



FDE INTERN ONBOARDING · SESSION 2 OF 2

The FDE Role

Where Finance context turns into your job, and why it exists



Agenda

00

Quick recap and reframe

01

Enter the FDE

02

Walkthrough: raw checklist to Discovery Report

03

Why you're building FDE tools

04

Wrap-up and discussion

Where we left off

- The Finance department: who owns the close, who consumes it
- Just enough vocabulary: GL, JE, chart of accounts, subledger, reconciliation
- What the close is: a recurring, time-boxed process with real structure
- Why it's hard: manual, spreadsheet-native, full of hidden dependencies

Every client we bring onto ClearSky has their own version of everything you just learned, written down in their own way, or not written down at all. Someone has to turn that into what FlightPlan needs. That's the FDE.

Enter the FDE

The industry term, and what it means specifically at ClearSky.



What “Forward Deployed Engineer” means

The archetype

An engineer embedded directly with a customer, working their real-world problem rather than building in the abstract, back at headquarters

At ClearSky specifically

A ClearSky team (generally 1 business + 1 tech) embedded with the client, virtually or on-site, translating their real-world process into what the platform needs, fast

The mandate, stated plainly



So the client can go live on ClearSky.

What the FDE is listening and looking for

1

Sequence & dependencies

What has to happen
before what

2

Maker-checker

Who does the work,
who checks it

3

Completion signal

What “done” looks like
for each task

4

Timing anchor

Business day, not
calendar date

5

Risk & controls

Where things can go
wrong, and what
catches it

Why this is slow today, and what it produces

Why it's slow and expert-dependent: it relies on one person's judgment, notes, and manual spreadsheet-wrangling across weeks of interviews and client review cycles.

The output: a structured task list and customer-facing Discovery Report that drive the FlightPlan schema, ready for platform configuration.

From Checklist to Discovery Report

Four real artifacts, in the order they actually got used.



The final deliverable was never the starting point



The next slides walk through steps 1 to 4 using the actual artifacts from this engagement.

The raw accounting ops checklist

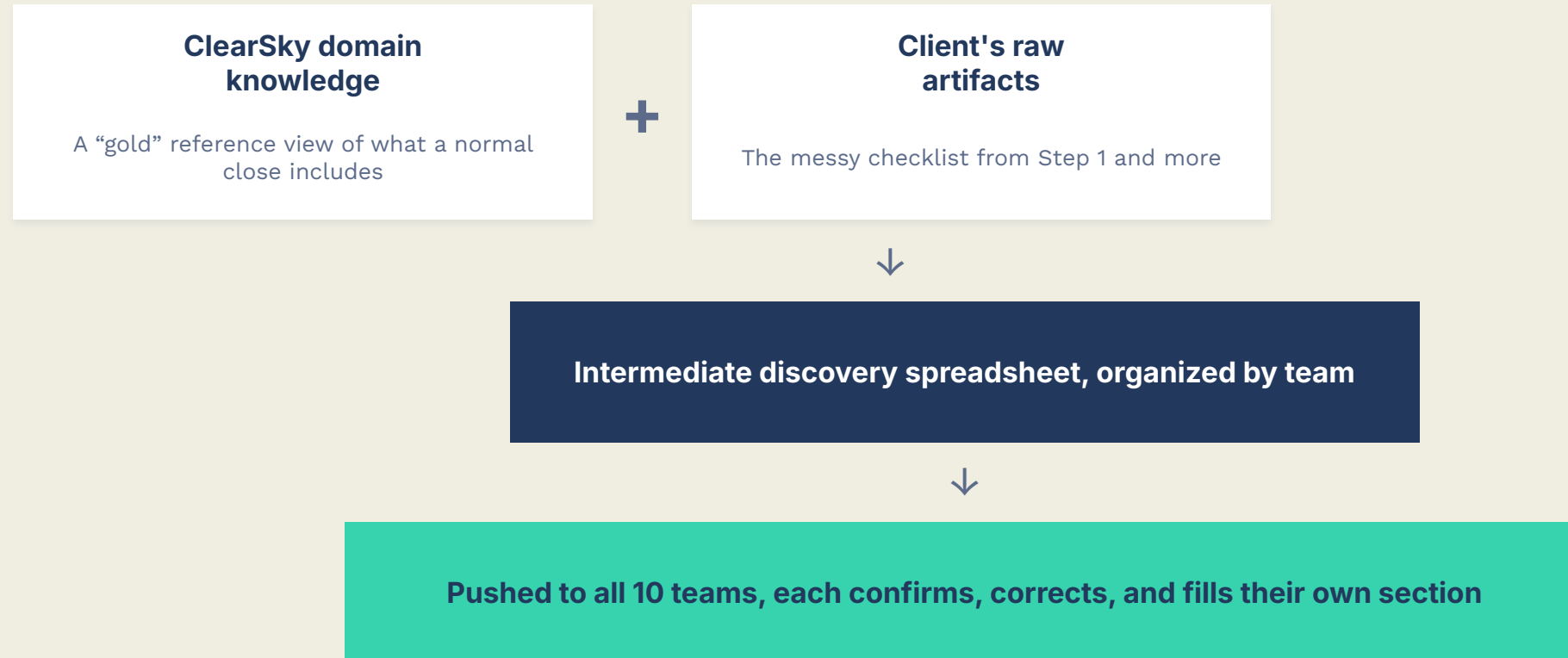
Bring up the actual client checklist. Point out what makes it raw rather than usable:

- Tasks are grouped loosely by category (Bank/Cash, Prepaid Expenses, Capital Assets, Leases), not by owning team
- “Responsible” is a staff-initials column, not a role or team, so ownership isn't legible to a newcomer
- Timing is a day-of-month offset relative to period end, not a date, since the sheet regenerates every period
- Dozens of near-identical rows repeat the same task once per legal entity or bank account
- Dependencies and controls are implicit: a “review, sign and date” row is a maker-checker step, but nothing marks it structurally

This is exactly what your extraction tooling has to make sense of. Real, messy, and where most engagements start.

The discovery spreadsheet, built and pushed to teams

ClearSky didn't take the raw checklist straight to interviews.



Discovery interviews

- Interviews with each team followed the spreadsheet update pass, focused on handoffs, dependencies, and risks
- These are the things people say out loud in a conversation, but rarely write down in a shared spreadsheet
- This step resolves what's still ambiguous or missing after Steps 1 and 2, it does not start from scratch

The key relationship to land here: interviews are downstream of artifact analysis, not parallel to it. They exist to close gaps the first two steps surfaced.

The Discovery Report: the client-facing deliverable

Once the discovery spreadsheet is complete and interviews have closed the gaps, the FDE turns that raw material into a formal deliverable the client actually receives.

Executive Summary



The findings, in plain language, for a CFO audience

Pain Points & Risks



What's manual, what's fragile, what could break

Effort & Risk by Team



Every task, classified and rolled up

Per-Team Detail



Role, systems, learnings for each of the 10 teams

Executive Summary, in the report's own words

ClearSky conducted structured discovery interviews with ten BlackBerry finance and accounting teams responsible for the quarterly close, covering 776 discrete close tasks across Accounting Operations, FP&A, Tax, Revenue, Treasury, Payroll & Benefits, AP, AR, External Reporting, and the QNX Revenue sub-group.

The findings reveal a close process that is heavily manual, deeply dependent on Excel as a bridging tool between disconnected systems, and reliant on individual expertise rather than codified workflows.

776

close tasks mapped

10

finance teams interviewed

142

high-risk tasks (18%)

94%

fully manual, no automation

Effort & Risk Breakdown, by team

Team	Tasks	Eff. High	Eff. Med	Risk High	Risk Med
Accounting Operations	438	41	353	44	366
FP&A	113	13	84	10	89
Tax	103	39	41	38	36
Treasury	36	2	11	15	11
Revenue	27	13	12	19	7
Accounts Payable	16	-	-	1	14
External Reporting	15	1	8	3	7
Payroll & Benefits	12	9	3	12	-
Accounts Receivable	9	-	-	-	7
Revenue - QNX	7	-	4	-	4
TOTAL	776	118	516	142	541

Of the 776 total close tasks, 118 (15%) are high effort and 142 (18%) are high risk. Accounting Operations alone accounts for 56% of total task volume.

Pain Points & Risks the report surfaces

Manual consolidation

Pulling reports from disconnected systems (NetSuite, Adaptive, Workday, ADP...) into Excel before close work can even begin

Key-person concentration

One analyst in Accounting Operations owns 243 tasks. The former AR manager who left took much of that team's knowledge with her

Cascade risk

Currency revaluation can't run until bank clearing and AP close; the tax provision waits on accounting. One upstream delay propagates through the whole close

Highest concentration of high-risk tasks: Tax (38) and Revenue (19), both tied to accuracy of financial statements.

From Discovery Report to FlightPlan input

The report is the client-facing artifact. Underneath it, the same 776 tasks resolve into the structured fields FlightPlan needs.

Task	Owner	Checker	Completion signal	Dependency	Timing anchor
------	-------	---------	-------------------	------------	---------------

The thing to emphasize: every field traces back to an artifact, a team update, or an interview moment. That traceability is what your tools need to preserve, not just the final answer.

Note: the “Frequency” field (Monthly/Quarterly/Annual/Adhoc) is a scheduling signal your tooling needs to parse.

Why You're Building FDE Tools

The payoff: connecting all of this directly to your work.



The core idea, mapped to the walkthrough

Tooling should do what Steps 1 through 4 currently do by hand, faster and with less variance, while keeping the FDE in the loop for judgment calls.



Explicit non-goal for this session: the exact schema, field names, and platform mechanics. That's the next deck.

TIME-TO-LIVE IS THE METRIC

Every week shaved off discovery is a week closer to a client running on ClearSky.

Why speed matters commercially: discovery today runs across weeks of interviews and manual spreadsheet-wrangling. Faster, lower-variance tooling compresses that timeline directly.

What “good” tooling looks like

Preserve provenance

Which artifact, which team update, which interview moment a value came from, not just the final structured answer. Easy to lose if it isn't designed in from the start.

Explicit non-goal for this session: the exact schema, field names, and platform mechanics. That's the next deck.

WRAP-UP

The FDE's bridge role, and the lineage from raw checklist to Discovery Task List to FlightPlan input.

Discussion prompts

- *Looking at the raw checklist, what would you want a tool to flag automatically before a human ever reads it?*
- *What's the riskiest thing to get wrong when merging a team's spreadsheet update with what the interview transcript says?*

UP NEXT

ClearSky Platform: FlightPlan itself, and the exact shape of what we're building toward.