Technical Due Diligence: Company X

Prepared for: Deal team (sample) · Cristian Calabro · Engineering & AI Advisor · ccalabro.dev · Oct 26, 2026

Sample report. Company X, its people, numbers and findings are fictional, included only to show the format of a technical due diligence review.

1. Executive summary

Target: Company X, scheduling, dispatch and invoicing software for small field-service businesses (HVAC, plumbing, electrical). Seed stage, 12 people, about 420 paying customers and USD 1.4M in annual recurring revenue. Engineering is the co-founder CTO, two in-house engineers and a three-person outside agency that builds the mobile app.

Question: Is the technology safe to buy and grow on, and what has to be fixed, confirmed or priced in?

Verdict: Proceed, with conditions (Yellow). The product is sound, the revenue flows work and the team ships weekly. The risks sit around the code: access control, recoverability, IP ownership of the mobile app and dependence on the CTO. Three items should be resolved before close, four confirmed before close or priced in, and the rest belong in the 100-day plan.

Top 3 issues (no deal-stopping red flags if the conditions are met):

1. **The agency holds the mobile app's code and store accounts, with no IP assignment (H1).** Make signed assignments and the transfer a closing condition.
2. **Former contractors still have admin access, and MFA isn't enforced (C1).** 1–2 days. Fix before close and confirm with a fresh access export.
3. **Backups have never been restored (C2).** Ask for a timed restore before close; it takes 2–3 days.

Total cost to fix (estimate): 40–66 engineer-days, about **$32,000–$52,800** at an assumed USD 800/day blended rate. **Before close:** C1, C2 and the H1 assignments; confirm H2, H3, H4 (CTO retention) and M5. **Price in:** H5, M1 and M2. Cloud clean-up (M3) adds back about $2,300 a month.

What's in good shape: weekly releases; TypeScript across the API and web apps, strict mode in the API; managed Postgres with daily snapshots and point-in-time recovery on; card data handled entirely by Stripe; most infrastructure in Terraform; a strong API test suite around billing; a rules-based scheduling engine that is well tested and genuinely differentiating; design notes written before big changes.

2. Scope and method

Two-week review, Oct 12–23, 2026. Management interviews (CEO and CTO), plus the two in-house engineers, the agency lead and the head of customer success. Data room: agency and contractor agreements, key vendor contracts, org chart, 12 months of cloud and AI invoices, the incident log and the latest customer security questionnaire. Read access to all four repositories, the AWS console, CI, the Stripe dashboard and both issue trackers. Review of 90 days of incidents and pull requests. Read-only: no changes were made to any system.

System or area	What was reviewed	Rating
Customer booking portal	Homeowner booking and card payment flow (React), payment handling, uptime	Yellow
Business web app	Scheduling, dispatch and invoicing for customers' office staff; tests, releases	Yellow
Internal admin dashboard	Support tooling, impersonation, refunds, exports, access	Red
Crew mobile app	iOS and Android app (React Native), ownership, release process, tests	Red
Backend API & workers	Node/TypeScript API, scheduling engine, SMS and billing jobs, runtime versions	Yellow
Data & payments	Postgres, backups and restore, Stripe subscriptions and card payments, webhooks	Red
Cloud & costs	AWS setup (ECS Fargate, RDS, S3), environments, 12 months of spend	Yellow
Security & access	AWS, GitHub and Google Workspace access, MFA, secrets, offboarding, vendor access	Red

Reliability & incidents	Monitoring, alerting, on-call, incident history	Yellow
Team, process & IP	Key-person risk, agency dependence, code review, documentation, IP assignment, licenses	Yellow
AI use	AI claims vs the product, the AI feature, data and vendor terms, AI costs, coding tools	Yellow

3. Findings, ranked

Severity: Critical fix before close · High confirm before close or price in · Medium fix in 30–90 days · Low housekeeping. Costs are estimates at USD 800/engineer-day.

Critical

Critical C1. Former contractors still have admin access, and MFA isn't enforced · Security & access

Evidence: Two of the five people who left in the last 12 months (both agency staff) are still in the GitHub organization. One still has an AWS user with full admin rights and an active access key, last used in August, the week they left. MFA is off for 4 of 9 AWS users and isn't required in GitHub or Google Workspace. The AWS root login is shared through a password manager. No offboarding checklist.

Business impact: Anyone holding those credentials can reach production and customer data across every system. A breach between signing and close would land on the buyer.

Fix: Remove the accounts and key, require MFA on AWS, GitHub and Google Workspace, put root behind hardware keys, and move AWS access to single sign-on. Confirm with a fresh access export.

Cost to fix: 1–2 days · $800–$1,600 | **When:** Pre-close

Critical C2. Production backups have never been restored, and everything lives in one AWS account · Data & reliability

Evidence: RDS automated backups (7-day retention) and point-in-time recovery are on, but no restore has ever been run and there is no runbook. Backups, production, staging and the CI deploy user share one AWS account. The S3 bucket with job photos and invoices (about 380 GB) has no versioning.

Business impact: Recovery time after data loss is unknown, and one compromised admin key could delete production and its backups together. Schedules, invoices and payment records are the product.

Fix: Run a timed restore and document it, copy snapshots to a separate backup account, turn on S3 versioning, and repeat quarterly.

Cost to fix: 2–3 days · $1,600–$2,400 | **When:** Pre-close (verify)

High

High H1. The agency holds the mobile app's code, store accounts and signing keys, with no IP assignment · IP & licensing

Evidence: The crew app's React Native repository is in the agency's GitHub organization (Company X has read access). The Apple and Google developer accounts are registered to the agency; the Android signing key is on an agency laptop. The agency contract (month-to-month) has no IP assignment clause, and neither do two of the three individual contractor agreements in the data room.

Business impact: About 60% of daily active users are field crews on mobile. Without assignment, ownership of that code is open to dispute, and the company can't ship an app update without the agency.

Fix: Signed IP assignments from the agency and the two contractors; transfer the repository and both store accounts; move signing keys into Company X's secret store.

Cost to fix: 2–3 days · $1,600–$2,400 | **When:** Pre-close (closing condition)

High **H2. Support staff can sign in as any customer, with no audit trail** · Admin dashboard

Evidence: The internal admin dashboard (part of the API repository) has a “log in as customer” feature. Nine staff accounts can use it, two of them shared logins. Impersonation, refunds and data exports are not logged. The dashboard is reachable from the public internet behind a password only.

Business impact: There's no way to show who viewed or changed a customer's data, which customers' security reviews and privacy law increasingly expect, and a leaked support password exposes every account.

Fix: Remove shared logins, put the dashboard behind single sign-on with MFA, log every impersonation, refund and export, and limit impersonation to named roles.

Cost to fix: 3–5 days · $2,400–$4,000 | **When:** Pre-close (confirm), 30 days

High **H3. Customer job notes go to an AI provider with no data agreement, spend limit or separate keys** · AI use

Evidence: The “AI job summary” feature (launched in July) sends technicians' notes, which often include names, addresses and phone numbers, to a third-party LLM API. One key is shared by all environments and is in the API repository's history. AI spend rose from about $300 in July to $1,900 in September with no limit. No data processing agreement in the data room; the privacy policy doesn't mention the feature.

Business impact: A privacy and contractual exposure on every customer using the feature, and an uncapped, growing cost line.

Fix: Confirm the provider's data terms and sign a DPA, rotate and split keys with spend limits, mask personal data before sending, update the privacy policy, and track AI cost per customer.

Cost to fix: 3–5 days · $2,400–$4,000 | **When:** Pre-close (confirm)

High **H4. Key-person risk: the CTO holds deploys, billing logic and infrastructure** · Team & process

Evidence: The co-founder CTO wrote about 70% of the billing code, is the only person with production deploy access, and is the only one who knows the Terraform setup. No architecture overview. Both in-house engineers have under two years' tenure.

Business impact: Losing the CTO after close would stall releases and billing changes for weeks. Retention and knowledge transfer are part of the deal value.

Fix: Retention terms for the CTO; pairing on deploys and billing, a written architecture overview and runbooks, and production access for a second engineer within 60 days.

Cost to fix: 4–6 days · $3,200–$4,800 | **When:** Pre-close (confirm retention), 100-day plan

High **H5. API releases run by hand from the CTO's laptop, with no tested rollback** · Delivery

Evidence: Deploys (about two a week) run from a script on the CTO's machine. Database migrations run in the same step and can't be reversed. Staging is updated by hand and was three weeks behind production during the review. The last release that had to be undone (Sep 18) took 2 h 40 min to recover.

Business impact: Releases depend on one person and one laptop; a bad release means hours of disruption in customers' working day.

Fix: Deploy from CI with approval for production, deploy to staging automatically first, keep one-step rollback, and write backward-compatible migrations.

Cost to fix: 5–8 days · $4,000–$6,400 | **When:** 100-day plan; price into deal

High **H6. No alerting: customers find outages first** · Reliability & incidents

Evidence: Errors are collected, but no alert reaches a person. Both incidents in the last 90 days were reported by customers: SMS appointment reminders failed silently for 9 hours on Aug 27 after the SMS provider credentials expired, and the business web app was down for 47 minutes on Oct 2. No on-call rotation or incident notes; the incident log in the data room was written for this review.

Business impact: Customer-facing reliability is lower than the uptime figures in the data room suggest, and churn risk rises with every silent failure.

Fix: Uptime checks for the booking portal, web app and API; alerts on errors, queue age and SMS failures; an on-call rotation and a note after each incident.

Cost to fix: 3–5 days · $2,400–$4,000 | **When:** 30 days

Medium

Medium **M1. Payment webhooks aren't idempotent, and the payments SDK is two major versions behind** · Data & payments

Evidence: The Stripe webhook handler for booking-portal card payments doesn't check whether an event was already processed. Three duplicate charges in the last 90 days were refunded by hand. The payments SDK is two major versions behind current.

Business impact: Small today, but duplicate charges grow with volume and reach customers' customers directly.

Fix: Store processed event IDs and skip repeats, add tests for retries, upgrade the SDK in stages with checkout and refund tests.

Cost to fix: 3–5 days · $2,400–$4,000 | **When:** 100-day plan; price into deal

Medium **M2. Tests are strong in the API and thin everywhere else** · Code quality & testing

Evidence: API: about 1,100 tests, 71% line coverage, 84% in billing. Web (booking portal and business app): 46 tests, mostly snapshots, not required to pass. Mobile: no automated tests. The API runs on Node 18, out of support since April 2025; 47 open dependency alerts, 6 rated high.

Business impact: Regressions in the web and mobile apps reach customers, and the runtime upgrade is unbudgeted work.

Fix: Make web tests block merges, add end-to-end tests for book, dispatch, complete, invoice and pay, a mobile smoke test, and upgrade to Node 22 LTS.

Cost to fix: 6–10 days · $4,800–$8,000 | **When:** 100-day plan; price into deal

Medium **M3. Cloud spend is about a third higher than it needs to be** · Costs

Evidence: AWS costs about $6,800 a month. Two unused environments cost about $900, staging runs 24/7 at production size (about $1,100), and the production database averages 12% CPU on a large instance.

Business impact: About $2,300 a month (34%) can go without affecting customers: roughly $27,600 a year of margin.

Fix: Remove unused environments, shrink and schedule staging, right-size the database, tag resources and set budget alerts.

Cost to fix: 2–4 days · $1,600–$3,200 | **When:** 30 days

Medium **M4. Code review is optional, and the agency approves its own changes** · Team & process

Evidence: 38% of pull requests merged into the API and web repositories in the last 90 days had no review. Branch protection is off. Mobile changes are reviewed only inside the agency.

Business impact: Risky changes to billing or permissions can ship unseen, and knowledge stays with whoever wrote the code.

Fix: Branch protection with one approval and passing CI; in-house review for billing, sign-in, data deletion and all agency changes. Add automated reviews: an AI review bot on every pull request (such as CodeRabbit or GitHub Copilot code review) and a review agent developers run locally before opening one.

Cost to fix: 1–2 days · $800–$1,600 | **When:** 30 days

Medium **M5. An AGPL-licensed library is used in the invoice service, and no license scan runs** · IP & licensing

Evidence: Invoice PDFs are generated server-side with an AGPL-licensed library. CI runs no license check across the four repositories.

Business impact: Probably manageable for a hosted service, but it needs a legal view and is the kind of item that delays a later sale or audit.

Fix: Get counsel's view; replace the library if needed (invoice generation is small and well tested); add a license check to CI.

Cost to fix: 2–3 days · $1,600–$2,400 | **When:** Pre-close (confirm), 100-day plan

Low

Low **L1. Documentation is thin, and onboarding takes 9 days** · Team & process

Evidence: The last hire took nine working days to run the stack and ship a first change. READMEs describe 2024 setup steps. Planning is split between two trackers and a spreadsheet.

Business impact: Slows the two hires planned for next year and makes post-close integration harder.

Fix: One-command local setup, current READMEs, one tracker for in-house and agency work.

Cost to fix: 2–3 days · $1,600–$2,400 | **When:** 100-day plan

Low **L2. AI coding tools are used with no shared rules** · AI use

Evidence: Four different AI coding assistants are in use; two agency developers use personal accounts. No guidance on customer data, secrets or review of AI-written code.

Business impact: Customer data or code can end up in tools the company doesn't control.

Fix: Approve one or two tools on company accounts with training off; a one-page AI use policy; same reviews as other code (M4).

Cost to fix: 1–2 days · $800–$1,600 | **When:** 30 days

4. AI claims check

Claim: the pitch deck describes “AI-powered scheduling that fills your crews' calendars.”

What we found: scheduling is a rules-based engine (travel time, skills, time windows, crew capacity), well tested and a real asset, but not AI. The only AI in the product is the job summary feature (H3), a thin layer over a third-party LLM API. AI spend is about $1,900 a month and rising with usage; no cost is tracked per customer.

What it means: value the scheduling engine as strong conventional software. Treat the AI feature as easy to copy, with a variable cost and an open data-terms question that needs answering before close.

5. Team and dependencies

Person or vendor	Role and dependence	Risk
Co-founder CTO	Only production deployer; owns billing logic and infrastructure (H4)	High
Two in-house engineers	API, web apps and the scheduling engine; 8 and 20 months' tenure	Medium
Outside agency (3 people)	Builds and releases the crew app; month-to-month, no IP clause (H1)	High

Key vendors	Cloud, payments, SMS: standard terms. No DPA on file for the LLM provider (H3)	Medium

6. 30/60/90-day plan

When	Actions
Before close	Remove leavers' access, enforce MFA, lock down root (C1). Timed restore test (C2). Signed IP assignments from the agency and contractors (H1). Confirm admin access controls (H2), LLM data terms (H3), CTO retention (H4) and the license question (M5).
Days 1–30	Transfer the mobile repository and store accounts (H1). Admin audit log and single sign-on (H2). Split AI keys, masking, spend limits (H3). Alerts and on-call (H6). Cloud clean-up (M3). Review rule and automated reviews (M4). AI tool policy (L2).
Days 31–60	Deploys from CI with rollback (H5). Knowledge transfer from the CTO and a second deployer (H4). Idempotent payment webhooks (M1). Backup account and S3 versioning in place (C2).
Days 61–90	Web, end-to-end and mobile tests; Node 22 upgrade (M2). Payments SDK upgrade (M1). License replacement if needed (M5). Documentation and onboarding (L1).

7. Cost-to-fix summary

#	Severity	Finding	Eng-days	USD	When
C1	Critical	Former contractors still have admin access, and MFA isn't enforced	1–2	$800–$1,600	Pre-close
C2	Critical	Production backups have never been restored, and everything lives in one AWS account	2–3	$1,600–$2,400	Pre-close (verify)
H1	High	The agency holds the mobile app's code, store accounts and signing keys, with no IP assignment	2–3	$1,600–$2,400	Pre-close (closing condition)
H2	High	Support staff can sign in as any customer, with no audit trail	3–5	$2,400–$4,000	Pre-close (confirm), 30 days
H3	High	Customer job notes go to an AI provider with no data agreement, spend limit or separate keys	3–5	$2,400–$4,000	Pre-close (confirm)
H4	High	Key-person risk: the CTO holds deploys, billing logic and infrastructure	4–6	$3,200–$4,800	Pre-close (confirm retention), 100-day plan
H5	High	API releases run by hand from the CTO's laptop, with no tested rollback	5–8	$4,000–$6,400	100-day plan; price into deal
H6	High	No alerting: customers find outages first	3–5	$2,400–$4,000	30 days
M1	Medium	Payment webhooks aren't idempotent, and the payments SDK is two major versions behind	3–5	$2,400–$4,000	100-day plan; price into deal
M2	Medium	Tests are strong in the API and thin everywhere else	6–10	$4,800–$8,000	100-day plan; price into deal
M3	Medium	Cloud spend is about a third higher than it needs to be	2–4	$1,600–$3,200	30 days
M4	Medium	Code review is optional, and the agency approves its own changes	1–2	$800–$1,600	30 days
M5	Medium	An AGPL-licensed library is used in the invoice service, and no license scan runs	2–3	$1,600–$2,400	Pre-close (confirm), 100-day plan
L1	Low	Documentation is thin, and onboarding takes 9 days	2–3	$1,600–$2,400	100-day plan
L2	Low	AI coding tools are used with no shared rules	1–2	$800–$1,600	30 days
Total			**40–66**	**$32,000–$52,800**	

Assumes a USD 800/day blended rate and the current team. Excludes cloud savings (M3) and legal costs (H1, M5). Delivered with a walkthrough call.