

Getting reality into the schedules

A new product development program is on a collision course with failure when reality does not get into the schedules. The problems themselves are solvable. Problems exposed early almost always are; problems discovered in the eleventh hour rarely are, unless you get lucky. That is the first principle of FTTM: **pull the pain forward**. Surface the real gaps now, then rationally develop the actions that close them. The table below is the checklist: seven observations that recur across NPD portfolios, the root cause behind each, and the actions only the general manager can take.

What we see	Root cause	Your actions
1. Leaders are not leading PDT leaders are still asking what their role is. Leaders given a problem start organizing people to solve it. They do not complain about missing skills; they find out what is missing and fix it.	The role is carried part-time with no shared definition of the PDT leader / PM split, inside a culture where stepping out has historically been punished.	■ Tell each PDT leader directly: you own the outcome. Organize people around the problem. Stop waiting for a role definition. ■ Dotted-line every PDT leader to a single overall owner. ■ Judge them on whether they find and fix what is missing, not on how well they report it.
2. Confirmation bias on schedules The teams have convinced themselves they are on schedule even though they know they are not. You are not looking at the true schedule gap.	Schedules were built from derivative-product templates and go unchallenged. A happy schedule feels safer than a real one.	■ Ask each PDT leader two questions: does your schedule reflect reality today, and does it include every iteration you expect to need? ■ Treat a schedule with no gap to target as a red flag, not good news. ■ Make schedule currency by a fixed day each week a standing, non-negotiable expectation.
3. The critical 10% is unmanaged Each project has a few hard technical problems holding up the 90% the teams have done a hundred times. That 10% is presently unmanaged, by the wrong people.	Research-grade problems are running inside product schedules, splitting the few experts who can solve them across teams and meetings.	■ Isolate the 10%. Stand up a dedicated team: your best people, roughly ten, three months, nothing else on their plate. ■ Give them everything they ask for. Remove every distraction. ■ Run their plan on twice-weekly refreshes with executive visibility.
4. The schedules are not real The real schedule rarely exists in any project. Most carry a single learning cycle where the work will demand more.	Optimistic NPI templates applied to NPD work. First-of-a-kind iteration was never planned in, because the research was never done ahead of the products.	■ Direct every PM to plan learning cycles explicitly: about two for easy problems, three or four for medium, five or more for hard. ■ Require a worst-case placeholder on every open unknown so the schedule stays closed-loop. ■ Walk each full schedule with its PDT leader for one hour: the problems, the timeline, the customer commitment.

What we see	Root cause	Your actions
5. No safe way to surface bad news The real gaps stay hidden because surfacing them is dangerous. Gaps exposed early can almost always be closed. Gaps exposed in the eleventh hour rarely can.	An organizational history where the people who stepped out into the wind got blown away. Protection has never been visible.	■ Publicly back the first person who surfaces an ugly gap. Make an example of the right behavior. ■ Reward gap-surfacing in reviews. Never punish the messenger. ■ Say it out loud, often: bad news early is good news.
6. The resource excuse is untested "Not enough resources" is the standing explanation for slow progress, and it has never been checked against data. It is rarely as big as they think.	No one has made the PDT leaders quantify the ask, so the excuse stays vague and unanswerable. It usually comes down to a few key people with key skills, not an army.	■ Ask each PDT leader: how many people do you need to accelerate? What skills are missing today? If you had everything, what would that look like? ■ Expect names and numbers, not headcount totals. ■ Fund the specific asks fast. Kill the general excuse.
7. Excuses are still on the table The trick to FTTM is to remove all their excuses, then hold their feet to the fire. Today the excuses are still available, and the teams are counting on them when it fails.	Dedication is treated as a preference, roles are stacked on existing jobs, and accountability arrives only after the miss.	■ Remove every named excuse, in writing: people, skills, decisions, access. ■ Then hold delivery accountability with no exceptions. ■ Get the dashboard, unfiltered. Insist on seeing the instruments; schedule the review.
Start this week. Ask the two schedule questions of all of your PDT leaders, and put the unfiltered dashboard review on the calendar. Everything else on this checklist follows from seeing the true gap. Until then you are flying in the clouds with instruments you are not being shown.		

A

The seven observations, expanded

A1 · Leaders organize; they do not ask

FTTM defines the PDT leader as the CEO of the product: looking ahead, clearing obstacles, provisioning the team. The PM is the COO, head down in the near-term detail. When a leader asks what their role is, the answer is the problem in front of them. Best practice: leaders inventory what is missing across skills, decisions, and access, close the gaps themselves, and escalate only what they cannot reach. A leader who reports missing skills without a plan to acquire them is describing the job rather than doing it.

A2 · The gap is the tool

FTTM never builds a happy schedule. The distance between the customer target and the realistic finish is the engine of before-the-fact urgency. Hiding the gap does not shrink it; it moves the discovery to a date when nothing can be done about it. Best practice: calculate the critical path honestly, publish the gap, and use it to justify the resources, decisions, and parallel paths that close it. A schedule that matches the target on day one has been reverse-engineered from the answer.

A3 · Isolate the hard 10%

Every first-of-a-kind product carries a small set of problems nobody has solved before. Left inside the product schedule, they leak into every meeting, consume the leaders, and stall the routine 90%. Best practice: pull them into their own project-managed schedule with a dedicated core team of SMEs who wake up thinking about nothing else. Refresh twice a week. Give the team everything it asks for, plus executive visibility. Solutions come from focused people, not from large meetings.

A4 · Plan the learning cycles

NPD work iterates; that is what makes it NPD. A schedule with one pass through design, build, and test is a prediction that everything works the first time, which has never happened. Best practice: rate each technical problem easy, medium, or hard, then budget roughly two cycles for easy, three or four for medium, five or more for hard, using known DOE and build-and-test cycle times. An educated guess with a method beats a schedule that pretends. Put a worst-case placeholder on every unknown so the plan stays closed-loop, then pull it in as reality improves.

A5 · Pull the pain forward

A problem exposed early is cheap. The same problem exposed late is existential. The only variable leadership controls is the date of exposure. Best practice: build an environment where surfacing a gap is rewarded and hiding one is the offense. The general manager sets this personally, because the first reaction to bad news determines how much bad news reaches you afterward. In a culture with a history of punishing the people who stepped out, protection has to be visible and repeated before behavior changes.

A6 · Expose the resource problem with data

"We need more people" is usually wrong in an interesting way: the real constraint is a few key people with key skills. Best practice: make every PDT leader answer three questions with names and numbers. How many people to accelerate the schedule? Which skills are missing today? What would "everything you need" look like? The answers turn an unanswerable complaint into a fundable request. Then fund it fast; the speed of provisioning is itself the signal that excuses are being removed.

A7 · Remove the excuses, then hold the fire

Accountability without provisioning is unfair. Provisioning without accountability is charity. FTTM does both, in strict order: first remove every named excuse, in writing, so nothing is left to point at when the date arrives. Then hold delivery accountability without exception. Teams that have been counting on their excuses will test whether you mean it. The dashboard matters here: unfiltered instruments are what let you tell a real obstacle from a rehearsed one.