TIDEWELL FINANCIAL

Technology Governance Program

DevOps and Secure Software Delivery Standard

Standard No. TGP-STD-360

Version 2.0 | Effective October 1, 2026

Portfolio writing sample by David Parker Jr. Tidewell Financial is a fictional organization; all names, document numbers, and requirements in this document are invented for illustration.

Contents

1 Purpose
2 Scope and Authority
3 Adherence Reporting
4 Compliance
5 Exceptions and Risk Acceptance
6 References
7 Standard
7.1 Source Code Management
7.2 Code Review
7.3 Build Pipelines
7.4 Security Testing
7.5 Secrets and Credentials
7.6 Deployment and Release
7.7 Environments and Infrastructure as Code
7.8 AI-Assisted Development
7.9 Applications with AI Components
7.10 Monitoring and Reporting
8 RACI Matrix
9 Terms and Definitions
10 Reviewers and Approvers
11 Revision History
Appendix A Vulnerability Remediation Timeframes
Appendix B Data Permitted in AI Tools

1 Purpose

This Standard sets the minimum requirements for how Tidewell Financial builds, tests, and releases software. It is meant to let teams deliver changes frequently and safely: every change is reviewed, tested, and traceable from source code to production, and AI tools used in development or built into applications are used within defined limits.

2 Scope and Authority

This Standard is issued under the authority of the *Technology Governance Policy (TGP-POL-100)*. It applies to all software, scripts, and infrastructure configuration developed by or for Tidewell Financial that is deployed to a production environment, including work performed by contractors and third-party development partners.

Section 7.8 applies to anyone who uses AI tools to write, change, test, or review code or configuration. Section 7.9 applies to any Tidewell application that sends data to, or acts on output from, an AI model in production. Commercial software that Tidewell configures but does not develop is governed by the Software Asset and License Management Standard (TGP-STD-350).

3 Adherence Reporting

The Standard owner must be able to show that the requirements in Section 7 are being met. Platform Engineering will test the evidence in Figure 1 each quarter and report the results to the Technology Governance Committee. Where possible, evidence is collected automatically from repository and pipeline settings.

Figure 1. Adherence evidence

Requirement area	Evidence tested
Source code and review (7.1, 7.2)	Repository settings export shows branch protection on all production repositories; sample of merged pull requests shows approval by a reviewer other than the author.
Build pipelines and security testing (7.3, 7.4)	Pipeline configuration shows required scans on every pull request; SBOMs exist for each production release in the period.
Secrets (7.5)	Secrets scanning report shows no unresolved secrets in production repositories older than 24 hours.
Deployment and release (7.6)	Sample of production deployments traces to a pipeline run and an approved change record.
AI-assisted development (7.8)	AI tools in use match the Approved Software Catalog; sample of AI agent identities shows no production credentials; pull request templates include the AI-assistance disclosure.
Applications with AI components (7.9)	Each production application that uses an AI model has an AI Use Inventory entry and a completed AI risk assessment.

4 Compliance

All new applications and repositories must meet this Standard before their first production release. Existing applications must also comply. Where an existing application does not comply, the Application Owner must submit a remediation plan to Platform Engineering within 30 days, listing the actions, owners, and target dates needed to reach compliance. Requirements added in version 2.0 (Sections 7.8 and 7.9) must be met by January 31, 2027.

5 Exceptions and Risk Acceptance

When a requirement cannot be met, the Application Owner must request an exception through the Technology Risk exception process before the non-compliant activity begins. Each request must describe:

- the requirement that cannot be met and why;
- the compensating controls that will reduce the resulting risk;
- the plan and target date for reaching compliance.

Exceptions are granted for no more than 12 months. A Senior Director or above who is accountable for the affected application must accept the residual risk in writing. Exceptions to Section 7.9 also require review by the AI Governance Office.

6 References

This Standard must be read and maintained in conjunction with the following documents.

Tidewell Financial documents

- TGP-POL-100 Technology Governance Policy
- TGP-STD-310 IT Change Management Standard
- TGP-STD-350 Software Asset and License Management Standard
- TGP-STD-380 AI Governance Standard
- ISP-STD-210 Data Classification and Handling Standard

External frameworks

- NIST SP 800-218, Secure Software Development Framework (SSDF)
- NIST SP 800-218A, Secure Software Development Practices for Generative AI and Dual-Use Foundation Models
- NIST AI 100-1, Artificial Intelligence Risk Management Framework
- OWASP Top 10 for Large Language Model Applications

7 Standard

7.1 Source Code Management

7.1.1 All source code, pipeline definitions, and infrastructure configuration must be stored in an enterprise source code repository approved by Platform Engineering.

7.1.2 The main branch of every production repository must be protected so that changes can be merged only through an approved pull request. Direct commits to protected branches are not permitted.

7.1.3 Access to production repositories must require single sign-on with multi-factor authentication. Repository access must be reviewed every six months by the Application Owner.

7.2 Code Review

7.2.1 Every pull request to a protected branch must be approved by at least one qualified reviewer who is not the author.

7.2.2 Changes to authentication, authorization, encryption, or payment-processing code must be approved by two reviewers, one of whom must be a designated security champion or a member of Information Security.

7.2.3 Reviewers must not approve a pull request until all required automated checks have passed.

7.3 Build Pipelines

7.3.1 Software deployed to production must be built by an approved build pipeline. Artifacts built on a developer’s workstation must not be deployed to production.

7.3.2 Pipeline definitions must be stored as code in version control, and changes to them must follow Section 7.2.

7.3.3 Build artifacts must be stored in an approved artifact registry, and each artifact must be traceable to the commit and pipeline run that produced it.

7.4 Security Testing

7.4.1 Every pull request to a production repository must pass static application security testing (SAST), software composition analysis of dependencies, and secrets scanning.

7.4.2 New Critical and High findings must block the merge unless an exception is approved. Existing findings must be remediated within the timeframes in Appendix A.

7.4.3 Container images must be scanned before they are pushed to the artifact registry.

7.4.4 The build pipeline must generate an SBOM for each production release and store it with the release artifact.

7.5 Secrets and Credentials

7.5.1 Secrets must not be stored in source code, pipeline definitions, or container images. They must be retrieved at run time from the enterprise secrets vault.

7.5.2 Any secret detected in a repository must be treated as exposed. It must be revoked and replaced within 24 hours, even if the commit has been removed.

7.5.3 Pipeline service accounts must be non-human identities granted only the permissions needed for the stages they run.

7.6 Deployment and Release

7.6.1 Production deployments must be performed by the build pipeline. Manual changes to production systems are permitted only as emergency changes under the IT Change Management Standard (TGP-STD-310).

7.6.2 Each production deployment must be linked to an approved change record. Pipeline deployments that meet the CAB-approved criteria for automated releases may be recorded as Standard changes.

7.6.3 The author of a change must not be the only person who approves its production deployment.

7.6.4 Every production release must have a tested rollback or roll-forward method that can be carried out within the approved change window.

7.7 Environments and Infrastructure as Code

7.7.1 Cloud infrastructure for production must be defined as code and deployed through a build pipeline.

7.7.2 Platform Engineering must run drift detection at least daily and must correct or record any difference between deployed infrastructure and its code definition within five business days.

7.7.3 Restricted data must not be copied to non-production environments unless it has been masked or de-identified using a method approved by Information Security.

7.8 AI-Assisted Development

7.8.1 Only AI tools listed as Approved or Restricted in the Approved Software Catalog (TGP-STD-350) may be used for development work. Consumer versions of AI tools must not be used for Tidewell work, even if an enterprise version of the same tool is approved.

7.8.2 Approved AI tools must be configured, and contracted, so that Tidewell prompts, code, and outputs are not kept by the vendor beyond the session or used to train the vendor’s models.

7.8.3 Data entered into an AI tool must be limited to the classifications permitted in Appendix B. Secrets must never be entered into an AI tool.

7.8.4 The developer who submits AI-generated code is accountable for it as if they had written it. They must understand what the code does and must test it before submitting it for review.

7.8.5 AI-generated code is subject to the same review and security testing as all other code. The pull request template must include a disclosure indicating whether a substantial part of the change was generated by AI, so that reviewers can adjust their review.

7.8.6 Where an approved AI coding tool offers a filter that blocks suggestions matching public code, the filter must be turned on.

7.8.7 An AI tool may assist a human reviewer but must not be the only reviewer or approver of a pull request.

7.8.8 AI agents that take actions on their own must:

- run under a dedicated non-human identity, not a developer’s personal account;
- be limited to development and test environments, with no access to production credentials or Restricted data;

- be unable to approve or merge their own pull requests;
- have their actions logged and retained for at least one year.

7.9 Applications with AI Components

7.9.1 Before its first production release, any application that uses an AI model must be registered in the AI Use Inventory with its purpose, the model and provider used, the data the model receives, and the Application Owner.

7.9.2 The Application Owner must complete an AI risk assessment as required by the AI Governance Standard (TGP-STD-380) before production release, and again before any material change.

7.9.3 Before production release, the application must be tested for prompt injection, disclosure of sensitive data, and unsafe handling of model output, using the OWASP Top 10 for LLM Applications as the minimum test scope.

7.9.4 A change to the model, model version, system prompt, or retrieval data source that could materially change the application's output must be treated as a change under TGP-STD-310 and must be tested before release.

7.9.5 Model output must not trigger a financial transaction, an account change, or a decision about a customer without either human approval or controls documented in the AI risk assessment.

7.9.6 Model inputs and outputs must be logged in a way that supports monitoring and investigation, and logs must be retained and protected according to the classification of the data they contain.

7.10 Monitoring and Reporting

7.10.1 Platform Engineering must report the following metrics to the Technology Governance Committee each quarter: deployment frequency, change failure rate, time to restore service, and the number of Critical and High vulnerabilities open beyond their Appendix A timeframe.

7.10.2 The AI Governance Office must report each quarter on the number of active AI coding tool users, AI Use Inventory entries, AI risk assessments completed, and any incidents involving AI tools or AI components.

8 RACI Matrix

Figure 2 identifies who is Responsible, Accountable, Consulted, and Informed for each section of this Standard.

Figure 2. Roles and responsibilities

Section	Dev Team	App Owner	Platform Eng	Info Security	AI Governance
7.1 Source Code Management	R	A	R	C	I
7.2 Code Review	R	A	I	C	I
7.3 Build Pipelines	C	I	A/R	C	I
7.4 Security Testing	R	A	R	C	I
7.5 Secrets and Credentials	R	A	R	C	I

Section	Dev Team	App Owner	Platform Eng	Info Security	AI Governance
7.6 Deployment and Release	R	A	R	I	I
7.7 Environments and IaC	R	I	A	C	I
7.8 AI-Assisted Development	R	A	R	C	C
7.9 Applications with AI Components	R	A	C	C	R
7.10 Monitoring and Reporting	I	I	A/R	C	R

R = Responsible A = Accountable C = Consulted I = Informed

9 Terms and Definitions

Term	Definition
AI agent	An AI tool that can take actions on its own, such as editing files, running commands, or opening pull requests, rather than only suggesting text or code.
AI-assisted development	Using an AI tool to write, change, explain, test, or review code or configuration.
Build pipeline	The automated process that builds, tests, scans, and packages code for deployment (also called a CI/CD pipeline).
Infrastructure as code (IaC)	Defining servers, networks, and cloud resources in version-controlled configuration files rather than configuring them by hand.
Non-human identity	An account used by a system, service, or AI agent rather than by a person.
Production repository	A source code repository whose contents are deployed to a production environment.
Pull request	A request to merge a set of code changes into a protected branch, which triggers review and automated testing.
SBOM	Software bill of materials: a list of the components and dependencies included in a software release.
Secret	A credential such as a password, API key, token, or private key.

10 Reviewers and Approvers

Role	Title
Standard Owner	Director, Platform Engineering
Reviewer	Manager, Application Security
Reviewer	Lead, AI Governance Office
Reviewer	Manager, Change Management Office
Approver	Chief Technology Officer

11 Revision History

Date	Version	Summary of change	Updated by	Approved by
06/16/2025	1.0	Initial release.	Platform Engineering	CTO
09/08/2026	2.0	Added Sections 7.8 (AI-Assisted Development) and 7.9 (Applications with AI Components), Appendix B, and AI metrics in 7.10.2. Added SBOM requirement (7.4.4).	Platform Engineering	CTO

Appendix A Vulnerability Remediation Timeframes

Timeframes begin when a finding is first reported by a scanning tool. Severity is based on the scanner's rating as adjusted by Information Security.

Severity	Remediation timeframe	Merge or release blocked?
Critical	15 calendar days	Yes
High	30 calendar days	Yes, for new findings
Medium	90 calendar days	No
Low	180 calendar days	No

Appendix B Data Permitted in AI Tools

Data classifications are defined in the Data Classification and Handling Standard (ISP-STD-210). Secrets must never be entered into any AI tool, regardless of classification.

Data classification	Examples	Approved AI coding assistant	Other approved enterprise AI tools	Unapproved or consumer AI tools
Public	Published documentation, open-source code	Permitted	Permitted	Not permitted for work use
Internal	Internal source code, architecture diagrams	Permitted	Permitted	Not permitted
Confidential	Security configurations, vendor contracts	Permitted for source code only	Only if approved for Confidential data	Not permitted
Restricted	Customer personal data, account data, secrets	Not permitted	Only if approved for Restricted data	Not permitted