

# Software Development Lifecycle and Quality Practices

**Status:** Published, [Secureframe Link](#)

**Published date:** July 29, 2026

**Last updated:** July 29, 2026

**Policy owner:** Anatoly Geyfman

How Deviceflow builds, tests, releases, and maintains its platform.

## Overview

---

Deviceflow follows an agile software development lifecycle with continuous integration, automated testing, and controlled release management. This document describes our current practices for customers who need to evaluate our development processes as part of their supplier qualification and computerized system validation.

## Technology stack

---

Deviceflow is a TypeScript monorepo built on SvelteKit (frontend and API layer), PostgreSQL 16 (database), and Drizzle ORM (type-safe database access). The platform runs as containerized services on AWS ECS Fargate. Infrastructure is managed as code using Terraform.

## Development methodology

---

Deviceflow uses an agile development process. Work is tracked in Linear, where requirements are captured as issues linked to user needs and organized into project milestones. Features move through a defined workflow: backlog, in progress, in review, and done.

All source code is managed in GitHub with branch protection rules. Changes require pull requests with at least one peer review before merge to the main branch. Commit messages reference the corresponding Linear issue for traceability.

## Requirements management

---

User needs are documented and maintained in our project management system. Each user need traces to one or more user stories (engineering work items) that describe the functional behavior built to satisfy it. This traceability is published in the Requirements Traceability Matrix (RTM) included in every release's validation pack.

Requirements originate from customer feedback, business analysis, and regulatory needs. They are reviewed and prioritized by the product team before entering development.

## Testing

---

Deviceflow employs multiple levels of testing:

- **Unit and integration tests** — Run on every push and pull request via GitHub Actions CI. These verify business logic, service functions, and component behavior.
- **Automated end-to-end tests (Playwright)** — Browser-based automated tests using Chromium that exercise the application as a real user would. These are the primary source of validation evidence. Each test maps to a User Requirement Specification (URS) and produces structured results including step-by-step screenshots, video recordings of test execution, database audit event verification, and performance timings. Results are published at [validation.deviceflow.com](https://validation.deviceflow.com).
- **Static analysis (Semgrep)** — Custom rules enforce security patterns, prevent SQL injection via raw queries, validate import conventions, and

check database migration safety. Semgrep runs in CI and blocks merges on ERROR-severity findings.

- **Manual testing** — Performed during development for usability, functionality and validation.

All automated tests run as part of our GitHub Actions CI pipeline. Tests must pass before changes can be merged and deployed. Test results are retained with timestamps and are traceable to the requirements they cover.

## Validation test evidence

Each URS test produces a structured evidence package containing: a result.json file with test metadata and pass/fail status per scenario, a narrative report, step-by-step screenshots, video recordings of each test step, database audit event logs generated during the test, and other related database evidence. This evidence is committed to version control and published on the validation site, where it is browsable, searchable, and versioned by release.

## Release and deployment

---

Deviceflow ships on a roughly weekly release cadence. Each release is identified by date and Git commit SHA (e.g., [2026-04-20-d326c2b](#)) In the case of multiple releases in a given day, subsequent releases include an additional number to understand ordering (e.g., [2026-04-20-1-5c893b2](#)).

## Our release process:

1. Changes are developed on feature branches and reviewed via pull request.
2. Approved changes are merged to the main branch, triggering CI (lint, unit tests, build verification).
3. On merge to main, changes are automatically deployed to the staging environment with a health check gate for manual testing and validation by the team.

4. Database migrations run as isolated ECS Fargate tasks in private subnets with direct RDS access.
5. After staging verification, changes are promoted to production ([app.deviceflow.com](https://app.deviceflow.com)) via merge to the production branch.
6. Production deployment includes health check verification and Slack notification on success or failure.
7. The validation site is updated with new test evidence and documentation. (TBD)

Releases can be rolled back if issues are discovered in production. The ECS deployment model supports rollback to the previous task definition.

## Change management

---

All changes to the Deviceflow platform are tracked in version control (Git) and our project management system (Linear). Each change record includes the author, timestamp, description, rationale, linked requirement, and reviewer.

Changes are classified by scope and risk:

- **Feature changes** — New functionality or modifications to existing behavior. Accompanied by updated requirements, tests, and validation evidence.
- **Bug fixes** — Corrections to existing behavior. Tested and documented before release.
- **Infrastructure changes** — Updates to hosting, dependencies, or configuration. Assessed for impact on validated functionality.

Customers are notified of changes that may impact validated functionality. Notifications include a description of the change, affected features, and guidance on whether re-validation may be warranted.

# Security and access controls

---

## Application security

- Unique user identification required for all users. Shared accounts are not permitted by system design.
- Role-based access control (RBAC) with configurable permission levels (administrator, principal, rep, and custom roles).
- Passwords hashed using industry-standard algorithms and transmitted only over TLS.
- Session timeout of 7 days after inactivity.

## Infrastructure security

- All data encrypted in transit (TLS 1.2+). HTTPS enforced for all connections.
- All data encrypted at rest. RDS storage encryption is mandatory across all database instances.
- Database credentials stored in AWS Secrets Manager with random 32-character passwords.
- Production databases are not publicly accessible. Access is restricted to the VPC (private subnets) via security groups.
- Database access for engineering requires VPN (Twingate) through a secure tunnel.
- All application data changes go through the application layer where audit trail controls are enforced.
- AWS CloudTrail enabled for infrastructure audit logging.
- AWS GuardDuty enabled for threat detection.
- VPC Flow Logs enabled across all regions.

## Backup and recovery

---

- Automated RDS backups with a 7-day retention period.
- Backup window: 03:00–04:00 UTC daily.

- Database runs in Multi-AZ configuration (us-west-2a, 2b, 2c) for automatic failover.
- A read replica is maintained for analytics workloads, isolating reporting queries from production traffic.
- RDS Performance Insights enabled for monitoring.
- Maintenance window: Sundays 04:00–04:30 UTC.
- Docker images retained in ECR for 30 days, enabling rollback to any recent release.

## Incident and defect management

---

Defects are tracked in our project management system with severity classification. Critical defects affecting validated functionality are prioritized for immediate resolution. Customers are notified of any incidents that may affect their use of the platform.

## Regulatory alignment

---

Our SDLC and quality practices are designed to support customers' compliance with:

- 21 CFR Part 11 (Electronic Records; Electronic Signatures)
- EudraLex Volume 4 Annex 11 (Computerised Systems)
- ISPE GAMP 5 Second Edition (July 2022)
- PIC/S PI 011-3
- FDA Computer Software Assurance draft guidance (2022)

## Current state and roadmap

---

Deviceflow is an early-stage company. We are transparent about where our practices are today and where we are headed. We update this document as our practices mature. Customers can request the current version at any time.

# Attestation

---

I attest that the information provided in this document is accurate and complete to the best of my knowledge.

| Field     | Value                                                                              |
|-----------|------------------------------------------------------------------------------------|
| Name      | Anatoly Geyfman                                                                    |
| Title     | CEO                                                                                |
| Signature |  |
| Date      | 07/29/2026                                                                         |

For more information, visit [deviceflow.com](https://deviceflow.com) or contact us directly.