appian

Application Delivery Governance Model

Introduction.

The Application Delivery Governance Model provides customers a lightweight, effective way to govern their Appian delivery teams. By implementing this model, Appian program owners ensure that applications are:

1. built according to best practices and
2. adhere to program-specific requirements and standards.

Designed by Appian’s customer success professionals, this model provides delivery teams with the standards, measurements, and process alignment needed to develop quality low-code applications, quickly and effectively.

Governance roles.

While this model can be implemented with slight variations based on organizational requirements and risk appetite, Appian strongly encourages the involvement of at minimum the following three roles in order to implement effective governance.

Program owner.

The Program owner wears a few different hats, but is ultimately the individual responsible for making sure the governance model is followed. They are the ultimate approval authority for governance standards, ensuring that project managers enforce standards, and making the “go forward” decisions at readiness checks. They also provide delivery teams with contextual information and awareness of the program’s direction.

Program architect.

The Program architect’s primary concern is executing the governance program through a series of lightweight reviews and readiness checks. They will ensure the high quality of your application.

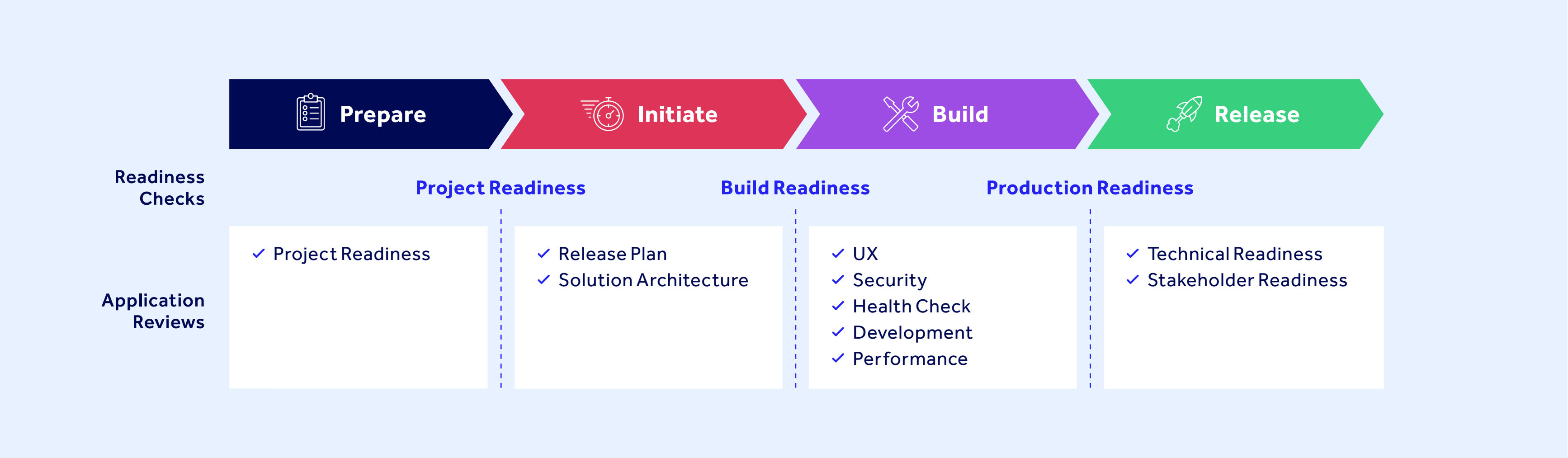
Tip: To get the most value out of your program architect, Appian encourages this role to be a part of the planning efforts early in the project lifecycle. The visibility will not only help prepare them for readiness checks throughout, but their domain expertise will also help navigate unforeseen obstacles in planning sessions.

Team lead(s).

Team leads are the day-to-day coordinators for application development ceremonies and serve as the main point of contact for delivery teams. They ensure their teams follow governance guidelines and standards set by the program owner and facilitate discussions between teams where needed. They also attend all readiness checks and reviews for their teams.

Readiness checks and application reviews.

Application delivery requires different types of governance attention across the application development lifecycle. Appian has condensed this lifecycle into four phases and mapped a series of mandatory, lightweight readiness checks and application reviews to each phase: Prepare, Initiate, Build, and Release (see Figure 1). These readiness checks and application reviews ensure best practices and program standards are implemented consistently and that the application does not create any unnecessary or unforeseen organizational risk.



Prepare → Initiate → Build → Release

Readiness Checks: Project Readiness | Build Readiness | Production Readiness

Application Reviews:

- Prepare: ✓ Project Readiness
- Initiate: ✓ Release Plan, ✓ Solution Architecture
- Build: ✓ UX, ✓ Security, ✓ Health Check, ✓ Development, ✓ Performance
- Release: ✓ Technical Readiness, ✓ Stakeholder Readiness

Application reviews.

During each phase of the application development lifecycle, delivery teams engage in select reviews where their progress will be checked by an expert against outlined best practices and program-standards. These reviews are designed to be lightweight and favor discussion over documentation; although some reviews do require artifacts for facilitation.

Application reviews will result in a prioritized list of recommended next steps categorized by risk profile (High, Medium, Low). If there are findings in the "High" category, the team should address the recommendation and perform the review again. Teams should still address "Medium" and "Low" findings, but these should be done in parallel as development continues.

Readiness checks.

After a team is ready to advance the next phase of development, delivery teams engage in readiness checks. Here, program management will review the current status of all findings and determine whether the team is ready to advance to the next lifecycle phase. If a team is not deemed ready to proceed, program owners will provide teams with the steps needed to advance to the next phase. Delivery teams will continue this process until the application has been released.

Solution architecture review.

Goal: Mitigate architectural risk early in the process by modeling and reviewing foundational design decisions.

What to Bring: Completed Appian architecture document

When to Review: End of Initiate phase

Attendees: Team lead, project manager, program architect

- □ Non-functional requirements have been captured including:
  - Total users
  - Peak current users
  - Volume estimates for all key entities
- □ Platform team has been notified of any significant changes in usage patterns that may require a resizing of the infrastructure.
- □ Application design does not contain anti-patterns.
- □ User provisioning design has been established.
- □ Application security requirements have been collected, and the design accounts for role-based access to relevant data.
- □ All external system integrations have been identified and key design decisions determined including: transport mechanism, authentication, and logging.
- □ Reporting requirements have been collected.
- □ Application design adheres to organizational and platform-specific guidelines.

Development review.

Goal: Ensure the developed features are designed according to standards.

What to Bring: Demonstration of completed features

When to Review: As needed, but at least a few times every iteration

Attendees: Team lead, key developers, program architect

- □ Application objects adhere to Appian Development Standards and any platform-specific guidelines.
- □ Key interfaces are performant: most interactions should be under one second.
- □ Role-based security has been configured for the developed features.
- □ Reusable component catalog is being used according to the platform-specific guidelines.
- □ Team has performed database volume testing.
- □ Indexes have been created for key entities according to the Indexing Guidance.

UX review.

Goal: Ensure interfaces are intuitive, adhere to best practices, and are consistent across the platform.

What to Bring: Mockups or demonstration of developed UX

When to Review: During Initiate phase or during sprint 1 and again before the end of the Build phase.

Attendees: Team lead, key developers, program architect

- □ Interfaces are succinct and contain just what's needed. Any redundant or obvious labels should be removed.
- □ Key interfaces are performant: common interactions should be imperceptibly fast (less than 200ms) and most other interactions should be around one second.
- □ User experience is intuitive and responses make sense. The outcome of an action should match the expected result.
- □ User experience is consistent across the application and complies with Appian UX Guide, organizational and any platform-specific UX guidelines.

Health check review.

Goal: Ensure developed features adhere to development standards via the Health Check tool.

What to Bring: Health Check results from the development environment

When to Review: Every iteration

Attendees: Team lead, program architect

- □ Health Check tool has been executed on the development environment for the current iteration.
- □ All high-risk findings are correctly prioritized.

Technical readiness check.

Goal: Ensure the application is technically ready to release. Though the delivery team will have created a release-ready application throughout the Build phase, this is an opportunity for a final check.

What to Bring: Deployment plan, defect log

When to Review: During the Release phase

Attendees: Team lead, key developers

□ End users have performed hands-on testing of the application.

□ Team has addressed any critical feedback from user acceptance testing and that any outstanding application and system defects have been reviewed by the program owner.

□ Testing included all relevant user groups (personas, geographic regions, etc.).

□ Team has established a comprehensive plan for the deployment that includes timing, owners, tasks, dependencies and ensured that all the necessary supporting parties have awareness of their responsibilities (networking, DBAs, third party systems integrations, etc.).

□ A complete application package has been created for the release, including database scripts.

□ Team has verified that users have been provisioned in the appropriate authentication groups.

□ Team has verified results of performance testing of the system based on the concurrent user activity expected and that it was conducted with production-like data and data volumes.

- Ensure that load tests approximate the anticipated load on the entire environment/site across all major applications and that the tests accounted for peak usage days/weeks/months.
- Verify that hardware sizing has been adjusted based on real and anticipated production data and usage based on load testing.

Appian is the unified platform for change. We accelerate customers' businesses by discovering, designing, and automating their most important processes. The Appian Low-Code Platform combines the key capabilities needed to get work done faster, Process Mining + Workflow + Automation, in a unified low-code platform. Appian is open, enterprise-grade, and trusted by industry leaders. For more information, visit **appian.com**.