Case study

Innovation Accelerated

advanced display fab and technology development

Three programs in one, delivered through COVID

Advanced semiconductor program

A research-grade process, transferred to a fab that did not exist at kick-off, accelerated to a breakthrough yield target on a reset three-year timeline — through a pandemic, on one core team colocated at the fab.

Prepared by

lateralworks
Technical program management

Program

Advanced display device
Five-year duration; lateralworks engaged 3+ years

Online

lateralworks.com
Case study series

© 2026 lateralworks

Table of contents

Innovation Accelerated

Core thesis. You cannot accelerate a program you do not see. The first job was making the program visible end-to-end — one integrated plan, one core team, one refresh cycle — and then closing the gap one critical-path week at a time.

© 2026 lateralworks

Overview

Abstract

A consumer-electronics client was building a next-generation product that required thirty-eight discrete inventions to come together at one time. The advanced display device was one of those inventions — and on its own, three programs run as one.

The base technology had been demonstrated in a European research institute. Standing it up at production scale meant first building the fab that would do the research, then doing the research, then accelerating that research to a breakthrough yield, performance, and cycle-time target the product team's insertion milestones would not bend on. Phase 1 was a brownfield fab retrofit — more than one hundred tools spec'd, ordered, installed, hooked up, and qualified, the bulk of it executed through COVID lockdowns. Phase 2 was a research-to-development process transfer for all three color emitters. Phase 3 was a worldwide engineering acceleration to tune the fab and the process to the breakthrough target. The same heavyweight core team owned all three.

lateralworks ran the technical program management end-to-end, with the program lead colocated at the receiving fab for the full duration of the engagement. The first lateralworks deliverable was telling the client the original eighteen-month timeline was wrong. The second was a reset to a three-year plan grounded in honest learning-cycle accounting. The rest was the discipline of working that plan one critical-path week at a time, through a pandemic, inside a multinational with deep pockets, CEO-level commitment to the product, and many technical stakeholders pulling on the team.

The team hit the revised target. The breakthrough technology was delivered. This case study describes how the program was framed, structured, and managed — and the decisions that mattered most along the way.

© 2026 lateralworks

01

Three programs in one

Three programs, one core team

The product behind this engagement was a consumer device that required thirty-eight discrete technology inventions to converge on a single launch window. Each invention was a program in its own right. This case study covers one of them: an advanced display device that combined a novel emissive front-plane bonded onto custom CMOS driver wafers — a stack the industry had not produced at scale, and a process the client did not yet own.

On its own, this single invention was three programs run as one. That framing matters more than any single technical fact in the case study, because every structural decision in the engagement followed from it.

© 2026 lateralworks

Three phases, one program, one team

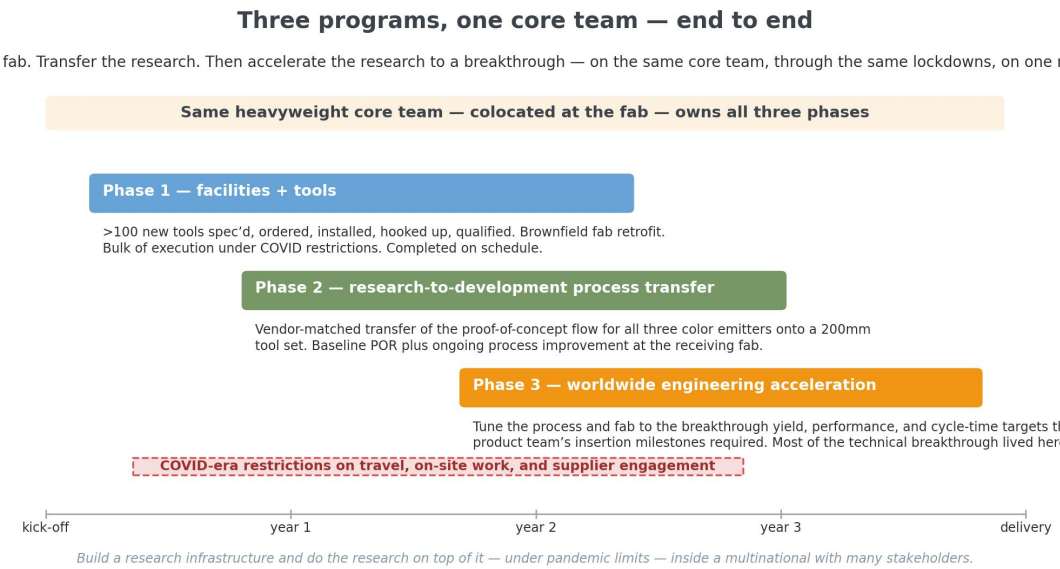

Figure 1. The three programs run as one. Phase 1 built the infrastructure (fab retrofit and tool buildout). Phase 2 transferred the research-grade process from the originating institute to the new development line. Phase 3 accelerated that process to a breakthrough yield target. The same heavyweight core team, colocated at the receiving fab, owned all three phases end-to-end — through the pandemic.

Phase 1 — facilities and tools. The receiving site was a brownfield fab. It needed more than one hundred new tools spec'd to the transferred process, ordered, hooked up to a sub-fab that did not yet exist for this technology, installed, and qualified. The facilities work — utility runs, gas and chemical distribution, layout for development followed by production volume — ran as a parallel critical path against the tool deliveries. The bulk of this phase executed through COVID-era travel restrictions, supplier on-site limits, and reduced fab staffing. It was completed on schedule.

Phase 2 — research-to-development process transfer. The originating institute's proof-of-concept flow for all three color emitters had to be vendor-matched onto a 200mm tool set the receiving site had never used. Every step of the flow needed a complete process and integration record, mask data, and FMEAs in place before transfer. The transfer was structured so that what arrived at the receiving fab was a process the receiving team could actually own — not a recipe they would have to reverse-engineer. A parallel track of process improvement at the receiving fab ran alongside the baseline transfer; the two were managed as one technical program with shared accountability.

Phase 3 — worldwide engineering acceleration. With the transferred process running on the development line, the work shifted from delivery of infrastructure to acceleration of learning. The product team's insertion milestones required a breakthrough in yield, performance, and cycle time that no single team had ever achieved on this technology. The engineering effort was worldwide, with technical contributions from the originating institute, the receiving fab, the client's design organization, and equipment and materials suppliers on three continents. Most of the technical breakthrough lived in this phase. The fab itself had to be tuned to world-class cycle-time efficiency so the engineering team could run enough learning cycles per quarter to make the breakthrough achievable inside the schedule.

The decisive structural choice was that all three phases ran on the same heavyweight core team, with a single program leader colocated at the receiving fab, from kick-off through risk-production start. The team

© 2026 lateralworks

that ordered the tools also ran the process transfer also drove the acceleration. There was no handoff. There was one program.

Why this got harder — COVID and multinational complexity

Two factors outside the technology made this program harder than other semiconductor transfers of comparable scope.

First, the pandemic. Phase 1 executed almost entirely under COVID restrictions. Equipment suppliers could not travel for factory acceptance tests. Tool installers needed visa exemptions to enter the country. The fab itself ran reduced on-site staffing for months at a time. Phase 2 and the start of Phase 3 ran under the same conditions. The schedule absorbed all of this without slipping the Phase 1 commitment date. It absorbed it because the core team had built a planning system that surfaced consequences early and treated supplier interfaces as live, jointly-planned tasks rather than checkpoint hand-offs.

Second, the organizational geometry. The client was a large multinational with CEO-level commitment to the product and the resources to back that commitment. That is an unmitigated advantage in some respects. It also meant the program had more technical stakeholders than any single core team could process if they had been treated as decision-makers: multiple design organizations, several in-house process and packaging teams, a research institute on a different continent, and the receiving fab as a separate legal entity. Ownership and boundaries were genuinely difficult to draw. The thing that made it work was that the program had one boss, colocated at the fab, running one core team on one mission. Stakeholders got information and a clear interface; they did not get a seat on the team.

The project mission, stated and acted on from the first week: accelerate the client's learning on a development line of its own, by failing faster. On a technology this immature, failing faster was the only way to reduce risk inside the schedule.

Three intertwined problems — administration, facilities, technology

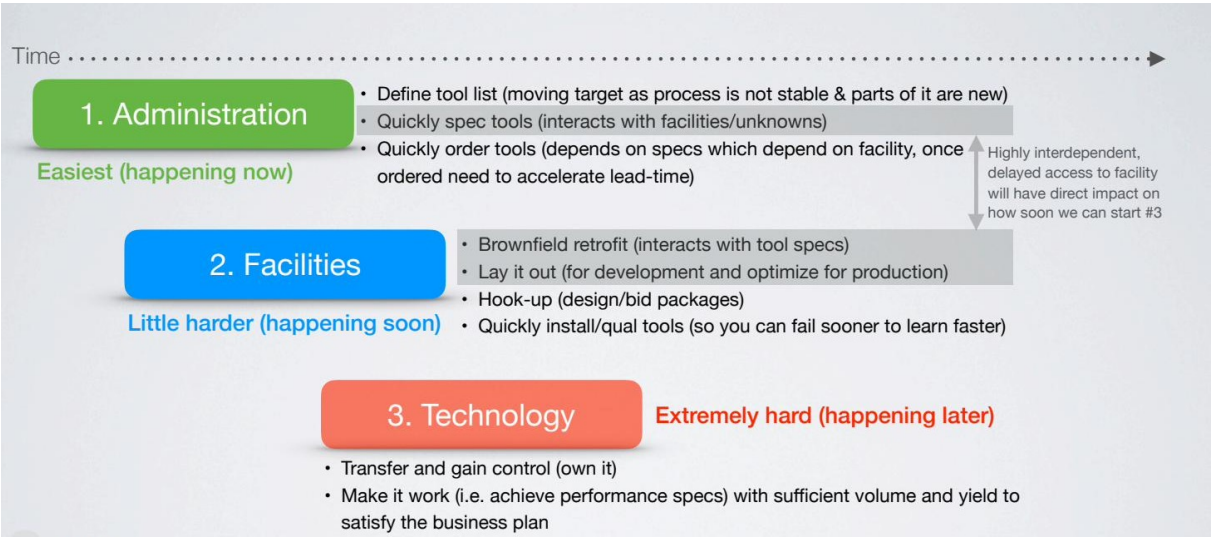

Figure 2. The three problem clusters that defined the early plan, in order of difficulty. Each layer depends on the one above it. A delay anywhere upstream compresses the time available for the hardest problem.

At kick-off lateralworks framed the program around three problem clusters that had to be solved in sequence but were tightly coupled. Administration was the easiest because it was about decisions the client could

© 2026 lateralworks

make immediately: define the moving target tool list, spec the tools, negotiate the orders. Facilities was the next layer up, harder because it depended on a brownfield site that did not yet exist as a development fab — utilities, layout, hook-up, install-and-qual sequencing all interacted with tool specs. Technology was the hardest and last in order because it depended on the first two: transfer the process, gain control of it, then make it work at the volume and yield the business plan demanded. A delay anywhere upstream compressed the time available for the hardest problem.

Why the original eighteen-month plan was wrong

The early commitment to an eighteen-month timeline was a planning artifact, not a technical forecast. It set the date the product team needed and worked the development plan backwards from it. The plan did not include the number of learning cycles the process actually needed to mature, because the maturity of the transferred process was over-stated and the rate at which a new development line could learn was over-estimated. The result was a gap between target and reality that grew, week after week, with no mechanism in the planning system to surface why.

The first real win of the engagement was getting that gap on the table. A program that thinks it is six months late will negotiate; a program that has not yet accepted it is years late cannot.

© 2026 lateralworks

02

Scope

The engagement, end to end

The scope of the engagement spanned the lifecycle of a semiconductor development program — from selecting where it would happen, to running the core team that delivered it. The work fell into seven streams that ran in parallel and reinforced each other.

© 2026 lateralworks

Selecting a foundry codevelopment partner

The client needed a partner that already had the silicon fabrication capability, clean-room infrastructure, and operating discipline to absorb the research-line process and run it at low-volume manufacturing scale. lateralworks structured the partner search, evaluated candidates against the program's technical, commercial, and program-execution criteria, and supported the negotiation of the master agreement that anchored the codevelopment relationship.

Transferring the process from the research institute

The transfer was not a "copy exact." It was a vendor-matched transfer of the proof-of-concept flow for all three color emitters, onto a 200mm tool set, with the receiving fab purchasing tools that did not exist in the originating site. Every step of the flow needed a complete process and integration record, mask data, and FMEAs in place before transfer. lateralworks coordinated the transfer plan with technical leads at both the originating institute and the receiving fab so that what arrived was a process the receiving team could actually own.

Expanding, retrofitting, and standing up the development fab

The receiving site was a brownfield fab. It needed more than one hundred new tools installed and qualified to support the transferred flow, plus the facilities infrastructure to support those tools — utility runs, gas and chemical distribution, sub-fab build-out, layout optimized for both development and the production volume that would follow. lateralworks structured the facilities work as a parallel critical path tied to the tool ordering schedule, and made the tool list itself an active planning object — refreshed twice a week, costed, and visible to every function that depended on it.

Developing the process to product KPIs

Five process super-modules — coring, bonding, planarization, pixelation, post-pixelation — each had to be qualified individually and then together as a full flow. Qualification at the partner fab meant the module produced material that met the defined performance, yield and cycle-time targets handed down from the product KPIs. lateralworks held the master plan that tied module-level qualification milestones to the product team's intercept points and refreshed the gap analysis weekly.

Defining, structuring, and managing the program — the TPM role

For more than three years lateralworks served as the strategic technical program manager for the development line. That meant owning the program's planning architecture, running the core team, escalating the right issues to executive sponsors at the right time, and protecting the team's ability to focus on the hard technical work by absorbing the program-management load.

Forming and chartering the core team

The core team spanned the client's process and product organizations, the foundry partner's development and manufacturing teams, the originating European research institute, multiple external equipment and materials suppliers, and an industrial-engineering services partner that handled facilities and capacity modeling. lateralworks chartered the team, defined its operating cadence, and held it to the discipline of working as one team rather than as a federation of organizations that occasionally met.

© 2026 lateralworks

One integrated master schedule, end-to-end

The single most important artifact of the engagement was a master plan that connected every work stream end-to-end and refreshed twice a week. From it lateralworks ran an aggressive acceleration program — pull-in actions identified every week, executed by the team that owned them, tracked against trend, escalated on exception. The acceleration program closed a multi-year gap to a realistic and committed schedule the team then delivered against.

© 2026 lateralworks

03

Organization

Core team, charter, and a single point of ownership

A semiconductor program of this complexity fails when the people doing the work report up through separate organizations and meet only to share information. lateralworks observed the textbook failure pattern on a parallel program inside the same client — a group of functional representatives with no shared ownership of outcomes, each function trying not to be the one labeled as the critical path. Patrick Lencioni's five dysfunctions of a team — inattention to results, avoidance of accountability, lack of commitment, fear of conflict, absence of trust — were all present.

The development line could not be run that way. The fix was structural.

© 2026 lateralworks

A heavyweight core team, not a steering committee

The team was five to ten people with complete authority over program outcomes. Each member represented a function — facilities, tools and process, transfer and research, infrastructure, procurement, development operations, product interface, industrial engineering — and each member could commit on behalf of that function in real time. The team owned the integrated schedule, owned the weekly refresh, owned the critical path, and owned the actions to close the gap. They identified blocking issues and brought resources to resolve them. They were not a reporting body; they were the program.

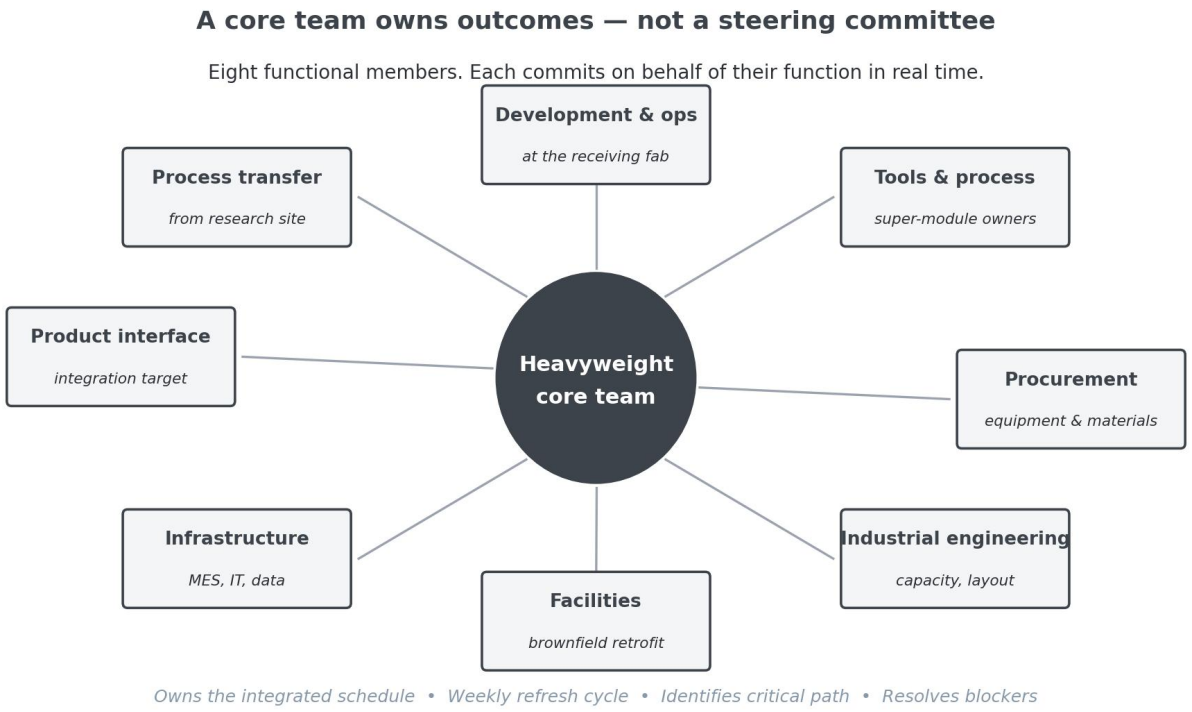

Figure 3. The heavyweight core team. Eight functional members, each able to commit on behalf of their function in real time. The team owns the integrated schedule and the actions to close the gap.

One leader, colocated at the fab, one mission

Heavyweight team structures fail when the leader runs them remotely. The lateralworks program lead was colocated at the receiving fab for the duration of the engagement. Decisions made in the morning could be tested on the floor by the afternoon; equipment suppliers visiting for factory acceptance or hook-up walked into the same building as the program office. Colocation cost personal time. It bought speed nothing else could buy.

The core team was 100 percent dedicated to this program — not matrixed, not shared with two other projects, not borrowed for a quarter. Functional members were resourced full-time. The receiving fab's engineering organization ran as a paired partner team to the core team, with one interface and joint ownership of outcomes. There was one mission, and one team executing it. This single structural choice did more to make the program work than any planning technique in this document.

© 2026 lateralworks

Three roles fast programs need defined

Programs that move fast need three roles defined precisely, and they need to be different people. lateralworks worked with the client to name them, staff them, and protect the distinction between them.

The integrator looks laterally and ties the work of every function together. The integrator does not own technical content; they own the connections between technical contents.

The architect makes system-level trade-offs that do not compromise overall system performance. When a tool slips a week and three sub-modules need new sequencing, the architect decides what holds and what gives.

The chief engineer makes the final technical decision. When the integrator has surfaced a question and the architect has framed the trade-off, the chief engineer says yes or no and the team moves.

Programs that try to merge these roles — integrator and chief engineer in the same person, or architect and integrator — slow down. The three modes of thinking conflict.

One technical voice, one strategic program voice

The program designated a single chief engineer with technical authority over device integration and compliance with system-level performance requirements, and a single strategic TPM responsible for delivery from process transfer through the long-horizon product milestones. Both roles answered to the executive steering committee but neither was a committee. When trade-offs came up — and they came up constantly — the program knew where the decision lived.

Engineering-driven planning

lateralworks insisted that the integrated schedule be developed by the engineering subject-matter experts together with the TPM, not by the TPM alone. Schedules built by a program-management function without engineering ownership are accurate the day they are published and irrelevant the day after. Schedules built by the engineers who will execute them belong to the team and survive contact with reality.

Suppliers as joint planners

Equipment and materials suppliers were brought into joint planning sessions and ran on transparent schedules that included both their tasks and the client tasks each interface required. This was a departure from the more common pattern of checkpoint-only supplier engagement, in which the program only learns about a problem after the problem has occurred. With the joint-planning model in place, suppliers were a forward-looking part of the schedule rather than a source of unwelcome surprises.

The cross-functional FTTM model

The product was a system, and the technology programs that fed it were each their own system. Both layers ran on the same heavyweight core-team pattern — small empowered teams, single point of decision authority, one operating cadence. The lateral interfaces between the layers were held by two named roles: the product architect at the system level and the process architect at the technology level. The corporate host provisioned, decided fast, and stayed out of each team’s way.

© 2026 lateralworks

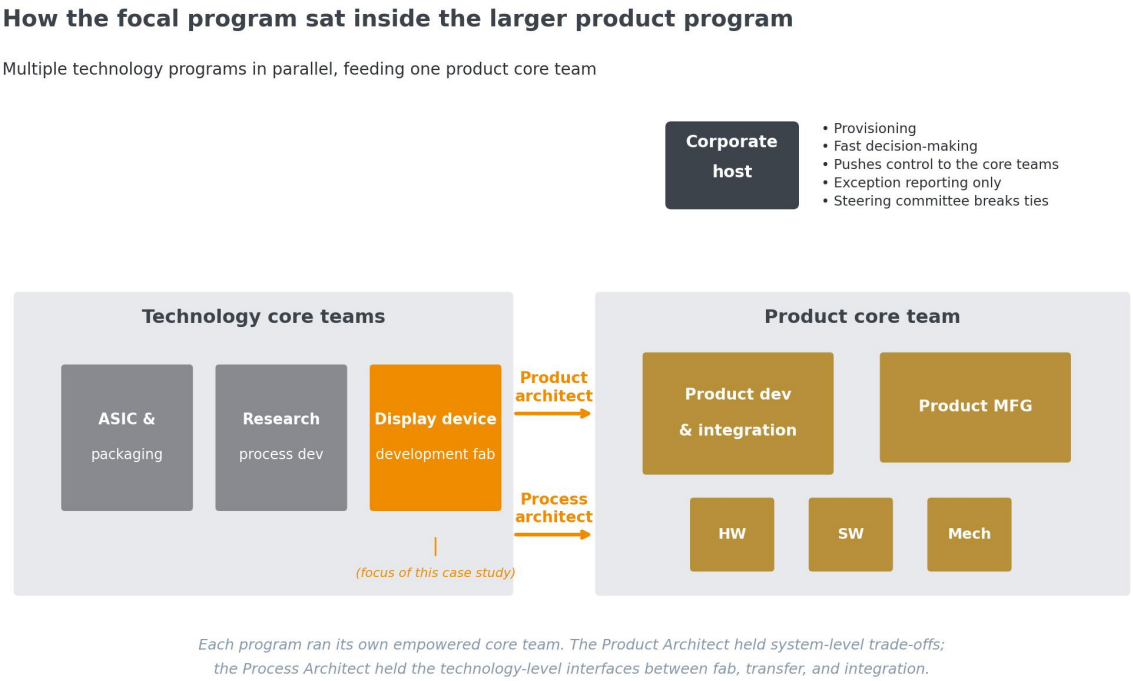

Figure 4. The cross-functional team structure across both layers. Several technology programs ran in parallel — the focal display-device development fab being one of them — each feeding into the product core team. The Product Architect held system-level trade-offs; the Process Architect held the technology-level interfaces between fab, transfer, and integration.

© 2026 lateralworks

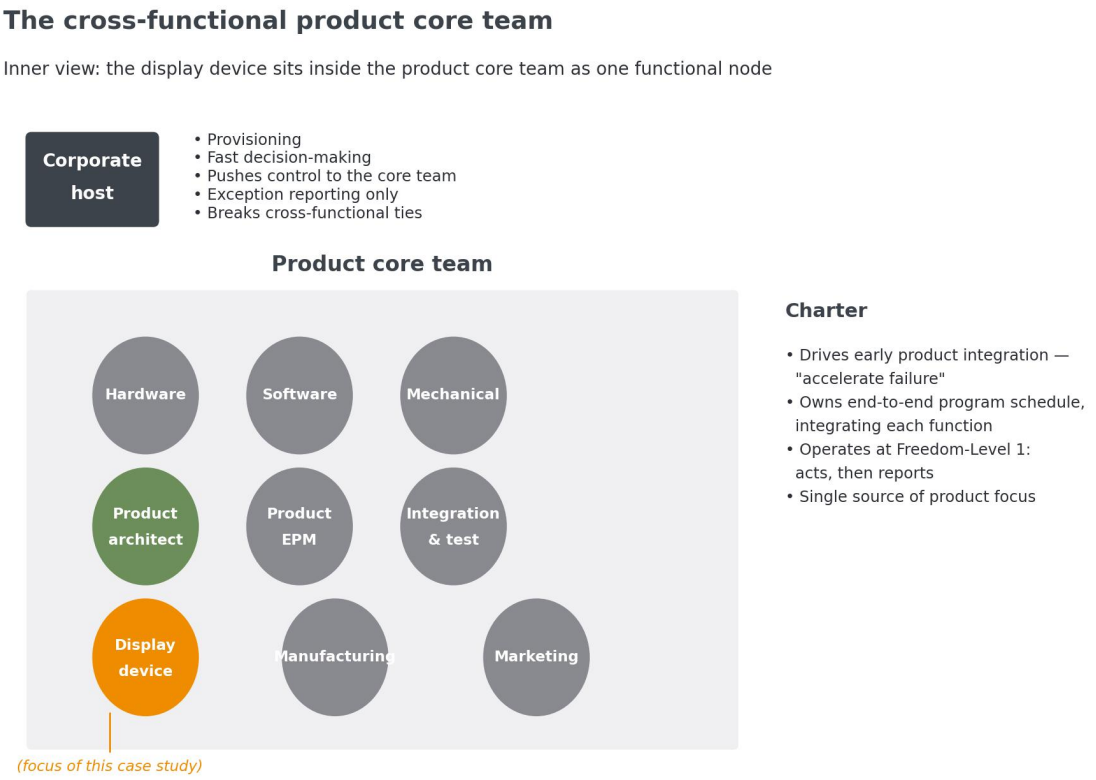

Figure 5. The inner view of the product core team. The display device was one node among Hardware, Software, Mechanical, Manufacturing, Marketing, Integration & Test, Product Architect, and Product EPM. Each technology team mirrored this same pattern internally.

© 2026 lateralworks

04

Planning architecture

One macro plan, many micro plans, twice-weekly

The planning architecture was the operating system of the program. Every major decision — pull-in action, supplier negotiation, tool order, scope trade-off — ran through it.

© 2026 lateralworks

Data architecture, all driven from the critical-path schedule

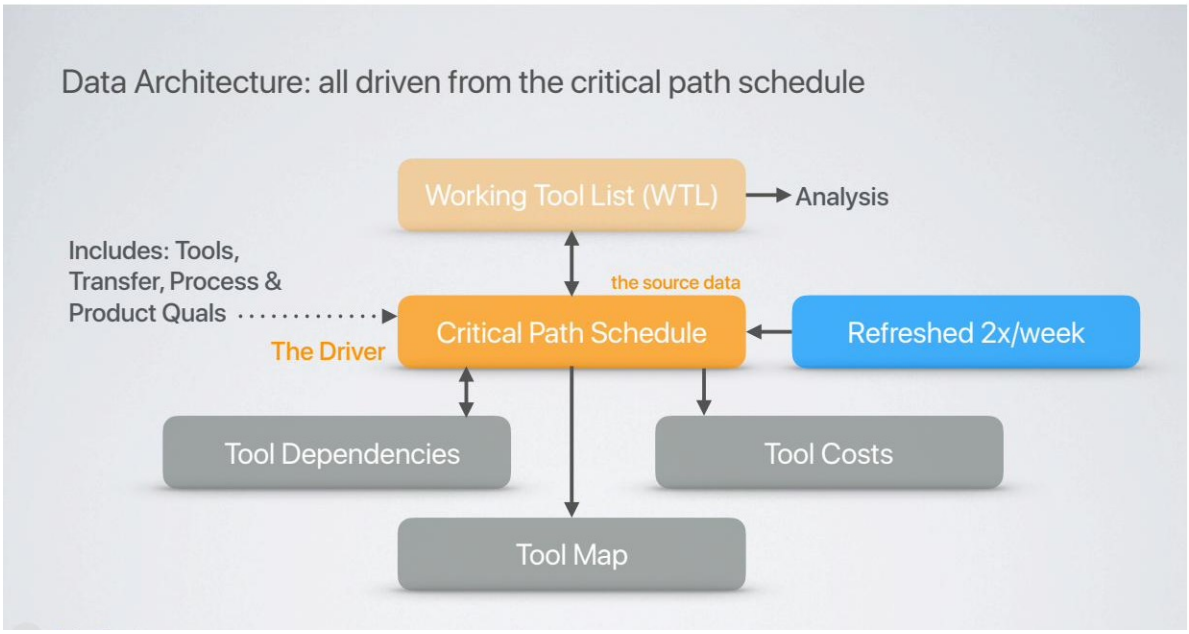

Figure 6. Planning data architecture. The critical-path schedule is the master record. The working tool list feeds in and receives data back. Tool dependencies, costs, and the tool map sit alongside and refresh twice a week.

The critical-path schedule was the master record. The working tool list — the program's tool POR — fed into the schedule with names, suppliers, costs, and dependencies, and received dates and the refreshed critical path back out. The tool map showed which tools were needed for which super-module and sub-module qualification, with explicit tool-to-tool dependencies that propagated through the schedule. Cost data was prorated over the PO-placed, docked, and qualified milestones for every tool, so the program's burn rate tracked alongside its critical path. Refresh frequency was twice a week, not once a quarter.

Macro-micro schedule architecture

The master schedule was the macro view: program-level critical paths to the major milestones — first prototype delivery, all modules qualified, risk production start. Beneath each critical path sat a micro schedule for the work-stream owning that path. Each micro schedule was refreshed weekly by the technical DRI and TPM responsible for it, with rollup into the macro on the same cadence. The result: a program with a single answer to "where are we?" at any level of zoom.

© 2026 lateralworks

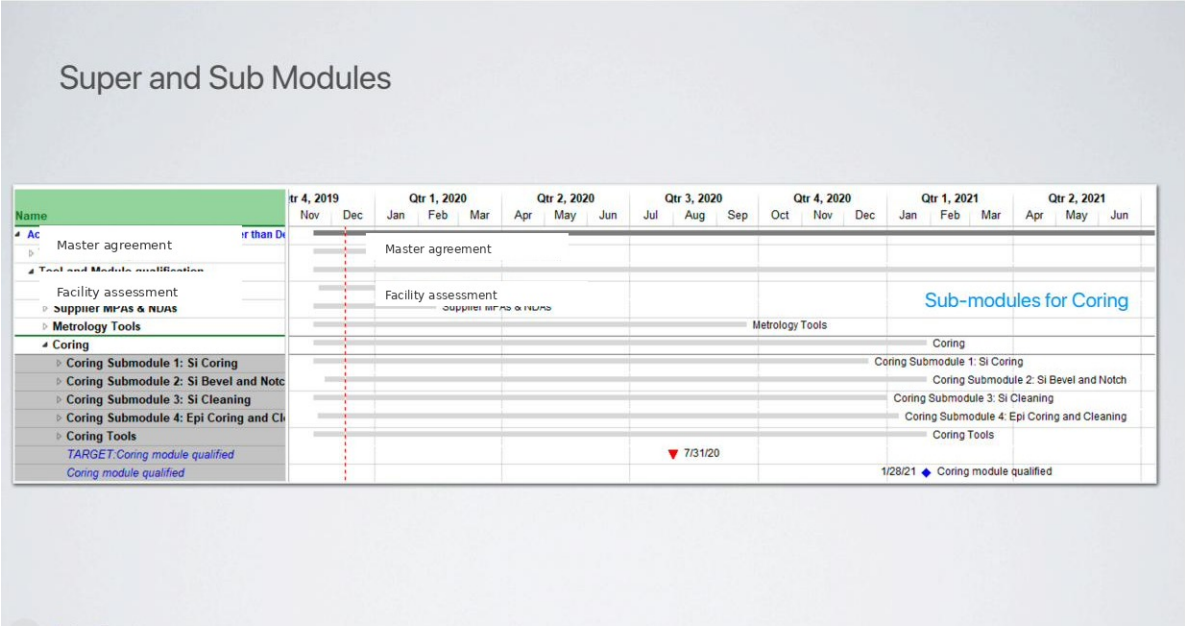

Figure 7. The macro-micro WBS structure. Top: the program-level view (master agreement, facility assessment, tool and module qualification, process and product qualification). Bottom: sub-modules under a single super-module, with target and predicted dates and the qualification milestone.

The refresh process

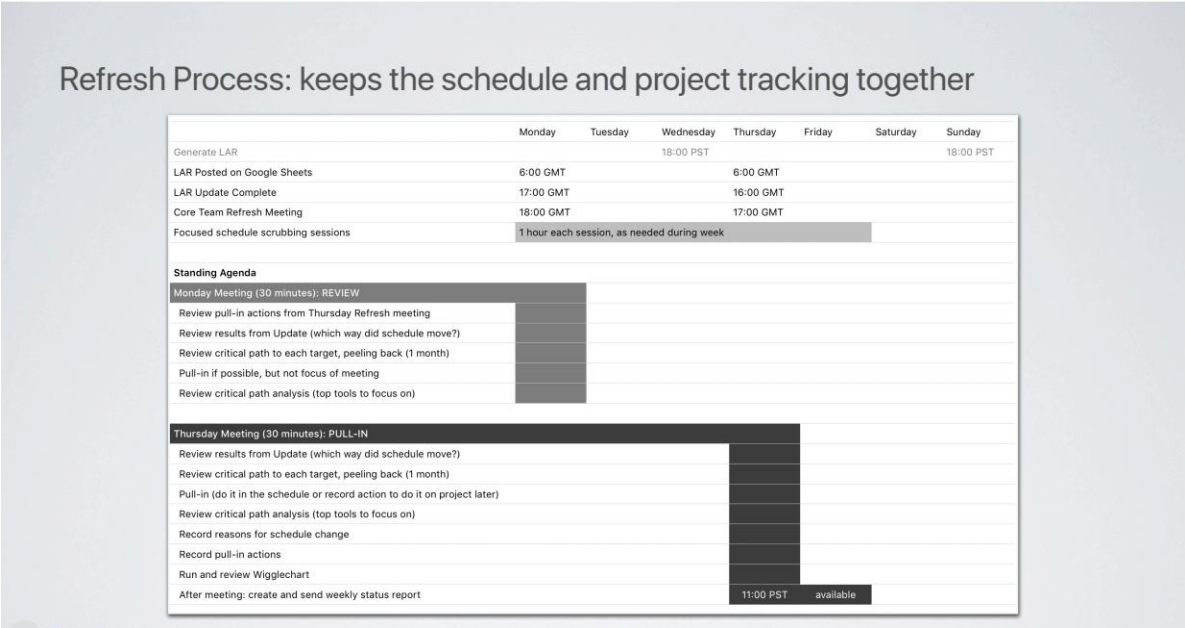

Figure 8. The weekly refresh cycle. A Monday review of pull-in actions and a Thursday session for the next week's acceleration moves. Status reporting is a byproduct of the planning process, not a parallel activity.

The week had two anchor meetings: a Monday review of pull-in actions taken since the previous Thursday and a Thursday pull-in session where the next week's acceleration moves were identified and committed.

© 2026 lateralworks

Between the two, the team conducted focused schedule-scrubbing sessions on specific work streams. The cadence was deliberate. Reviews and pull-ins were separated so that one meeting was not asked to do both jobs. Status reporting was a byproduct of the planning process, not a parallel activity.

Tool dependencies and the move from tool list to schedule

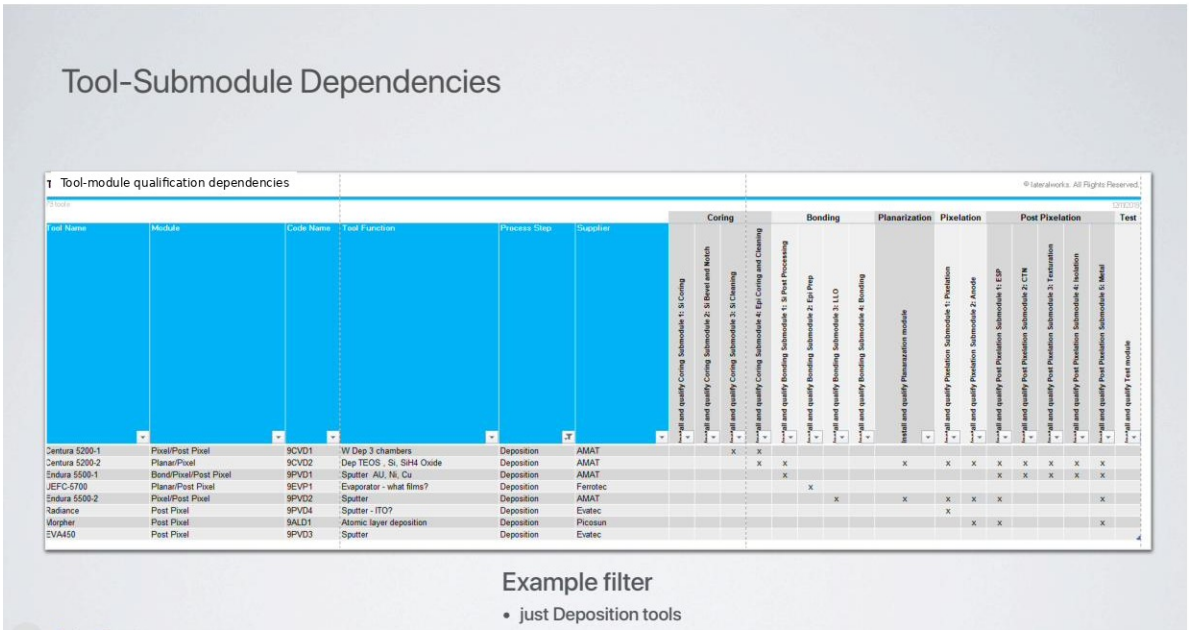

Figure 9. Tool-submodule dependency map (filtered example showing deposition tools only). Each tool feeds into the qualification of specific sub-modules; when a tool slips, the planning system propagates the consequence downstream automatically.

The tool-to-tool and tool-to-submodule dependency tables propagated changes through the schedule automatically. When a single tool slipped — qualification delayed at the supplier, or a sub-fab utility hook-up running late — the schedule showed the downstream submodule, then the module, then the full-flow qualification, then the prototype delivery, in one connected view. The team did not have to chase the consequences of a slip; the planning system did it for them.

© 2026 lateralworks

Turning point

Reset the plan

You cannot accelerate a program until the program agrees on where it actually is.

Engagement reflection

On resetting the development-line target from 18 months to 3+ years

© 2026 lateralworks

05

Closing the gap

Working the critical path, one week at a time

The acceleration program existed because the gap to the original target was enormous and the gap to the revised target was still aggressive. Acceleration was not a one-time exercise; it was the operating mode of the program for the full duration of the engagement.

© 2026 lateralworks

Force acceleration of each step — it is all connected

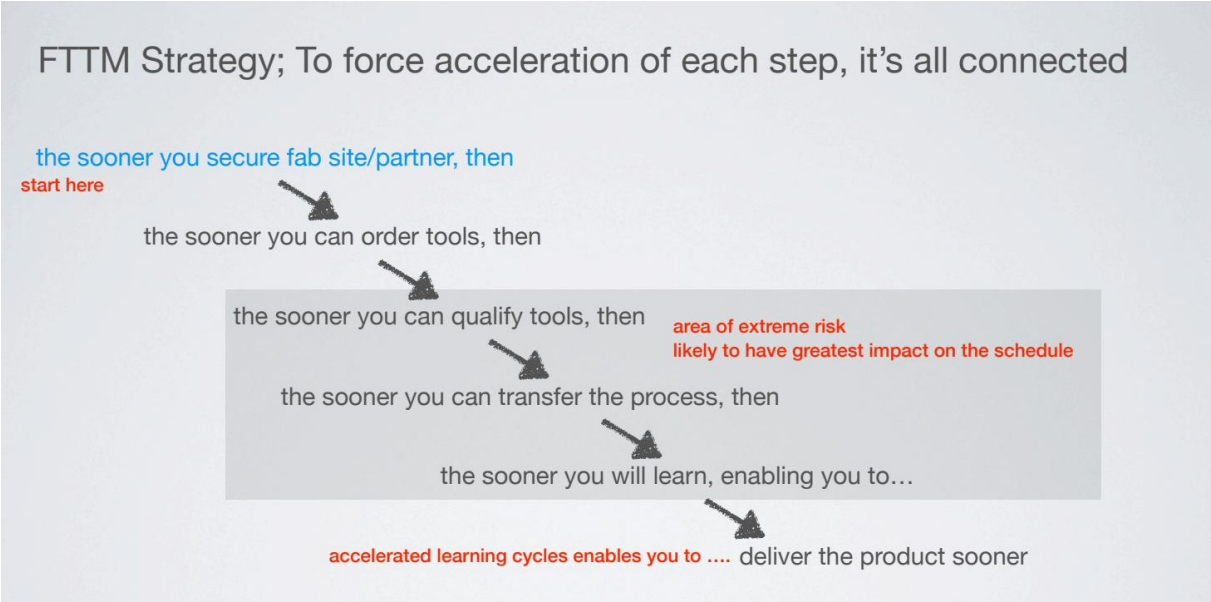

Figure 10. The FTTM acceleration cascade. Each step compresses the next. The area of extreme risk — process transfer and learning at the receiving fab — sits in the middle of the chain. Acceleration upstream buys time for the hardest work downstream.

The acceleration logic ran backwards from the product milestone. The sooner the fab was secured, the sooner the tools could be ordered. The sooner the tools were ordered, the sooner they could be qualified. The sooner the tools were qualified, the sooner the process could be transferred. The sooner the process was transferred, the sooner learning began. And accelerated learning was the only mechanism that delivered the product sooner. Each step compressed the next, and the area of extreme risk — process transfer and learning at the receiving fab — sat in the middle of the chain. Every week the program asked: what can we pull in upstream of that risk zone?

Pull-in actions on the critical path

© 2026 lateralworks

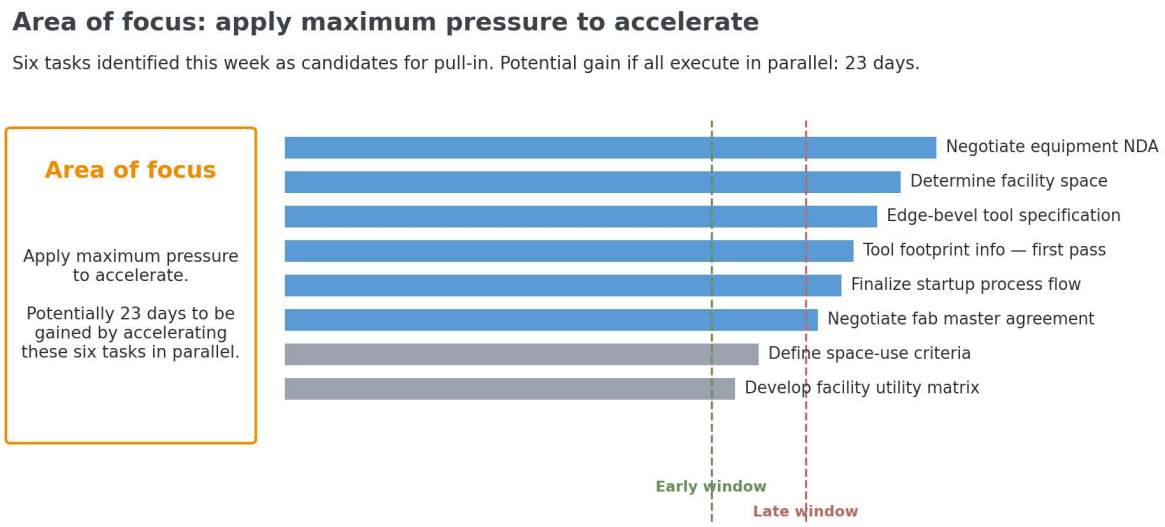

Figure 11. Weekly area-of-focus pull-in chart. In one representative week the program identified six tasks worth twenty-three days of pull-in if executed in parallel. Each task has an owner, a target date, and a current finish trend. The team negotiates execution, not value.

Every weekly status report opened with the area where maximum pressure was being applied. In one representative week the program identified a cluster of tasks worth twenty-three days of pull-in if executed in parallel. The tasks were specific — negotiate a single NDA, finalize a startup process flow, obtain preliminary tool footprint information, develop the facilities utility matrix — and each had an owner, a target date, and a current finish-date trend. The team did not negotiate the value of the pull-in. They negotiated whether they could execute the plan that delivered it.

Peeling back the critical path

© 2026 lateralworks

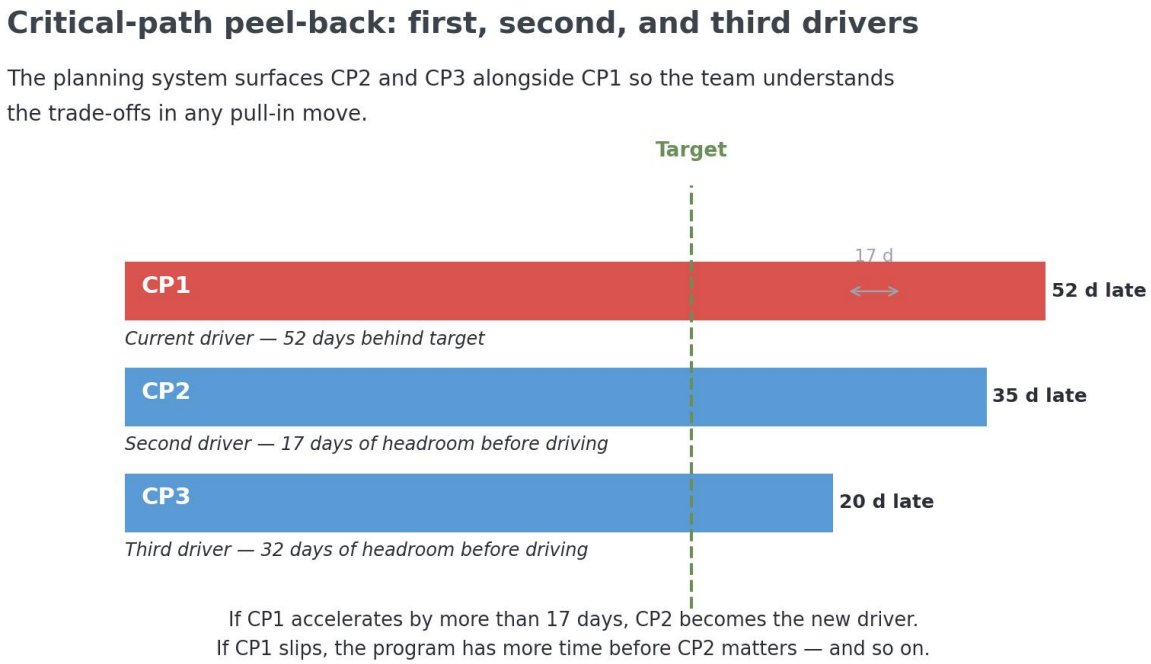

Figure 12. Critical-path peel-back. The planning system surfaces CP2 and CP3 alongside CP1 so the team understands the trade-offs in any pull-in move. If CP1 accelerates by more than the gap to CP2, CP2 becomes the new driver.

The first critical path is rarely the only critical path. The planning system surfaced the second and third critical paths as well, so the team understood the consequences of acting on the first. If the first critical path accelerated by more than the gap to the second, the second became the driver. If the first slipped, the program had more time before the second mattered. The peel-back view turned acceleration from a single-track activity into an informed sequence of trade-offs.

Honest learning-cycle accounting

The hardest planning decision the program made was to include the learning cycles each team actually needed to solve the known technical problems — not the cycles the original schedule had budgeted. Doing this widened the gap to the build-target milestones in the short term. It was the right call. The widened gap defined what the acceleration program needed to close; the previous "build schedule" approach had hidden the real gap behind whatever happened to be ready on the appointed day.

© 2026 lateralworks

06

Results

Breakthrough technology, delivered on the reset plan

The development line delivered the breakthrough technology on the revised three-year target. The original eighteen-month plan was unachievable; the team's first job was demonstrating that. Once a realistic target was in place, the acceleration program worked: the team closed the gap from the new baseline and hit the milestones it had committed to.

© 2026 lateralworks

What "delivered" meant

The development line produced functional prototype wafers from back-to-back lot runs at a four-week cycle time, with performance specs meeting targets across all three color emitters. First-generation yield and performance data was available to the product team on schedule, and risk production started against the year-over-year roadmap with the eighty-percent yield and performance targets in hand.

What "delivered" did not mean

The case study would be dishonest if it framed the outcome as smooth. It was not. Process maturity at transfer was lower than expected; the receiving fab needed time and management changes to reach the learning rate the program required; partner-relationship dynamics introduced friction that no planning system can eliminate. lateralworks documented these issues honestly to the client throughout the engagement, including hard recommendations to invoke step-in clauses and replace partner leadership when the rate of improvement at the fab fell below what the program needed. Some of those recommendations were taken; some were not. The team’s ability to deliver against the revised plan owes a lot to the recommendations that were taken.

What the planning system left behind

At handoff the client had a single integrated master schedule with weekly refresh discipline and explicit critical-path management; a working tool list driving the schedule, with cost and dependency data live; a chartered core team with clearly defined roles, operating cadence, and decision authority; a planning architecture demonstrated in production for more than three years; and a learning-cycle-aware approach to setting milestones that the product organization could apply to the rest of the inventions inside the larger program.

© 2026 lateralworks

07

Reflections

Three things that made it work

Three patterns explain why the program landed where it did, and they travel beyond this particular semiconductor engagement.

© 2026 lateralworks

Three programs, one core team, end to end

The single most consequential structural choice was that the same heavyweight core team owned the engagement across the facilities-and-tools phase, the process-transfer phase, and the engineering-acceleration phase. Most semiconductor programs hand off between phases — the facilities team finishes and goes home, the transfer team brings the process in and leaves, the engineering team takes over to qualify. Every handoff costs context, momentum, and the institutional memory of why earlier decisions were made. This program had no handoffs. The team that ordered the tools also ran the transfer also drove the acceleration. The cost of running a stable team for three-plus years was real. The cost of not doing it would have been far higher.

One leader, colocated, on one mission

The development line had a steering committee — the right one for the program, with the right authority — but the program ran on a core team that owned outcomes, owned the schedule, and made decisions inside its remit without waiting for the next escalation forum. That core team worked because it had one boss colocated at the receiving fab, full-time, for the duration; because every functional member was 100 percent dedicated to the program; and because the receiving fab's engineering organization ran as a paired partner team with one interface and joint ownership of outcomes. One team. One mission. Inside a multinational with deep pockets and many stakeholders, this is what difference looks like.

Make the gap visible — then live in the planning system

Acceleration is impossible until the program agrees on the size of the gap it is accelerating against. The most consequential analytical work in the first months of the engagement was not the acceleration plan; it was the honest gap analysis that made the acceleration plan defensible. The original eighteen-month plan was not lying — it was answering the wrong question, namely "what date does the product team need?" The right question was "given the technical maturity of the transferred process and the rate at which the receiving fab can learn, when can we credibly deliver?" The two answers differed by years. Naming that difference is what unlocked the rest.

The planning system that surfaced the gap then closed it. The macro-micro schedule with twice-weekly refresh, the tool-list and dependency tables, the critical-path peel-back, the Thursday pull-in cadence — none of it works as a deliverable. It works as an operating system the team uses every day. lateralworks did not hand the client a schedule; lateralworks ran the schedule alongside the team for three years, demonstrating what its weekly use looked like, until the planning architecture was theirs. The client's program-management team carried it forward from there.

The breakthrough technology was delivered. A research-grade process is now in production at a fab that did not exist three years earlier, inside a program system that the client has operated since handoff. The case study covers one of the thirty-eight inventions that had to converge for the larger product to ship — three programs run as one team, one mission, through a pandemic. It is also a working illustration of what fast time to market looks like when the structural choices are right at the start.

© 2026 lateralworks