

CHECKLIST

Governance Checklist for No-Code Applications

40-point security, compliance, and lifecycle checklist for citizen-built applications. Use this before every production deployment to ensure your no-code apps meet enterprise standards.

CitizenDevelopers.app

1. Identity & Access Control

□ SSO enforced for all platform users

No personal email accounts. All citizen developers must authenticate through your organization's identity provider (Azure AD, Okta, Google Workspace).

□ Role-based access configured

Define at least 4 roles: Viewer (read-only), Builder (create/edit own apps), Admin (manage team apps), IT Governance (full oversight). Document who has each role.

□ MFA enabled for all builder accounts

Multi-factor authentication is non-negotiable for anyone with the ability to create or modify applications.

□ De-provisioning workflow documented

When an employee leaves or changes roles, what happens to their apps? Document the transfer/archive process and test it quarterly.

□ Service account inventory maintained

If apps use service accounts or API keys, maintain a central registry with owners, purposes, and expiration dates.

2. Data Governance

□ Data classification applied to all data sources

Every data source used by citizen-built apps must be classified: Public, Internal, Confidential, or Restricted. Restricted data requires IT involvement.

□ PII handling reviewed and documented

If the app collects, processes, or displays personally identifiable information, document what PII, why it is needed, how it is stored, and who can access it.

□ Data retention policy defined

How long does the app store data? When is it archived or deleted? This must align with your organization's data retention policies and regulatory requirements.

□ Cross-border data flows identified

If the platform stores data in a different country than your users, document the data flow for GDPR/privacy compliance.

□ Data backup and recovery tested

Can you restore the app's data if something goes wrong? Test the backup/recovery process before going to production. Document the RTO and RPO.

3. App Development Standards

□ App registered in the central catalog

Every citizen-built app must be registered with: name, owner, department, description, data sources used, user count, and business criticality (Low/Medium/High).

□ Naming convention followed

Use a consistent naming convention: [Department]-[Function]-[Version]. Example: HR-OnboardingTracker-v1.

□ Development environment separated from production

Never build directly in production. Use dev/staging environments for testing, then promote to production through a documented process.

□ Version history maintained

The platform should maintain version history so you can roll back to a previous version if a deployment causes issues.

Error handling implemented

What happens when the app encounters an error? Missing data? API timeout? Handle edge cases gracefully with user-friendly error messages.

Input validation configured

All user inputs must be validated: required fields, data types, length limits, format patterns (email, phone, dates). Never trust user input.

Mobile responsiveness tested

If the app will be used on mobile devices, test on at least 3 device sizes (phone, tablet, desktop).

4. Security Controls

Authentication flow tested

Test the login flow end-to-end. Verify that unauthorized users cannot access the app. Test what happens when a session expires.

Authorization rules verified

Test that each user role can only see and do what they should. A viewer should not be able to edit. A builder in Department A should not see Department B data.

API endpoints secured

If the app connects to APIs, verify that API keys are stored securely (not hardcoded), connections use HTTPS, and rate limiting is configured.

File upload restrictions configured

If users can upload files: restrict file types, limit file sizes, scan for malware if possible. Never allow executable file uploads.

SQL injection / code injection prevented

If the platform allows custom queries or formulas, ensure user inputs are parameterized and sanitized to prevent injection attacks.

Sensitive data not exposed in URLs or logs

Review that API keys, passwords, PII, and other sensitive data do not appear in browser URLs, application logs, or error messages.

HTTPS enforced

All app traffic must use HTTPS. No exceptions. Most no-code platforms handle this by default, but verify.

5. API & Integration Governance

All external API connections documented

List every external service the app connects to: service name, purpose, data exchanged, contract/SLA, security posture, and owner.

API keys rotated on schedule

API keys should be rotated at least quarterly. Set calendar reminders. Use secrets management (environment variables, vaults) — never hardcode keys.

Rate limiting configured

Prevent citizen-built apps from overwhelming external APIs. Set reasonable rate limits and implement exponential backoff for retries.

Webhook endpoints validated

If the app receives webhooks, validate the source (signature verification), process payloads safely, and handle webhook replay/deduplication.

Third-party vendor security reviewed

Any third-party service connected to a citizen-built app must meet your organization's security standards. Review SOC2 reports, privacy policies, and DPAs.

6. Testing & Deployment

Happy path tested with real data

Walk through the primary workflow with realistic data. Does every step work as expected? Are all notifications sent correctly?

Edge cases tested

Test with: empty fields, maximum-length inputs, special characters, concurrent users, slow network conditions. What happens at boundaries?

Permission boundaries tested

Log in as each role type and verify access controls. Attempt to access data or functions outside your role. Document any gaps.

Load testing performed (if >50 users)

For apps expected to have more than 50 concurrent users, perform basic load testing.

Rollback plan documented

If the deployment fails or causes issues, how do you roll back? Document the specific steps and verify they work.

End-user training delivered

Users should receive training before the app goes live: walkthrough, FAQ document, and a designated support contact.

7. Compliance & Documentation

Privacy impact assessment completed

If the app handles personal data, complete a Privacy Impact Assessment (PIA). Many organizations have a template — adapt it for citizen-built apps.

Accessibility requirements met

Verify basic accessibility: keyboard navigation, screen reader compatibility, sufficient color contrast, alt text on images. WCAG 2.1 AA is the standard.

Regulatory requirements identified

Does the app fall under any industry regulations (HIPAA, SOX, PCI-DSS)? If yes, involve your compliance team before deployment.

Documentation completed

Every production app needs: user guide, admin guide, data flow diagram, and a list of known limitations.

App owner succession plan documented

If the app owner leaves the organization, who takes over? Document a backup owner and ensure they have the necessary access and knowledge.

8. Post-Launch Operations

Monitoring and alerting configured

Set up alerts for: app errors, unusual usage patterns, data threshold breaches, and API failures.

☐ User feedback channel established

Create a simple way for users to report issues and suggest improvements. A shared Slack channel or feedback form works well.

☐ Quarterly health review scheduled

Every 90 days: review usage metrics, check for unused features, validate data integrity, update documentation.

☐ Annual security re-certification

Once per year, re-run the security checklist for all production apps. Technologies change, threats evolve, and standards update.

This checklist is maintained by CitizenDevelopers.app. Subscribe to our newsletter for updates and new governance resources.