SECURITY

Secure Development Policy

Policy Owner: Paul Jones

Version: 2.1

Effective Date: 2022-07-05

Last Reviewed: 2026-02-24

1. Purpose

To ensure that security is integrated into the design, development, testing, and deployment of Crystal Project Inc systems and applications.

2. Scope

This policy applies to:

- All internally developed software and infrastructure
- All business-critical systems
- Any systems that process, store, or transmit Confidential or Customer data
- All engineers and contractors contributing to Crystal Project systems

3. Development Model

Crystal Project follows a continuous integration and continuous deployment (CI/CD) model.

Security controls are embedded directly into the engineering workflow through:

- Version-controlled source code
- Automated testing
- Automated security and dependency scanning
- Branch-based preview environments
- Pipeline-controlled production deployments

- Role-based production access controls

Security validation is enforced through technical controls rather than manual approval gates.

4. Source Code Management

All source code must:

- Be maintained in a centralized version control system (GitHub)
- Be attributable to an identified contributor
- Maintain full revision history
- Restrict direct modification of protected production branches via branch protection rules

All changes to production branches require peer code review and approval prior to merge. Branch protection is enforced at the platform level.

Automated dependency monitoring (Dependabot) is enabled on all production repositories. It generates pull requests for dependency updates and raises alerts for known vulnerabilities in third-party dependencies.

Access to repositories is role-based and periodically reviewed.

5. Change Management and Deployment Controls

All production changes must:

- Be committed through version control
- Pass defined automated test suites
- Pass configured CI validation checks prior to deployment
- Be deployed through approved deployment pipelines

If automated tests or CI checks fail, deployment is blocked until issues are resolved.

Manual release checklists are not required where automated controls provide equivalent or stronger validation.

Emergency changes may be deployed when necessary to restore service or mitigate security risk. Such changes must be documented and validated after deployment.

6. Testing and Security Validation

Security validation includes:

- Automated unit and integration tests
- Dependency vulnerability scanning
- Static analysis where applicable
- Infrastructure configuration validation
- Independent third-party penetration testing performed annually

Testing is integrated into CI pipelines and must pass before code reaches production.

7. Environment Architecture

Crystal Project uses:

- Local development environments for engineers
- Branch-based preview environments for feature validation
- Production environments deployed via automated pipelines

Traditional long-lived staging environments are not required where preview environments and automated test controls provide equivalent or stronger validation.

Production systems are logically separated from development workflows and protected by role-based access controls.

8. Platform and Dependency Management

Changes to core platforms, dependencies, or infrastructure components must be validated through testing and monitoring.

Vulnerabilities identified through scanning, monitoring, or penetration testing are tracked and remediated according to the severity-based SLAs defined in the Operational Security Policy.

9. Protection of Test Data

Production customer data must not be used in development or preview environments unless:

- Explicitly authorized
- Appropriately protected
- Operationally necessary

Test data must be handled in accordance with contractual and regulatory obligations.

10. Outsourced Development

External contributors must:

- Use approved development workflows
- Follow repository and deployment controls
- Be subject to the same access and security restrictions as internal engineers

11. Exceptions

Exceptions to this policy must:

- Be documented
- Include risk justification
- Be approved by the Policy Owner

12. Violations and Enforcement

Violations of this policy may result in:

- Removal of system access
- Corrective action

- Disciplinary measures
- Termination of engagement where appropriate

13. Review and Revision History

Version	Date	Description	Author
1.1	2022-07-05	Reference updates	Paul Jones
2.1	2026-02-24	Aligned with CI/CD and preview-based architecture	Paul Jones