

COREFORCE FOR ENGINEERING TEAM

Engineering Collaboration Playbook

Stopping the Noise. Protecting Your Deep Work.
A Structured Request System Powered by NotebookLM and Notion.

PREPARED BY

Jayson Forde

Client Success Operations Strategy and Infrastructure

PRESENTED TO

Engineering Manager and Engineering Team

NotebookLM Query System

Async Engineering Queue

5-Stage Workflow

SLA Commitments

RACI and Authority

90-Day Roadmap

CONFIDENTIAL

ENGINEERING PLAYBOOK

CoreForce 2025

What Engineering Is Currently Experiencing

The documented operational issues coming from the Client Success and Account Management side

These are the documented issues identified through the discovery and interview process. They reflect the real friction that Engineering is absorbing because the Client Success department has not had a structured operating system in place. Each issue below has a direct solution in this playbook, and by Day 30 the infrastructure to resolve them will be live.

Rushed Commitments Made Without Engineering Validation

Account Managers are making commitments to clients about technical capabilities and timelines without first consulting Engineering. Clients receive promises that Engineering was never part of, and Engineering is then expected to deliver on them. This creates misaligned expectations, strains client trust when delivery does not match what was promised, and puts Engineering in the position of being the bearer of bad news for commitments they never agreed to. Under the new system, zero client-facing technical commitments are made without Engineering's written feasibility verdict first.

No Structured Query System Between CS and Engineering

There is currently no formal, trackable system for how the Client Success team submits questions and requests to Engineering. Communication happens informally, primarily through Slack, with no queue, no SLA, no priority classification, and no audit trail. This means Engineering cannot manage workload predictably, cannot hold CS accountable to a standard, and cannot point to a record of what was asked and what was answered. The Notion Async Queue resolves this entirely by creating a structured, tracked, SLA-governed pipeline for every CS-to-Engineering interaction.

Core Commitment: Engineering will never receive an unvalidated, context-free request from Client Success again.

NotebookLM: The Engineering Query Intelligence System

How AI-powered synthesis converts every client request into a structured, decision-ready packet before Engineering sees it

The single most powerful change this system introduces for Engineering is NotebookLM. Right now Engineering receives requests through informal channels, stripped of context and business rationale. NotebookLM acts as the intelligence layer between Client Success and Engineering. It absorbs raw client input and converts it into a structured, business-contextualized query that Engineering can evaluate and respond to without any additional information-gathering. It removes the noise and delivers only signal.

What is NotebookLM?

NotebookLM is Google's AI-powered research and synthesis platform. It ingests multiple sources of information including meeting notes, client records, account history, and prior Engineering responses, then uses AI to synthesize them into structured intelligence briefs. For this system it becomes the brain of the Client Success operation, translating raw client input into organized, decision-ready output that Engineering can act on.

Why This Protects Engineering

Engineering's most valuable resource is focused, uninterrupted time. NotebookLM protects that resource by ensuring every technical request has been pre-processed through an AI synthesis layer before Engineering sees it. The request arrives with the client's business goal explained, the account health included, the urgency pre-classified, and a clear specific question to answer. No back-and-forth. No context-gathering. Engineering reads the packet and issues a verdict.

How the NotebookLM Query System Works

Step 1 AM Captures	Step 2 NBLM Synthesizes	Step 3 CS Manager Reviews	Step 4 Engineering Responds	Step 5 CS Relays to Client
The AM completes a Structured Capture Form in Notion after every client interaction. This records the specific request, business goal, urgency, and known context. No documentation means no escalation.	The CS Manager submits the form into NotebookLM alongside the full account folder. NotebookLM generates a Decision-Ready Intelligence Packet: a structured brief with full business context, ready for Engineering to evaluate.	The CS Manager reviews the Intelligence Packet before it enters the queue. This confirms urgency classification is accurate and the question is specific. The CS Manager is the quality gate.	Engineering receives the packet in the Notion async queue. Clean, structured, no Slack noise. Engineering issues a verdict: Buildable Now, Roadmap Candidate, Out of Scope, or Needs PM Review.	The AM communicates the Engineering verdict using approved language templates. CS does not interpret or negotiate the verdict. The outcome is logged and the full request lifecycle is preserved in the audit trail.
Core Outcome for Engineering: Every request arrives pre-synthesized. Engineering reads intelligence, not noise. Their only job is the verdict.				

Notion: The Engineering Async Queue

How the Command Center organizes and governs every request Engineering receives from Client Success

NotebookLM generates the intelligence. Notion is where that intelligence lives, gets acted on, and gets tracked. The Notion Async Queue is Engineering's dedicated workspace, designed around how Engineering works. There are no real-time demands, no status-check pings, and no requirement to respond in a Slack thread. Engineering opens the queue on their schedule, reads structured packets, issues verdicts, and moves on.

Engineering Async Queue

Every Intelligence Packet routed to Engineering lands here. Each ticket displays the account, the specific query, urgency classification, SLA deadline, and the full packet. Engineering updates the verdict field and the CS Manager is automatically notified. No meetings. No Slack threads.

SLA Countdown Tracking

Every ticket carries a visible SLA timer. Standard requests show a 24-hour countdown. P1 Critical escalations show a 4-hour countdown. Engineering leadership can see at a glance how many tickets are within SLA and how many are approaching the deadline. Nothing hides.

Verdict Classification Panel

Engineering's response interface offers four options: Buildable Now, Roadmap Candidate, Out of Scope, and Needs PM Review. Each verdict triggers an automated notification to the CS Manager. Engineering adds optional notes but is not required to write a lengthy explanation.

Query Health Dashboard

A dedicated dashboard shows Engineering and CS leadership the full pipeline health: total open tickets, average response time, SLA compliance rate, and verdict breakdown over time. This is read-only for Engineering and exists so leadership can monitor health without status meetings.

Full Request Audit Trail

Every request from initial capture through final client response is preserved in the Notion account record. Engineering can reference prior verdicts, see how similar requests were handled, and track whether clients are re-submitting requests previously classified as Out of Scope.

Account Intelligence Access

Engineering has read access to the full account context for any ticket in the queue. If Engineering needs more background before issuing a verdict, the full CS account record, health score, and interaction history are available directly in Notion.

ZERO BACKDOOR POLICY: No Account Manager contacts Engineering directly outside the Notion queue. Every request enters the system or it does not move forward.

Engineering SLA Commitments from Client Success

Standard Technical Request	24-hour queue response	Full Intelligence Packet included with every submission
P1 Critical Escalation	4-hour response path	CS Manager personally confirms escalation before triggering
Feature or Product Request	PM-routed via formal intake	Engineering never receives product requests directly from AMs

Engineering Collaboration Playbook

Current state vs. new system: what Engineering absorbs today and what the new system delivers

With the NotebookLM Query System and Notion Async Queue in place, the Engineering team's interaction with Client Success becomes fundamentally different. Engineering is not being asked to do more. They are being asked to do their work with better context, formal SLA protection, and a guarantee that the three issues documented in Section 01 will not get through.

CURRENT STATE What Engineering Absorbs Today

Rushed commitments on Engineering's behalf

AMs confirm to clients that technical work is possible or will be delivered by a certain date without Engineering's input. Engineering learns about the commitment after the fact and must either deliver under pressure or absorb the client relationship damage.

Blind handoffs with no business context

Requests arrive through fragmented Slack threads with no account background, no explanation of the client goal, and no urgency classification. Engineering must gather all of that context before they can begin evaluating the request.

No formal query or tracking system

There is no queue, no SLA, and no audit trail for CS-to-Engineering communication. Requests get lost, timelines are unclear, and there is no accountability on either side for what was asked or what was answered.

NEW SYSTEM What Engineering Receives

No commitments without Engineering's written verdict

The CS Manager and AMs are prohibited from communicating any timeline or feasibility to clients until Engineering has issued a formal written verdict through the Notion queue. No exceptions. The AM communication template is gated on the verdict.

Every request arrives with a full Intelligence Packet

NotebookLM synthesizes the client's request, account history, and health context into a structured brief before it reaches Engineering. Engineering reads intelligence, not noise, and can issue a verdict without any additional context-gathering.

Structured SLA-governed queue in Notion

Every CS-to-Engineering request enters the Notion async queue with a priority level, an SLA timer, a full Intelligence Packet, and a verdict field. The entire lifecycle is tracked, visible, and accountable from submission through to client response.

Engineering Owns vs. Does Not Own

Clear ownership boundaries that eliminate the grey zones where confusion and overreach originate

One of the structural causes of friction between Client Success and Engineering is undefined ownership. When it is not clear who is responsible for a decision, the work defaults to whoever is most available, which is often Engineering. The table below draws a hard line between what Engineering is responsible for and what belongs entirely to the CS team and Product Manager. These boundaries are non-negotiable and are enforced by the CS Manager.

ENGINEERING OWNS

Feasibility classification of all technical requests

Engineering owns the verdict: Buildable Now, Roadmap Candidate, Out of Scope, or Needs PM Review. This classification is final. The CS team does not modify, interpret, or negotiate the verdict before relaying it to the client.

Async queue response within SLA

Engineering responds to every ticket in the Notion queue with a written verdict within the agreed SLA: 24 hours for standard requests, 4 hours for P1 critical escalations. No meeting is required for standard requests. A written verdict is the only deliverable.

Technical accuracy of all Engineering responses

Responses are relayed to clients exactly as classified by Engineering. The CS team uses approved language templates to communicate the verdict but does not alter the technical substance of what Engineering has determined.

Flagging when a live discussion is needed

If a request is too complex to classify asynchronously, Engineering flags it in Notion to request a focused 30-minute sync. The CS Manager schedules this. Engineering does not join open-ended client calls without an advance brief and a defined agenda.

ENGINEERING DOES NOT OWN

Client relationship management

CS and Account Managers own the client relationship entirely. Engineering does not manage client expectations, communicate timelines, or participate in client conversations unless specifically requested and properly briefed through the Notion system.

Timeline communication to clients

The CS Manager owns all timeline communication after Engineering issues a verdict. Engineering provides the classification. CS translates that into a client-appropriate message using the approved communication templates.

Feature request prioritization

The Product Manager owns all feature and roadmap prioritization entirely. Engineering classifies technical feasibility. PM decides what gets built and when based on the revenue-weighted intelligence CS delivers through the weekly Feature Request Log.

Responding to direct AM contact outside Notion

Under no circumstances does Engineering respond to direct Slack messages, emails, or verbal requests from Account Managers about client technical questions. All requests enter the Notion queue or they do not get addressed.

RACI Matrix Engineering Decision Authority

Every decision defined. Every owner named. No grey zones.

The RACI Matrix directly addresses one of the structural causes of friction: undefined ownership. When nobody is clearly accountable, the work gets handled by whoever is most available, which is often Engineering by default. This matrix names the owner of every key decision in the CS-to-Engineering pipeline and makes it non-negotiable.

R Responsible

A Accountable

C Consulted

I Informed

Action or Decision	CS Manager	Engineering	Product	Account Manager
Client technical request intake	R/A	I	I	R
Intelligence Packet creation via NotebookLM	R/A	I	I	C
NotebookLM synthesis and quality review	R/A	I	I	I
Engineering async queue routing	R/A	I	I	I
Feasibility classification verdict	I	R/A	C	I
P1 escalation confirmation	R/A	C	I	I
Client communication on technical items	A	I	I	R
SLA compliance monitoring	R/A	I	I	I
Feature request escalation to Product	R/A	I	C	C
Notion queue maintenance	R/A	I	I	C
Verbal commitment to client on tech items	NEVER	NEVER	NEVER	NEVER

90-Day Transformation Roadmap

The phased plan that transitions Engineering to structured, async-first collaboration

The 90-day roadmap is structured so that Engineering experiences measurable improvement as early as Week 3. Informal requests reduce in Month 1. The queue and SLA system is live by Week 2. Each phase has a specific output for Engineering and a clear measurement that confirms the system is working.

PHASE 1 Days 1 to 30	PHASE 2 Days 31 to 60	PHASE 3 Days 61 to 90
Foundation Install the infrastructure and establish the norms. Engineering's role in this phase is minimal: a 30-minute briefing in Week 1 to review the system, confirm the SLA structure, and agree on the Notion queue setup. From Week 2 onward the queue is live. MILESTONES WEEK 1 Engineering Briefing CS Manager presents the NotebookLM workflow, Notion queue structure, SLA commitments, and the zero-backdoor policy to Engineering leadership. Verdict classification confirmed. WEEK 2 Notion Queue Live Engineering async queue is deployed. CS Manager begins routing all requests through NotebookLM synthesis. No direct AM-to-Engineering requests permitted from this point. WEEKS 3 TO 4 Pilot and Calibration First Intelligence Packets enter the queue. Engineering issues verdicts. CS Manager tracks SLA compliance and collects feedback on packet quality.	Governance Activate the measurement layer and confirm the system is producing the outcomes it was designed for. Engineering leadership gains dashboard visibility into queue throughput, SLA compliance, and the reduction in informal request volume. MILESTONES WEEKS 5 TO 6 Dashboard Active Notion Query Health Dashboard goes live with Engineering leadership read access. CS Manager delivers bi-weekly pipeline report showing throughput and compliance. WEEK 7 Zero-Backdoor Enforced Any remaining informal AM-to-Engineering contact outside Notion is flagged and addressed by the CS Manager. AM team is fully trained on the capture form process. WEEK 8 Mid-Point Review Engineering Manager and CS Manager hold a 30-minute review: informal request volume versus Week 1 baseline, SLA compliance rate, packet quality feedback.	Scale and Optimize Prove the ROI of the system. By Day 90 Engineering can point to a measurable reduction in reactive interruptions, a documented SLA compliance rate, and a Notion queue functioning as the sole channel for all CS-to-Engineering communication. MILESTONES WEEKS 9 TO 10 Retrospective and Refinement Engineering provides formal feedback on Intelligence Packet quality and the Notion queue experience. CS Manager refines the NotebookLM synthesis prompts accordingly. WEEK 11 90-Day Metrics Report CS Manager prepares the Engineering Impact Report: informal request reduction, SLA compliance rate, and average response time. WEEK 12 Executive Presentation CS Manager presents 90-Day outcomes to leadership including Engineering's measured productivity improvement and proposal for the next phase of expansion.

Engineering should never have to manage client relationships, chase context, or absorb promises they never made.

This system gives Engineering back their most valuable resource:
focused, uninterrupted time to build.

Zero Unvalidated Requests

Every request arrives synthesized,
classified, and ready to evaluate.

Async on Engineering's Schedule

No Slack pressure. No unplanned
meetings. Your queue, your pace.

No More Blind Commitments

CS never promises what Engineering
has not formally approved.

Jayson Forde

Client Success Operations Strategy and Infrastructure
CoreForce Manager of Customer Success Candidate