# Secure Development and Change Management Policy

Dash Media Limited · Version 1.1 · Effective 28 October 2026 · Review by 28 October 2027 · Owner: Brad Hook, Director, Dash Media Limited

## 1. Purpose

To describe how software changes are made, reviewed, tested, and deployed so that participant data stays protected as the product evolves.

## 2. Development environment

- Source code lives in private GitHub repositories. Access requires a GitHub account with multi-factor authentication.
- Development is done locally against a throwaway local database (an in-process Postgres seeded with made-up people), never against production data. There is no staging copy of production.
- Secrets are loaded from environment files that are excluded from version control. The full history of each repository is scanned for committed secrets with gitleaks at every security review, and the result is recorded in the Trust Center verification notes.
- AI coding assistants (Claude Code) are used in development. They operate on the codebase and local environment; they are not given production credentials or production participant data.

## 3. Change process

Every change to production follows the same path:

1. Work happens on a branch, never directly on the main branch.
2. Pushing the branch creates a Vercel preview deployment with its own URL.
3. The preview is checked before merging: the feature works, existing flows still work, and any change to authentication, Row Level Security, or data handling is tested specifically.
4. Merging to main deploys to production automatically.
5. If a deployment causes a problem, Vercel's instant rollback restores the previous build.

As a one-person team, there is no separate reviewer. To compensate, changes touching authentication, access rules, data retention, or the AI prompt are deployed in their own small commits with a clear message, so they can be identified and reversed individually.

## 4. Security practices in code

- **Input validation** on every server-side route and database function.
- **Row Level Security** is the primary access control. Application code is not trusted to enforce who can read what; the database is.
- **Parameterized queries** via the Supabase client; no string-built SQL.

- **Dependencies** are monitored by GitHub Dependabot. Security updates are applied within 7 days of release for high or critical severity, and at the next routine update otherwise.
- **Framework updates** (Next.js, Supabase client) are applied at least quarterly.
- **No participant data in logs.** Logging statements are reviewed before merge to confirm they don't print responses, codes, or tokens.
- **AI prompts** that send data to the Claude API are reviewed to confirm they contain only the fields listed in the Data Protection policy.

## 5. Database changes

- Schema changes are made through migration files committed to the repository, not by editing production tables by hand. A script applies each file once and records it in the database.
- Every new table holding participant data has Row Level Security enabled and policies written before it receives any data.
- A backup is confirmed before any migration that alters or drops columns.

## 6. Testing

- Every push to the main branch is built by Vercel, and a type error or build error stops the deployment. Linting and the database permission test suite run locally before a push. There is no separate CI service.
- Authentication and access flows are tested manually on the preview deployment for every change that touches them.
- An annual review walks through the OWASP Top 10 against the app and records findings.

## 7. Emergency changes

If a security fix must ship immediately, it still goes through a branch and preview, but the preview check is limited to the fix itself. The change is reviewed in full within 48 hours and documented.

## 8. Review

Reviewed annually and when the development toolchain changes.

## 9. Version history

| VERSION | DATE | CHANGE |
|---|---|---|
| 1.0 | 2026-10-27 | Initial publication |