Capital Investment Club

Change Management Policy

Change request, approval, testing, deployment, rollback, emergency change, evidence, and post-implementation controls.

Last updated: 05/23/2026

Purpose

This policy defines how Capital Investment Club (CIC) requests, evaluates, approves, implements, validates, communicates, and monitors changes that may affect the CIC platform, customer data, financial workflows, security posture, production infrastructure, internal operations, or published governance commitments.

The objective is to ensure that changes are intentional, authorized, traceable, tested, recoverable, and aligned with CIC's obligations to users, organizations, grantors, pitchers, investors, assessors, support teams, administrators, financial partners, and regulators.

Scope

This policy applies to changes involving:

- CIC web applications, APIs, notification services, landing pages, and administrative portals.
- Production, staging, development, CI/CD, and cloud infrastructure.
- Authentication, authorization, organization context, KYC-adjacent workflows, support access, and administrative tooling.
- Wallets, grants, pitches, investments, equity calculations, ledger/PFS functionality, payments, withdrawals, and reconciliation workflows.
- Database schemas, indexes, migrations, one-time repair scripts, backfills, and data retention rules.
- Third-party integrations, financial API providers, identity providers, email/SMS providers, storage services, observability tooling, and vendor configuration.
- Production secrets, environment variables, deployment settings, access policies, and security controls.
- Public-facing policies, terms, privacy notices, support procedures, and governance documents.

This policy applies to employees, contractors, administrators, agents, service providers, and automated systems that propose, build, approve, deploy, or operate CIC changes.

Policy Statement

CIC must manage material changes through a documented change process. Changes must be reviewed and approved based on risk, tested before release where practical, deployed through controlled mechanisms, and monitored after implementation.

No material change may be introduced into production without:

- A defined business, security, compliance, operational, or customer need.
- An identified owner.
- A change record or issue/PR reference.
- Risk assessment proportionate to impact.
- Approval appropriate to the change type.
- Validation evidence.
- A rollback or recovery approach where practical.
- Post-change monitoring for high-risk or user-facing changes.

Change Categories

Standard Change

A standard change is a repeatable, low-risk change that follows an approved pattern and has a known validation path.

Examples:

- Routine dependency patch with no breaking changes.
- Content update to a non-regulated page.
- Low-risk UI copy correction.
- Approved configuration update with established runbook.

Standard changes may use pre-approved procedures but must still be traceable.

Normal Change

A normal change is any feature, fix, configuration update, workflow update, or operational change that requires review before release.

Examples:

- New product feature.
- Bug fix affecting user workflows.
- API behavior change.
- Database migration.
- CI/CD workflow change.
- Policy or terms update.

Normal changes require a change record, review, approval, testing, and deployment validation.

Emergency Change

An emergency change is required to address an urgent incident, production outage, active security risk, critical vulnerability, data integrity risk, financial exposure, or severe customer-impacting defect.

Emergency changes may use expedited approval, but must be documented after implementation with:

- Triggering event.
- Approver.

- Change made.
- Validation performed.
- Rollback or recovery outcome.
- Follow-up root cause analysis.
- Permanent remediation or control improvement.

Security-Sensitive Change

A security-sensitive change affects authentication, authorization, secrets, support access, file uploads, public profile visibility, user privacy, data exports, KYC-adjacent workflows, reporting, logging, incident response, vendor access, or production infrastructure.

Security-sensitive changes require heightened review and must consider confidentiality, integrity, availability, privacy, misuse, and abuse scenarios.

Financially Sensitive Change

A financially sensitive change affects wallets, grants, pitches, investments, contributions, withdrawals, equity calculations, fees, currency conversion, ledgers, PFS posting, reconciliation, payment provider configuration, or financial reporting.

Financially sensitive changes require data integrity validation, regression testing, and explicit review of affected calculations or transaction states.

Change Request Requirements

Each material change should capture:

- Request title and description.
- Business rationale.
- Requester and owner.
- Affected systems, repositories, services, data, and roles.
- Change category and risk level.
- Security, privacy, financial, legal, and operational impact.
- Testing plan.
- Deployment plan.
- Rollback or recovery plan.
- Required approvals.
- Communication plan where users, support, or operations are affected.
- Evidence requirements.
- Implementation date and deployment reference.

For CIC product work, the issue or pull request may serve as the change record if it contains the required information.

Risk Assessment

Changes must be risk assessed before implementation based on:

- User impact and workflow criticality.
- Data sensitivity.
- Authorization impact.
- Financial impact.
- Transaction integrity.
- Reversibility.
- Operational complexity.
- Dependency and vendor impact.
- Deployment blast radius.
- Compliance or contractual impact.
- Prior incident history.

High-risk changes include:

- Wallet, grant, pitch, investment, equity, withdrawal, ledger, or reconciliation changes.
- Changes to authentication, authorization, organization context, public profile visibility, support access, or admin privileges.
- Database migrations, backfills, repair scripts, and destructive data operations.
- Production secret, environment, CI/CD, or infrastructure changes.
- Changes that alter legal terms, privacy notices, consent handling, or data retention.

Approval Requirements

Approvals must be proportionate to risk.

Change Type	Minimum Approval
Standard low-risk change	Change owner or pre-approved runbook
Normal product change	Product/technical owner and code reviewer
Security-sensitive change	Technical owner and security/control reviewer
Financially sensitive change	Technical owner and finance/control reviewer
Database migration or backfill	Technical owner and data/control reviewer
Policy or legal document change	Policy owner and executive/legal reviewer where applicable
Emergency change	Incident commander or accountable executive, with post-change review

Approvers must have enough context to understand the change, risk, validation, and recovery path.

Development and Review Controls

CIC must maintain change controls for repositories and deployment workflows:

- Material code changes should be made on branches.

- Pull requests should describe scope, impact, test evidence, and deployment considerations.
- Code review should verify correctness, authorization, data integrity, security, error handling, regression risk, and user experience.
- Automated reviewer comments and material code review feedback must be resolved or formally dispositioned before merge.
- Branch protection, required checks, or compensating review procedures should be used for production-impacting repositories.
- Agents or automation must not bypass review requirements unless explicitly authorized for an emergency.

Testing and Validation

Testing must be proportionate to change risk.

Expected validation may include:

- Unit tests for changed logic.
- Integration tests for API and service behavior.
- Browser tests for user-facing flows.
- Negative tests for authorization, invalid inputs, insufficient funds, duplicate actions, and unavailable dependencies.
- Regression tests for previously broken behavior.
- Data migration dry runs and validation queries.
- Build and deployment tests.
- Manual verification of critical UI flows and accessibility basics.

High-risk CIC flows require stronger validation, including:

- Grant creation, publishing, funding, joining, applications, voting, and disbursement.
- Pitch investment and equity calculations.
- Wallet movement, withdrawals, and fee handling.
- PFS ledger posting, reversal, reporting, and access control.
- Organization context switching.
- Profile visibility and public profile behavior.
- Support identity verification, file request, and screen sharing flows.
- Messaging and notifications.

Data Change Controls

Database migrations, repair scripts, and backfills must include:

- Target environment.
- Scope of records affected.
- Dry-run plan where practical.
- Idempotency plan.
- Rollback or recovery approach where practical.
- Validation query or report.

- Execution evidence.
- Owner and approver.

One-time scripts should be retained or documented with execution evidence so CIC can explain what changed, when it changed, and why it changed.

Configuration and Secrets Changes

Configuration changes must be controlled because they can materially affect production behavior.

Requirements:

- Production secrets must not be committed to repositories.
- Environment variables must be managed through approved environment or secret-management surfaces.
- Secret rotations must be documented.
- Configuration changes affecting deployment, payment, financial, identity, support, email, storage, or API provider behavior must have validation evidence.
- Failed deployment or runtime configuration changes must be investigated from logs before repeated reruns.

Third-Party and Vendor Changes

Changes to vendors or third-party integrations must be reviewed for:

- Data shared with the provider.
- Authentication and credential model.
- Webhooks and callback security.
- Financial or user-impacting consequences.
- Contractual or compliance obligations.
- Monitoring and error handling.
- Vendor outage or fallback behavior.

Examples include payment providers, financial API providers, identity verification services, cloud hosting, email/SMS providers, analytics, and support tooling.

Release and Deployment Controls

Production deployments must be traceable to approved code and configuration.

CIC should maintain:

- Deployment logs.
- Build logs.
- Commit or PR reference.
- Environment and version information.
- Deployment status.
- Post-deployment validation result.

For high-risk releases, CIC should define:

- Release window.
- Rollback plan.
- Monitoring window.
- Support readiness.
- User communication plan where applicable.

Post-Implementation Review

High-risk, emergency, failed, or incident-related changes must receive post-implementation review.

The review should confirm:

- Whether the change achieved the intended outcome.
- Whether any incident, defect, regression, or user impact occurred.
- Whether the rollback plan was sufficient.
- Whether monitoring detected issues.
- Whether support documentation or customer communication is needed.
- Whether policy, runbook, test, or control updates are required.

Change Freeze and Restricted Windows

CIC may establish restricted change windows during:

- Major releases.
- Critical financial operations.
- Known market or customer events.
- Incident response.
- Independent audits.
- Production API certification reviews.

Emergency changes may proceed during restricted windows with appropriate approval and documentation.

Communication Requirements

Changes that affect users, support, operations, legal obligations, financial workflows, or customer trust must include a communication plan.

Communication may include:

- Release notes.
- Support briefing.
- Internal operating note.
- User-facing notice.
- Status-page update.
- Policy update notice.
- Partner or vendor communication.

Evidence and Records

CIC should retain evidence for material changes, including:

- Issue, ticket, or change record.
- Pull request.
- Review and approval record.
- Test results.
- Browser/UAT screenshots for user-facing flows.
- Build and deployment logs.
- Migration or backfill output.
- Rollback or recovery documentation.
- Post-deployment verification.
- Post-implementation review for high-risk or emergency changes.

Exceptions

Exceptions to this policy must be documented with:

- Reason for exception.
- Scope.
- Risk acceptance.
- Compensating control.
- Owner.
- Expiration date.
- Approval.

Permanent bypasses of review, testing, approval, logging, or deployment controls are not allowed without executive risk acceptance.

Roles and Responsibilities

Role	Responsibility
Change owner	Defines scope, risk, validation, deployment, and recovery plan.
Technical reviewer	Reviews implementation correctness, security, data integrity, and regression risk.
Product owner	Confirms user workflow, business requirement, and customer impact.
Security/control reviewer	Reviews security-sensitive and control-impacting changes.
Finance/control reviewer	Reviews financially sensitive changes.
Operations/support owner	Prepares support readiness and user communication where needed.

Role	Responsibility
Executive approver	Approves high-risk, emergency, or risk-accepted exceptions.

Metrics and Monitoring

CIC should monitor:

- Number of changes by type and risk level.
- Failed deployments.
- Emergency changes.
- Changes causing incidents or rollbacks.
- Open exceptions.
- Aging remediation items.
- Unreviewed or stale pull requests.
- Post-deployment validation failures.
- Repeated defects in the same workflow.

Review

This policy must be reviewed at least annually and after material incidents, audit findings, major architecture changes, or changes to CIC's operating model.