

S-06, Anonymized sample from a real engagement

Mid sprint progress note, sprint 14

Measured answers to two client questions, the status of all eight items and two decisions requested.

Anonymized sample from a real sprint for a US investment analytics platform, 2026

What this is. The progress note I sent after working through the client's written feedback document. It answers the two direct questions with measured facts first, then gives the status of every item, then the two decisions I need. Names are replaced. Nothing in this sprint was live yet, and the note says so. The paragraph on the infrastructure apply now describes the written, plan first step I use, where the original proposed doing it together.

[Client], sprint 14 progress

Your document is worked through end to end. Both of your questions have measured answers rather than estimated ones, and every item you raised is either done or waiting on a decision from you.

Your document	Shipped to the repository	Waiting on you	Not charged
8 of 8 items	6 commits, 31 files	2 decisions	4.75 hours

Short version. You can now file scripts into categories by hand and it does reach the tree, and I proved that against a database rather than telling you it should work. You also get a bulk tool, because filing 342 scripts one at a time isn't a thing I would ask you to do.

Your two questions

Can you file individual scripts by hand

Yes, now. Before this sprint, no, and not for the reason you or I assumed. The edit dialog on the script list had the same fault as the upload page. It offered only categories three levels down a tree you don't have. Worse, its Update button was switched off unless a category

was chosen, so while the tree is empty you couldn't file a script and couldn't rename one either.

Does it reach the tree

Yes. I built a small tree on a real database, filed scripts into it and read the tree back through its own endpoint. A script filed at the top level showed one on that category and two counting everything beneath it. The category three levels down showed one. The unfiled count dropped. So the counts move the moment you file something.

Why I went to that trouble. The honest answer to "will this flow through" was available two ways. I could read the code and tell you it should, or I could file a script and look. The first way is how the level filter fault survived three sprints without anyone noticing it, so I did the second.

The bulk tool

Tick as many scripts as you like on the list, press File, choose one category, and they all move at once. A clear option also unfiles everything selected, and it's there on purpose. A bulk assign with no bulk undo is a trap, because the fastest way to make a mistake across three hundred rows is a tool that only runs one way.

When you come to do this properly, the quickest route is to search or sort the list, select a run of related scripts and file them together. Your four sector returns tables are one decision rather than four.

Everything else from your document

What you asked	Where it stands
Back button should return to the page you were on	Done. The page number lives in the address bar now, so page 3 stays page 3.
The dashboard should not sign you out	Done, and it never was signing you out. Your session was intact. The public version of the dashboard has a different header, with a sign in button and no route back, so it only looked that way.
One daily run at a fixed morning time	Done, with two exceptions I would not overrule without telling you. See below.
Remove the duplicate run	Removed outright, not just disabled. Rule, target, task definition and template.
Daily PDF to the support mailbox	Done, now that you confirmed the mailbox.
List of accounts inactive for over a year	Built and working. It counts what each account owns so you are not deleting blind.

What you asked	Where it stands
Delete the elevated account	Safe to do. It needs one decision from you first, below.
A copy of the token	Coming separately, not in a file that gets forwarded.
Fund drill down	Next. It leads the next sprint, since you asked to finish the document first.

The two exceptions on the single daily run

Seven rules are switched off. Six of their commands are now ordered steps of the new eleven step run. The seventh was a partial script rerun every four hours, which the daily full run makes redundant. Two jobs stay outside it, both for the same reason, so I want to be explicit rather than have you find it later.

The fund returns pull has to follow the US market close. The morning run starts before the market opens, so folding the pull into it would put a pre-open number under a closing price heading, and your cards would read a day stale. It stays in the evening and the card email follows it an hour later.

The job that clears stuck scripts runs every two hours. Making it daily would leave a stuck script stuck for a day, which is exactly the fault it exists to prevent.

The two decisions I need

The accounts

Your two requests point in opposite directions, and I would rather you choose than guess. The elevated account under your analyst's name can be deleted safely, and nothing he made would disappear. Scripts, reports and summaries all keep their rows and simply stop naming an author. I checked that on all three rather than assuming it.

But your analyst has asked for edit access so he can move scripts onto the internal API, and the platform grants that through the very same administrator flag. So deleting the elevated account and giving him edit access are the same switch pointing two ways.

What I would do is delete the old elevated account and put the flag on his working account instead. One account, exactly the rights it needs, no forgotten administrator. Say the word and it's a two minute job. I haven't touched either account.

Worth knowing before you decide. That flag also opens the Django admin site, because the platform uses one flag for both things.

The infrastructure apply

This change has grown large. It covers the two dormant jobs you approved, the duplicate scrape rule switched off, seven rules folded into the single daily run, one old collection task removed and the card schedule added.

An apply reconciles the whole environment rather than only the lines that changed. This platform has already had a schedule quietly reverted that way once, when a job was moved by hand and the definition still said the old time. So I'll send you the plan first, resource by resource, with what each line does. I apply it only after your written go ahead, and I check each changed resource against the live account after the apply.

On the token you flagged

Thank you for taking it seriously rather than treating it as housekeeping. It was worse than a key in a document. Two management commands carried a live account's password as plain text in the source. The worse of the two called itself a diagnostic, then reset the account to that password whenever its check failed. So running a command whose only stated job was to look at something could rewrite a live credential, to a value anyone holding the repository already knew.

Both are rewritten. A sweep at the end of the sprint then found the same password in two more places I had missed, a report at the root of the repository and the migration that creates the admin account. Both are closed too, so no account password is written down anywhere in the tree now.

All of it stays in the git history, so both affected accounts want fresh passwords regardless of these changes. I can do that whenever you want, and it takes a minute.

Where the work sits

Six commits are pushed to a branch on your repository rather than to main, so nothing is live until you give the go ahead for the deploy. The engineering report, the compliance sheet and the time statement are in the sprint folder alongside this note.

Nothing in this sprint is deployed. That's deliberate, and it's the state I would like your decision on.

Michael

Axdale, Michael Lip. Warsaw, Poland (CET). michael@axdale.com. www.axdale.com. Code record github.com/theluckystrike. Names, tickers and amounts are replaced.