

S-01, Anonymized sample from a real engagement

Sprint record, March to October 2026

Twenty sprints from March to October 2026, with dates, themes, counts and hours, and no money. The engagement began in August 2025.

Anonymized record of a real engagement for a US investment analytics platform, 2026.

The engagement began in August 2025. This record covers the written sprints from March 2026.

The client runs a platform where analysts upload Python scripts. The platform runs the scripts every night against market data. Members read return cards, fund drill downs, PDF reports and a daily email. The stack is Python and Django, React and TypeScript, PostgreSQL, AWS containers and scheduled tasks, Terraform and GitHub Actions.

All work was async. The client wrote a request, a review or a list of bugs. I wrote back a scope, then updates, then pull requests, then a report with evidence.

Every count below comes from the delivered reports, time statements, commit lists and pull request records. A dash means the source folder doesn't hold that count. Names, domains, repository names, billing numbers, rates and amounts are removed.

The sprints at a glance

Sprint	Dates (2026)	Days	Theme	Counts from the record	Hours billed
1	Mar 17 to May 8	about 7 weeks, part time	Platform recovery after a takeover	5 linked faults resolved, 11 failure points fixed, 23 files changed, 14 resolution items (12 deployed, 1 in progress, 1 client step), coding rules audit of 11 Python files	25 in the last week. Earlier weeks are not in the folder

Sprint	Dates (2026)	Days	Theme	Counts from the record	Hours billed
2	May 11 to 15	5	Infrastructure audit, dead task broker removed	147 managed cloud resources audited, 11 task definitions corrected, 4 client bugs root caused, 17 scripts confirmed running again on day 2	14
3	May 14 to 15	2	Script runner fix	301 of 331 scripts failing, traced to shared output buffers between threads and to containers at 512 MB. Success rate from 0.6 percent to 82 percent. 3 commits deployed	6.5
4	May 18	1	Three remaining bugs and a script guide	3 bugs investigated, 2 ready to deploy the same day, deliverables checked in 5 audit passes	4
5	May 18 to 22	5	Sign in, registration, password reset, report pipeline	Reports producing PDFs from 0 of 9 to 8 of 9 after a 72 day outage. Script success from about 10 to about 95 percent. Stuck runs now recover after 1 hour	32
6	May 25 to 29	5	Security hardening and report pipeline restoration	18 security and infrastructure changes, 15 files changed, 3 files added, 8 critical findings closed in code	25
7	Jun 1 to 4	4	Legacy external API retired, full script QA	63 plus 27 plus 28 scripts moved to the platform database, 116 dead settings removed, 342 scripts validated with a screenshot each, 334 succeeding and 8 failing at the end, 15 commits	40
8	Jun 14 to 17	4	Blank charts, flagged scripts, mail cutover	5 flagged scripts fixed and checked live, a migration boundary fault fixed, backend test gate at 36 tests, 247 scripts scanned	15
9	Jun 23 to 26	4	Static chart default, output fix, API access	Wrong chart output fixed at the source, admin sign in restored, working API access from the client's own PC, 3 written answers	16.5
10	Jul 1 to 5	5	Output guard, price ingestion, report email	5 of 5 items live. A blank render can no longer pass as a success. 5,773 price rows loaded. Ticker field widened from 5 to 12 characters by migration	21

Sprint	Dates (2026)	Days	Theme	Counts from the record	Hours billed
11	Jul 9 to 16	6	Script source access, report sender, naming rule	6 workstreams deployed. A claims check of 12 statements found 11 verified and 1 overstatement, which I corrected	25
12	Jul 21 to Aug 3	14, two parts	Report email delivery, analysis sandbox, annualized returns	15 validation runs in a row on the sandbox. Card returns for 1, 2, 5 and 10 years. Data audit of 24 funds, 0 fabricated values, 10 year mean difference 0.035 points	57.25
13	Aug 4 to 16	13	Cards from 4 to 16 funds, server side search and sort	12 of 12 behaviour checks, 0 TypeScript errors, production release, developer access	73, plus 10.25 not charged
14	Aug 19 to 24	6	Script upload and filing blockers, one daily run	1 pull request, 9 commits, 41 files, 8 blocking faults closed, bulk filing of up to 500 scripts, 11 step daily run replaced 6 schedules	36.5, plus 4.75 not charged
15A	Sep 2	1	Bounded, restartable price history preload	2 pull requests, 8 files, 1 migration, 52 backend tests passing	5
15	Sep 4 to 7	4	Six points from a written review	4 pull requests, 30 commits, 73 files, 1 migration, tests from 52 to 166, a 3 month infrastructure lock cleared	36, plus 2 not charged
16	Sep 17 to 23	7	Fund drill downs, one list process	3 pull requests, 60 commits, 72 files, 3 migrations, tests from 169 to 341	42, plus 3.5 not charged
17	Sep 30 to Oct 1	2	Three urgent analyst requests	1 pull request, 60 commits, 59 files, 4 migrations, tests from 342 to 439, 11.8 million history rows loaded in 34 minutes	32.5
17B	Oct 2 to 7	6	Security, script sandbox, health email	1 pull request, 100 commits, 189 files, 3 migrations, tests from 439 to 683	37, plus 4 not charged
18	Oct 11 to 19	9	New return window, 1,800 row drill down, dated reruns	6 pull requests, 77 commits, 116 files, 3 migrations, tests from 683 to 922, 441 report statements checked against the live site	54

Totals from the record

– 20 sprints in 31 weeks, from March 17 to October 19, 2026.

- 597.25 hours billed where the time statement is in the folder. Another 24.5 hours weren't charged, because they went on merges, reviews and fixes to my own work.
- Sprints 13 to 18 came to 316 billed hours, with 5 of them in sprint 15A. The seven sprints on the sprints page (13 to 18 without 15A) merged 16 pull requests with a record, and their commit lists hold 336 commits. Sprint 15A added 2 more pull requests.
- The backend test suite grew from 52 tests at the start of sprint 15 to 922 at the end of sprint 18.
- After sprint 1, sprint length ran from 1 day to 14 days. The median sprint was 5 days.

How the engagement changed over time

March to May, recovery

The platform had stopped refreshing data on March 9. One reported fault turned out to be five linked faults across scheduling, email records, redirects, monitoring and a data scraper. The first sprints were short and diagnostic. Each one closed a root cause and gave the client a plain written account of it.

June to July, reliability

Work moved to the script fleet. I retired a dead external API that 118 scripts still used, built a QA pass that opened every script page and saved a screenshot, and added guards so a blank chart can never count as a success.

August to October, product work

With the base stable, sprints became feature sprints with a written scope and hours per item. Return cards, fund drill downs, a script sandbox, a health email and dated reruns. Tests, migrations and review passes grew with them.

Sources

Sprint reports, time statements, delivery messages, commit lists and pull request records in the client folder, read only. Hours come from the time statements.

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