, create a template, roll it out across programs. Cooper, Edgett, and Kleinschmidt note that portfolio comparability is the precondition for portfolio management [4]; the milestone schema is what produces comparability.

3. Run the dashboard for four weeks before the LLM layer.

The dashboard alone delivers most of the value. The LLM layer adds cross-project pattern detection, but it only earns its keep once the dashboard data is stable enough that the patterns are real. Running the dashboard for four weeks before adding the LLM analysis lets the leadership team calibrate against the underlying data first, and lets the PMs settle into the discipline of weekly updates without the additional pressure of an LLM critique.

© 2026 lateralworks

What it costs

And what it doesn't

The marginal cost of running fastProjectAI inside an FTTM-managed organization is small. The PMs are already producing the data; the aggregator is a single piece of software that reads project schedules and renders the dashboard; the LLM analysis runs weekly against the structured export. Once set up, the running cost is dominated by the LLM API charges, which for a typical 15-program portfolio amounts to a few dollars per week.

The setup cost is one-time and modest: schema standardization, aggregator deployment, dashboard configuration, prompt design for the LLM layer. Most of the work is upstream — establishing the FTTM planning discipline that makes the data meaningful. Organizations that have invested in that discipline find the dashboard layer trivial to add. Organizations that haven't will spend most of their effort on the planning discipline itself, which is the real foundational asset.

What the dashboard does not replace: program leadership, PM judgment, executive decision-making, or the relationships and context that make a program work. It is a visibility layer, not a management layer. Its purpose is to tell leadership where to look and what to ask. The actual work of leading the projects remains where it always was — with the people running them.

> **The economic logic.** The dashboard is free if the planning discipline is already in place. The planning discipline is not free, but it is the foundational asset every other capability rests on. Organizations that try to install the dashboard without the planning discipline will get either empty rows or fictional ones; either way, no value.

© 2026 lateralworks

Sources

References

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

[2] Lewis, J. P. *Mastering Project Management: Applying Advanced Concepts to Systems Thinking, Control and Evaluation, and Resource Allocation*. McGraw-Hill, 2nd ed., 2008.

[3] U.S. Government Accountability Office. "Schedule Assessment Guide: Best Practices for Project Schedules." GAO-16-89G, December 2015. https://www.gao.gov/products/gao-16-89g

[4] Cooper, R. G., Edgett, S. J., and Kleinschmidt, E. J. *Portfolio Management for New Products*. Basic Books, 2nd ed., 2001.

[5] Smith, P. G., and Reinertsen, D. G. *Developing Products in Half the Time: New Rules, New Tools*. Wiley, 2nd ed., 1997.

[6] Drucker, P. F. *The Effective Executive*. Harper & Row, 1967.

[7] lateralworks. "FTTM Reference Guide: Self-Paced Tutorial." Internal methodology document, 2024. lateralworks.com.

[8] Reinertsen, D. G. *Managing the Design Factory: A Product Developer's Toolkit*. Free Press, 1997.

[9] Vanhoucke, M. *Measuring Time: Improving Project Performance Using Earned Value Management*. Springer, 2010.

[10] Lipke, W. "Schedule is Different." *The Measurable News*, Summer 2003, College of Performance Management, pp. 31-34.

[11] Bommasani, R., et al. "On the Opportunities and Risks of Foundation Models." Stanford CRFM Technical Report, 2021. https://crfm.stanford.edu/report.html

[12] Bubeck, S., et al. "Sparks of Artificial General Intelligence: Early experiments with GPT-4." Microsoft Research, 2023. arXiv:2303.12712.

[13] Keil, M., and Robey, D. "Turning Around Troubled Software Projects: An Exploratory Study of the Deescalation of Commitment to Failing Courses of Action." *Journal of Management Information Systems*, vol. 15, no. 4, 1999, pp. 63-87.

[14] Argote, L. *Organizational Learning: Creating, Retaining and Transferring Knowledge*. Springer, 2nd ed., 2013.

[15] Hamel, G., and Zanini, M. "Excess Management Is Costing the U.S. $3 Trillion Per Year." *Harvard Business Review*, September 2016. https://hbr.org/2016/09/excess-management-is-costing-the-us-3-trillion-per-year

[16] Wheelwright, S. C., and Clark, K. B. *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*. Free Press, 1992.

[17] House, C. H., and Price, R. L. "The Return Map: Tracking Product Teams." *Harvard Business Review*, January-February 1991, pp. 92-101.

[18] Tukey, J. W. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[19] Lee, H. L., Padmanabhan, V., and Whang, S. "Information Distortion in a Supply Chain: The Bullwhip Effect." *Management Science*, vol. 43, no. 4, 1997, pp. 546-558.

[20] Kahneman, D. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

[21] Brown, T., et al. "Language Models are Few-Shot Learners." Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877-1901. arXiv:2005.14165.

[22] Anthropic. "Constitutional AI: Harmlessness from AI Feedback." arXiv:2212.08073, 2022.

[23] Pinto, J. K., and Mantel, S. J. "The Causes of Project Failure." *IEEE Transactions on Engineering Management*, vol. 37, no. 4, 1990, pp. 269-276.

[24] Ohno, T. *Toyota Production System: Beyond Large-Scale Production*. Productivity Press, 1988.

[25] Goldratt, E. M. *Critical Chain*. North River Press, 1997.

[26] lateralworks. "Internal assessment database: portfolio visibility engagements." Engagement data across client programs, 2018-2026.

© 2026 lateralworks