

## Briefing

# The weekly schedule refresh

Seven inputs, five sections, generated in minutes.

**A weekly pull-in meeting only works if it leaves behind an artifact the team can act on.** The schedule refresh is that artifact: a record of where the program stands, what moved this week, who is doing what next, and what could derail it. Built directly from the schedule and the meeting transcript in minutes rather than days, it turns a recurring meeting from a status ritual into the way the program actually steers.

Most program weeklies drift toward theatre. The PM walks the room through a status deck, the leads nod, owners take notes, and three days later no one remembers what was decided. The schedule refresh is the antidote. It is structured around the schedule itself rather than around a deck, captures the same seven inputs every week in the same order, and ships within hours of the meeting.

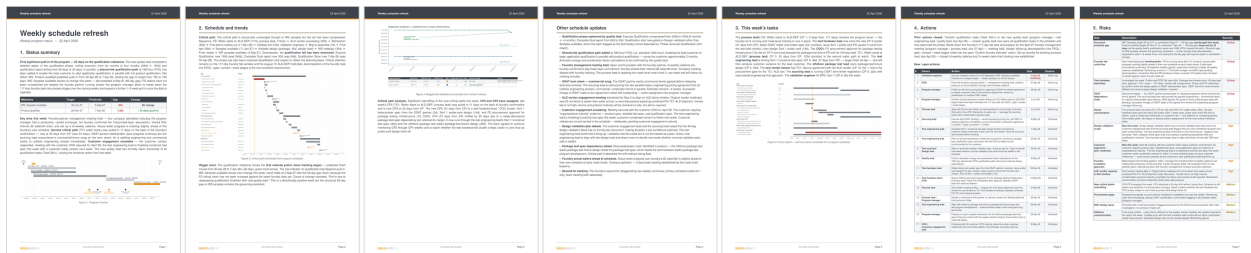


Figure 1. The seven pages of a real schedule refresh — status summary, schedule and trends, this week's tasks, actions, and a running risk register. Anonymized example attached as a downloadable companion to this briefing.

## What goes in

Six inputs come straight from the schedule. **Critical paths against last week's baseline** rank CP1 through CP10 by gap to target, with each path's change since last week. **Gap analysis to targets** shows the predicted-vs-target gap for every committed milestone and how it moved. **The wigglechart** traces predicted finish dates over the program's lifetime — a line trending up is bad, a line trending down is the team winning back time. **The action log** carries open and closed actions with owner, due date, and source. **Next week's focus** lists the work happening on critical and near-critical paths, by owner, with duration and float. **The running risk register** tracks every program risk at current severity and flags what changed since last week.

A seventh input is the meeting itself. **The pull-in meeting transcript** is where new actions get agreed verbally, concerns surface in passing, pull-in commitments are made or pushed back on, and observations about program tempo are voiced for the first time. None of that lives in the schedule. Without the transcript the report captures what moved; with it, the report captures why it moved and what the team agreed to do about it.

## Why the artifact matters more than the meeting

The meeting is where decisions are made. The artifact is what lets them stick. Without it, the same questions come up week after week — what did we decide about ALD, is the OSAT term sheet signed, what is our gap on samples now — because no canonical answer exists. With it, the team has a shared answer they can challenge, accept, or revise.

The artifact also exposes drift. When a structural gap on a milestone has not moved in eight weeks, the wiggglechart shows it as a flat horizontal line. When near-critical paths quietly migrate toward zero float, the critical-path table shows the rank changes. Both stand out at a glance — better than they ever do buried inside meeting minutes.

**The schedule refresh is not a deliverable.** It is the program's weekly memory. Without a structured artifact every issue arrives as a surprise; with one, the program steers from data rather than from recall.

## The generation problem

Compiling a seven-page schedule refresh by hand is a six-to-eight-hour job. The PM exports the schedule, runs a critical-path analysis, computes gap deltas, updates the wiggglechart, types out next week's tasks by owner, redrafts the risk register, re-types the action log, and works through the meeting transcript pulling out new actions and risks raised in conversation. By the time the artifact is ready, the meeting is three days old and most readers have moved on. That cost — multiplied by 52 weeks — is the friction that kills the practice for most programs. Teams know they should be producing this artifact. They cut it because the bill looks like a full-time PM's worth of work.

## What changes when an LLM does the assembly

Assembly time drops from a day to a few minutes. The model ingests the schedule export, the previous week's refresh, the meeting transcript, and the action log, and produces a draft ready for PM review. The report becomes consistent across editions because critical-path numbering stays stable and risk severities follow the same scale week after week — readers spot what changed because the surrounding scaffolding does not. The PM's time goes to what the artifact is for: running the meeting and pushing the decisions the data calls for.

A program running weekly schedule refreshes in this format knows where it stands every Friday afternoon — in writing, with deltas marked, with next-week focus by owner, and with risks ranked. Every meeting builds on the last one. Every issue gets a ticket, an owner, and a due date. Six months in, the artifact is the program's history rather than just its status. A program without one runs on memory, and memory is partial and short-lived.

Weekly schedule refresh generation is one of the standard functions in **fastProjectAI**. The seven-page sample attached to this briefing was produced from a real semiconductor program schedule, anonymized; structure preserved.