



Whitepaper

# Report by exception

How one page built from the weekly schedule files puts the five slipping initiatives in front of leadership, and leaves the other nine alone

## FTTM best practices series.

A worked example of a one-page program executive summary generated from weekly Microsoft Project files and wigglechart trend workbooks, and the reporting rule behind it. The program shown is illustrative and built from synthetic data. Its shape and its failure patterns are representative of a multi-project cost reduction portfolio tracked with fastProjectAI.

---

**Prepared by**

lateralworks  
Neal Mitchell

**Date**

October 2026  
White paper

**Online**

lateralworks.com  
FTTM best practices series

# Table of contents

## Report by exception

---

|                                                 |    |
|-------------------------------------------------|----|
| Abstract                                        | 03 |
| Fourteen initiatives, one page, five minutes    | 04 |
| What “slipping for more than three weeks” means | 07 |
| From weekly files to the page                   | 12 |
| What leadership does with the page              | 14 |
| References                                      | 16 |

**Core thesis.** A program status report earns its place in a leadership meeting by what it leaves out. A snapshot that colors nine of fourteen initiatives red tells leadership nothing about where to spend the next hour. A page that applies one written rule to thirteen weeks of forecast history, surfaces the five initiatives whose dates keep moving, and quotes the team's own stated cause for each, tells them exactly that. The rule and the verbatim cause are the report. Everything else is context.

## Overview

# Abstract

---

**Most program dashboards answer the wrong question.** They tell leadership where every project stands today. Leadership needs to know which projects are getting worse and why, because those are the only ones where an executive decision still changes the outcome. The example in this paper is a fourteen-initiative margin improvement program. At the September 29 refresh, nine of its fourteen tracked targets forecast late and six sit more than four weeks late. A conventional red-amber-green page would ask leadership to triage nine problems in a one-hour review. The exception page asks them to look at five.

The five are chosen by a rule, not by judgment. An initiative is an exception when its forecast completion date moved later in at least two of the last four weekly refreshes and finished the window later than it started. The rule reads the pattern in the forecast rather than its current value. It catches an initiative sliding a week at a time long before the health band turns red, and it leaves alone an initiative that took one large, explained hit and has held its date since.

The page is generated, not assembled. Each week the program's native Microsoft Project file and its wigglechart workbook (a wigglechart plots a milestone's forecast date at every weekly refresh against its fixed target) go into a Claude skill that lateralworks built for the purpose. A skill is a written procedure that the model runs the same way every time. It extracts the schedule, applies the rule to the trend history, draws the Gantt of the first two WBS levels across the top, and lays out one trend card per exception across the bottom, with the cause of change quoted verbatim from the workbook. A numbered badge ties each card to its Gantt row. A verification script refuses to release the page if any number on it disagrees with the source files.

The result changes what the leadership meeting is for. The status recital goes away. The hour goes to the three questions lateralworks has put to exception projects since 2019: is the trend improving, which path is pushing the date out, and what does the team need from the people in the room to pull it back [1]. The paper walks through the page, the rule, the pipeline that builds it, and the meeting it is built to run.

## 01

## Section one

**Fourteen initiatives,  
one page, five minutes**

A cost reduction program is a portfolio problem wearing a project costume. The margin improvement program in Figure 1 has fourteen initiatives under two summary bars, gross margin and operating expense, with completion targets spread from October 2026 to April 2027. Each one has an owner, a plan bar and a target marker. None is large enough to justify its own steering committee. All of them together are too many to review in an hour.

That is the problem the one-page summary solves. The top half shows every initiative at WBS levels 1 and 2 (the program and its two summary bars, then the initiatives beneath them) so nobody in the room can claim they were not told. The bottom half shows only the initiatives that meet the exception rule, with enough history on each to support a decision. The page is built to be read in five minutes and argued about for fifty.

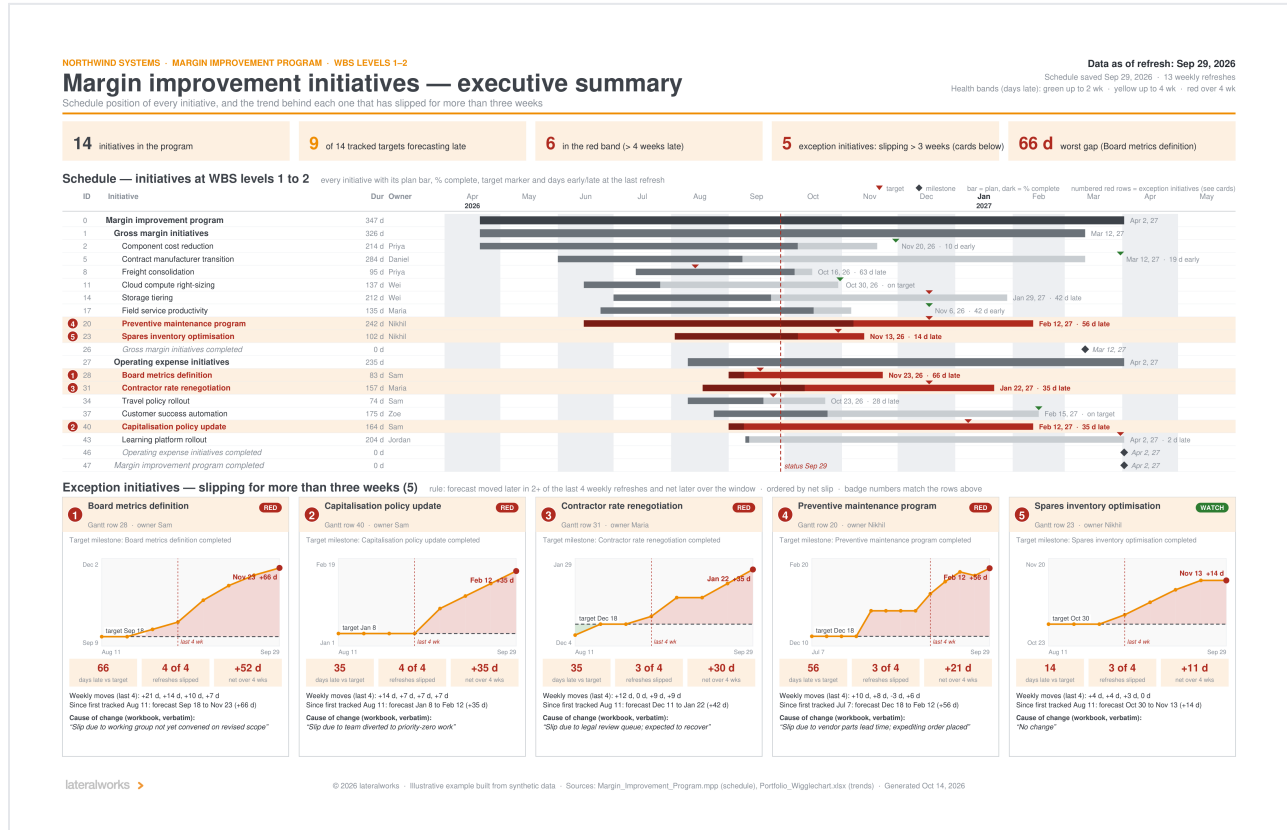


Figure 1. The one-page executive summary as generated for the September 29, 2026 refresh. Gantt of WBS levels 1 and 2 on top; one wigglechart trend card per exception initiative below. Numbered red rows in the Gantt match the badges on the cards. Illustrative program, synthetic data.

The five headline numbers across the top of the page tell the story of the whole program in one line: fourteen initiatives, nine forecasting late, six in the red band, five exceptions, and a worst gap of 66 days on the board metrics definition. The fourth number is smaller than the second and the third, and that is the point. Exceptions are not the late initiatives and they are not the red ones. They are the initiatives whose dates are still moving.

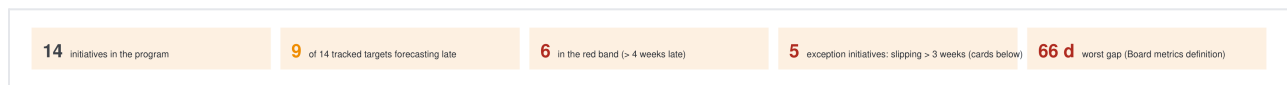


Figure 2. The headline strip. Nine late, six red, five exceptions. The exception count is deliberately not the same as the late count.

## Why status reports fail leadership

The research on project status reporting is unkind to the conventional dashboard. Keil, Smith, Iacovou and Thompson reviewed fifteen years of studies on how project status travels up an organization and summarized the findings as five inconvenient truths: executives cannot rely on staff to report status accurately; personality and workplace culture distort what gets reported; auditing does not fix misreporting and can make it worse; a more powerful sponsor gets less accurate reports, not more; and executives frequently ignore the bad news they do receive [2]. Social psychology has a name for the underlying habit. Rosen and Tesser called it the MUM effect in 1970: people are reluctant to pass on unwelcome information [3]. Iacovou and colleagues later measured it in a survey of 561 project managers and found that selective reporting is routine, and that optimistic bias degrades the quality of what reaches the executive [4].

A red-amber-green status field is the perfect host for these habits. The color is a judgment, the judgment belongs to the person being judged, and the only evidence behind it is a conversation. A forecast date in a weekly schedule refresh is a harder thing to shade. It comes out of the critical path, it is tied to the task network, and it is recorded every week whether or not anyone wants to talk about it. The exception page is built on those dates and only on those dates. Nobody rates anything.

## Management by exception, properly done

Reporting by exception is old. Frederick Taylor described the exception principle in Shop Management in 1903: the manager should receive only condensed, comparative reports covering the exceptions to past standards, so that attention goes where it is needed [5]. What is new is a cheap, trustworthy way to compute the exceptions from source data every week. Taylor's exception principle needed a clerk to decide what counted as an exception. This page needs a rule. Write it once, let software apply it, and print it where anyone can argue with it.

lateralworks made the same argument for new-product portfolios in 2019: aggregate every project's trend into one view, then drill down by exception into the ones at risk of missing their date [1]. The one-page summary is that advice made concrete for a cost program. The aggregate is the Gantt. The drill-down is the row of cards. The exception filter is the rule in the next section.

## 02

## Section two

**What “slipping for more than three weeks” means**

---

The rule printed under the card row reads: forecast moved later in two or more of the last four weekly refreshes, and net later over the window. Both halves matter. The first half catches a pattern of repeated small slips. The second half excludes an initiative that slipped twice and recovered more than it lost. The window is four weeks because, in our experience, three slips in a row is the point at which a team has stopped absorbing variance and started reporting it, and four refreshes is the shortest history in which three can appear.

The rule has two properties a health band lacks. It does not care how late the initiative is, and it does not care who owns it. Spares inventory optimisation is only 14 days late and would be yellow on any conventional page, yet it is card 5 because its forecast has moved in three of the last four weeks. Freight consolidation is 63 days late and would be the reddest row on a conventional page, yet it has no card, because whatever happened to its date happened before the window opened. One of those is a problem leadership can still influence. The other was decided weeks ago.

The rule also has a blind spot, and it is deliberate. A single 60-day jump in one week does not qualify, because one move is not a pattern. That initiative shows up in the Gantt as a red row with a large number beside it, and it belongs to the monthly project deep-dive, where one schedule gets an hour. The exception row is for the slow leaks, which are the ones a snapshot never shows.

## Reading a wigglechart

Each card is a wigglechart: the forecast completion date plotted at every weekly refresh since the initiative was first tracked, against a dashed line at the target. A flat line is a stable forecast. A rising line is a slipping one. The slope is the rate of slip, and a line that steepens is the single most important thing on the page, because it means the team's own estimate of the remaining work is still growing. The shaded area between the line and the target is the accumulated gap. The red dashed vertical marks the start of the four-week rule window.

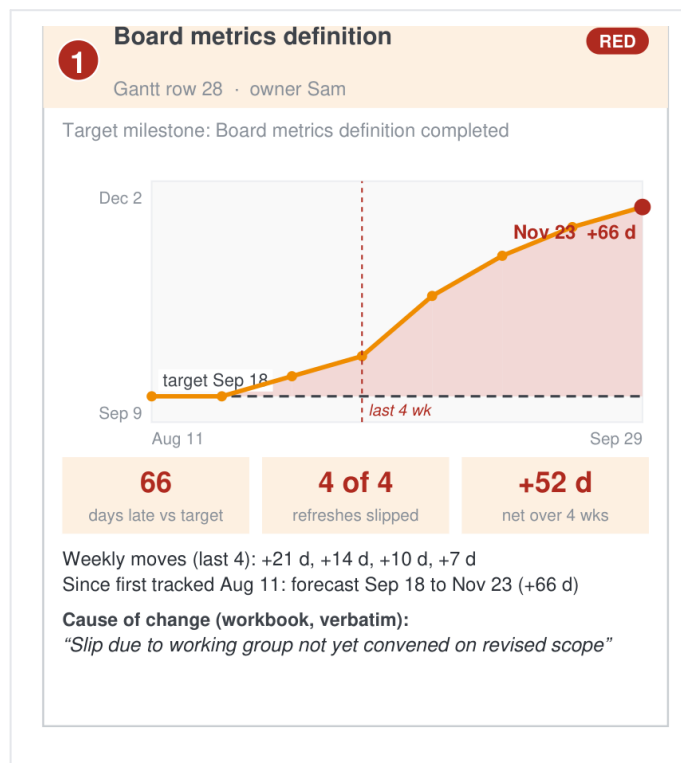


Figure 3. Card 1, board metrics definition. Four slips in four weeks, decelerating (+21, +14, +10, +7 days), 66 days late against a September 18 target. The stated cause is a working group that has not convened.

Card 1 is the program's worst gap and its clearest story. The forecast was at target when tracking began on August 11. It moved 14 days over the first weeks, then 21, 14, 10 and 7 in the last four. Four of four refreshes slipped. The weekly moves are shrinking, which is the one piece of good news on the card. The workbook cause of change, quoted verbatim, is a working group not yet convened on revised scope. That is a calendar problem, and the people in the leadership review can fix it by Thursday. This is the exception the page exists to surface: 66 days of slip with a two-day remedy.

Card 2, capitalisation policy update, shows the opposite shape. The forecast held at target for weeks, then moved 14 days and then 7, 7 and 7. The cause is the team being diverted to priority-zero work. The weekly moves are a metronome: one week of slip per week elapsed, which is what a stopped initiative looks like on a wiggglechart. Nobody is working on it, and the forecast is tracking the calendar. The question for leadership is whether the diversion was a decision anyone made.

## The one that is recovering

Card 4, preventive maintenance program, is the reason the rule says two of four refreshes rather than four of four. Its weekly moves are +10, +8, -3 and +6 days. One of the last four refreshes pulled the date in, because the team placed an expediting order against a vendor parts lead time, and the cause of change says so. It is still an exception at 56 days late with three of four refreshes slipped, but the card shows a team that has acted. Card 1 needs a meeting scheduled. Card 4 needs someone to ask whether the expedite is funded and whether a second supplier exists.

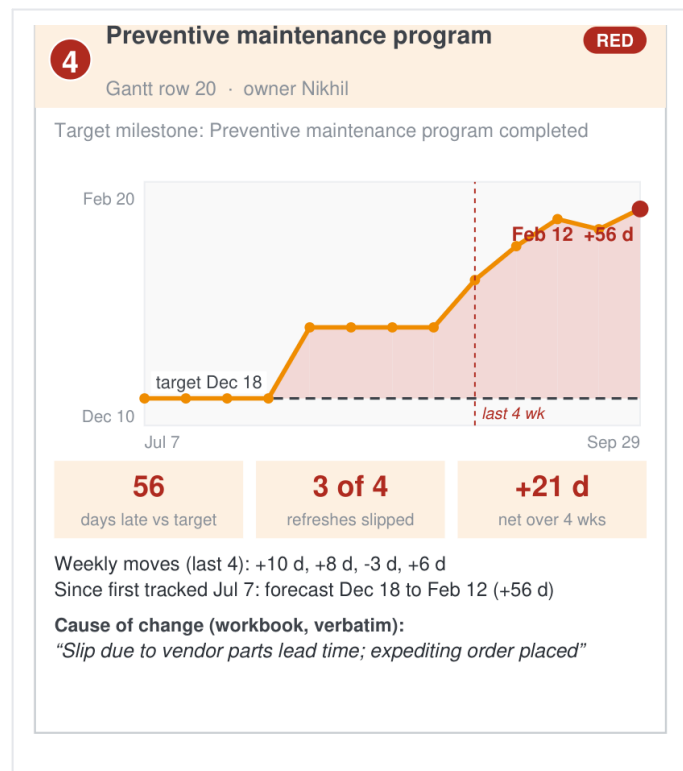


Figure 4. Card 4, preventive maintenance program. The dip in week three is a 3-day pull-in from an expediting order. The trend card makes a team's corrective action visible in the same frame as the problem.

## “No change” is a finding

Card 5, spares inventory optimisation, carries a cause of change that reads, verbatim, “No change.” The forecast moved 4, 4, 3 and 0 days across the window and the workbook explains none of it. On a conventional report this would be invisible: 14 days late, yellow band, reassuring note. On the exception page it is card 5 with an empty explanation, and the emptiness is the finding. A forecast that drifts eleven days in a month with no stated cause is either an owner who has not looked at the schedule or a problem the owner has not named. Either way it belongs in the room.

The verbatim rule is deliberate. The skill may not paraphrase, summarize or soften the cause of change. It copies the cell. That removes the layer of interpretation where the MUM effect lives, and it holds the owner to the words they chose in the workbook. Owners learn quickly that “No change” next to a rising line is a worse look than an honest sentence.

Exception reporting

## **The signal and the noise**

---

**Nine of fourteen are late.  
Five need the room.  
The page knows  
the difference.**

Margin improvement initiatives, executive summary

Illustrative program built from synthetic data, refresh of Sep 29, 2026

# 03

## Section three

### **From weekly files to the page**

The page in Figure 1 took no one any time to lay out. Two files go in, one PDF comes out, and the only human judgment in the loop is the one written into the rule. The discipline of the page depends on that. A report someone assembles by hand each week drifts: the layout changes, a card gets dropped because it was awkward, a cause of change gets tidied. A generated report cannot drift, and when it is wrong, it is wrong in a way that traces to a cell.

The inputs are the same two files a fastProjectAI program already maintains. The schedule is the native Microsoft Project file as saved after the weekly refresh. The trend history is the wigglechart workbook, one row per tracked target per refresh, with the forecast date and the cause of change the owner entered. Neither file is prepared for the report. The report is prepared from them.

## Four steps, one gate

The skill runs in four steps and ends at a gate. Each step produces a file the next step reads, and nothing is carried in memory between them.

| Step                | What happens                                                                                                                                                                                                                                                                                              | What it produces                                     |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| 1. Extract          | The .mpp file is read with the mpxj library through the lateralworks mpp-extract skill. Tasks, outline levels, durations, percent complete, owners and target milestones come out as structured tables. fastProject custom fields such as Activity Owner and Target Milestone resolve into named columns. | tasks.csv, project.json, summary.md                  |
| 2. Read the trend   | The wigglechart workbook is parsed into one series per tracked target: refresh date, forecast date, days early or late against target, cause of change. The last four refreshes are isolated for the rule.                                                                                                | One trend series per target                          |
| 3. Apply the rule   | For each target: count the refreshes in the last four where the forecast moved later, and compute net movement over the window. Flag as exception when the count is two or more and the net is positive. Order exceptions by net slip and assign badge numbers.                                           | The exception list, with badge numbers and card data |
| 4. Lay out the page | Gantt of WBS levels 1 and 2 across the top with exception rows tinted and badged. One card per exception across the bottom with the wigglechart, the metrics, and the cause of change copied verbatim.                                                                                                    | One-page PDF                                         |
| Gate                | A verification script recomputes every number on the page from the source files and compares. Any mismatch, missing badge, or paraphrased cause of change blocks delivery. The page is never hand-edited.                                                                                                 | Released PDF, or a failure report                    |

The gate is the part most teams skip and the part that makes the rest trustworthy. A page that leadership reads in five minutes has to be right in five minutes. If the badge on card 3 pointed at the wrong Gantt row, or the 66-day gap on card 1 disagreed with the annotation on row 28, the page would lose its authority and the meeting would revert to arguing about the data instead of the initiatives. The verification script exists so that argument never starts.

## What the generic skill carries between clients

One configuration file sets the pipeline up per client: program title, WBS depth, workbook layout, health-band thresholds, the pattern rule itself, and branding. The rule is a parameter because a program that refreshes every two weeks needs a different window than one that refreshes weekly. What does not change between clients is the structure: aggregate on top, exceptions below, verbatim causes, numbered badges, a gate. That structure is the lateralworks position on what a program summary should be, and the skill enforces it.

The same pipeline builds the weekly refresh record lateralworks described in May 2026, the written memory of what changed and who owns the next step [6]. That record is for the people in the program. The exception page is for the people above it.

# 04

## Section four

### **What leadership does with the page**

---

The page is designed to run a specific meeting. Leadership reads the headline strip and the Gantt in the first two minutes and confirms that nothing has been hidden. The next fifty minutes go to the cards in badge order. Every card gets the same three questions [1].

- › **Is the trend improving?** Read the slope of the last four points. Decelerating slips (card 1) and a pull-in (card 4) are different conversations from a metronome (card 2).
- › **Which path is pushing the date out?** The cause of change names it. A working group not convened, a legal review queue, a vendor lead time, a diverted team. Each of these is a long pole somebody in the room can shorten.
- › **What does the team need from this room?** A decision, a priority call, a budget line, a phone call to a supplier. The ask leaves the meeting with an owner and a date.

## Targets stay fixed

The rule only works because the target does not move. The dashed line on every card is the completion date the program committed to when the initiative was chartered, and it stays there regardless of what the forecast does. lateralworks has argued since 2019 that there is only one real date and it comes from outside the project [7], and more recently that a fixed target is what makes a forecast trend readable at all [8]. If the target were re-baselined every time the forecast moved, every card would show a flat line at zero and the page would be worthless. The discipline of the page is that it never re-baselines.

## Pull in, do not just recover

The exception page is a late-stage instrument. By the time an initiative is a card, it has already lost weeks, and in a cost program every week late is a week of that initiative's savings that never reaches the P&L.; The practice that keeps initiatives off the page is pull-in (accelerating work to bank time before you need it): the weekly refresh exists to find schedule gains ahead of the problems that will consume them, so that when a vendor part is late the program has buffer to absorb it [9]. Card 4 shows what that looks like in miniature. The expediting order is a pull-in action, taken late. The same action taken in July, when the initiative was first tracked and its forecast was still at target, would have kept the card off the page.

A program that runs the weekly refresh well should see its exception count fall over time, and the page makes that visible. A steady five cards a month is a program that is managing slips. Five cards falling to two is a program that is preventing them. The two-page version of this report that lateralworks runs with clients every two weeks gives its second page to the teams that pulled their forecasts in, for exactly this reason.

## Control without the cost of control

The objection to exception reporting is that it is a control system, and lateralworks has been clear that tighter control systems do not create faster projects [10]. The distinction is in who does the work. A control system that asks fourteen owners to prepare fourteen status packs each week consumes the time it is meant to protect. A generated page consumes none of the owners' time beyond the schedule refresh they already do and the one sentence they already write in the workbook. The cost of the report falls on the software. The attention of the room falls on five initiatives. A cost reduction program should be the first to recognize that trade.

**What to do Tuesday morning.** Pick one program. Compute its exceptions from forecast history, not from status colors. Write the rule down and print it on the page. Keep the target fixed. Quote the cause of change verbatim. Tie every card to its Gantt row with a number. Gate the page on a recomputation from source. Then spend the meeting on the three questions, and expect the card count to fall.

# Sources

## References

---

- [1] Aggregate, then Drill Down. lateralworks Ideas, January 19, 2019.  
<https://www.lateralworks.com/ideas/aggregate-then-drill-down>
- [2] Keil, M., Smith, H. J., Iacovou, C. L., and Thompson, R. L. "The Pitfalls of Project Status Reporting." *MIT Sloan Management Review*, Spring 2014 (published online March 18, 2014). Accessed October 14, 2026.  
<https://sloanreview.mit.edu/article/the-pitfalls-of-project-status-reporting/>
- [3] Rosen, S., and Tesser, A. "On Reluctance to Communicate Undesirable Information: The MUM Effect." *Sociometry*, vol. 33, no. 3, 1970, p. 253. Cited via The Decision Lab, "The MUM Effect." Accessed October 14, 2026.  
<https://thedecisionlab.com/reference-guide/sociology/the-mum-effect>
- [4] Iacovou, C. L., Thompson, R. L., and Smith, H. J. "Selective Status Reporting in Information Systems Projects: A Dyadic-Level Investigation." *MIS Quarterly*, vol. 33, no. 4, 2009, pp. 785–810. Accessed October 14, 2026.  
<https://aisel.aisnet.org/misq/vol33/iss4/11>
- [5] Taylor, F. W. *Shop Management*. Paper presented to the American Society of Mechanical Engineers, 1903. Project Gutenberg edition. Accessed October 14, 2026.  
<https://gutenberg.org/ebooks/6464>
- [6] The Weekly Schedule Refresh. lateralworks Ideas, May 8, 2026.  
<https://www.lateralworks.com/ideas/the-weekly-schedule-refresh>
- [7] Targets and Trends. lateralworks Ideas, January 16, 2019.  
<https://www.lateralworks.com/ideas/targets-and-trends>
- [8] Where targets come from. lateralworks Ideas, August 11, 2026.  
<https://www.lateralworks.com/ideas/where-targets-come-from>
- [9] Pull-in before you slip. lateralworks Ideas, June 22, 2019.  
<https://www.lateralworks.com/ideas/pull-in-before-you-slip>
- [10] Do tighter control systems create faster projects? lateralworks Ideas, June 22, 2019.  
<https://www.lateralworks.com/ideas/do-tighter-control-systems-create-faster-projects>
- [11] Margin improvement initiatives — executive summary. lateralworks, one-page program summary generated from `Margin_Improvement_Program.mpp` and `Portfolio_Wiggglechart.xlsx`, data as of refresh September 29, 2026, generated October 14, 2026. Illustrative example built from synthetic data. Supplied file: `Program_Exec_Summary_Example.pdf`.

Program data in Figures 1 to 4 is from reference [11]. The rule and the pipeline are lateralworks practice from client work, 2025 to 2026.