AIG SDLC DevOps Standard Operating Procedure

AIG SDLC DevOps Standard Operating Procedure

Contents

1. Version Control

Document Version	Description of Changes	Author(s)	Date Published
1.0	Initial Version of Policy Document	Naveen Jogi Weston Gaddis Casey Winzeler	Sep 2017
1.1	Minor edits and enhancements	Weston Gaddis Casey Winzeler	Oct 2017
1.2	Added Onboarding Process Details, Security Scanning Requirements section and edited Production Service Account Controls Section	Weston Gaddis Casey Winzeler	Nov 2017

2. Purpose

The purpose of the AIG Application Development DevOps Standard Operating Procedure (the "SOP") is to establish the baseline Software Development Lifecycle (SDLC) requirements to be followed by AIG Global Functions which include IT (GFs) and Business Units (BUs) that use the DevOps Mode 2 application development methodology.

The baseline SDLC requirements have considered regulatory requirements from BU Compliance Teams, Application Data Management (ADM), IT Security Controls, Federal Financial Institutions Examination Council (FFIEC) Application Development & Acquisition booklet as well as leading industry practices. This SOP and the included processes will be subject to a quarterly review process with revisions as necessary.

This SOP includes:
• Definitions of baseline application development, release tasks and artifact requirements.
• Described roles and responsibilities with regard to implementation, maintenance, and compliance with the SOP.

For purposes of this SOP, each application development, deployment or release task is required unless specified that it is only required under certain conditions. If that is the case, the task must be followed when the defined conditions are met.

3. Scope

The AIG DevOps Application Development SOP is an AIG Global SOP that applies to all AIG GFs and BUs involved in:
1) Developing new applications that will leverage the DevOps Mode 2 Standard Release Model OR migrating existing applications into the DevOps Mode 2 Standard Release Model; and
2) Enhancing, changing or maintaining applications that already leverage the DevOps Mode 2 Standard Release Model.

This SOP excludes:

• End-user application modifications that are permitted by existing application functionality and end-user permissions (e.g., a modification of SharePoint page by content administrator). ***NOTE*** *- Such modifications are subject to monitoring, control and/or testing procedures as determined by the application owner.*
• Production data modifications that modify content
• Retiring or disposing applications that currently leverage the DevOps Mode 2 Standard Release Model. This will be subject to existing AIG Agile Standard Operating Procedure

4. SDLC DevOps Mode 2 Standard Release Process - Application Project Track

The SOP establishes a track for application development, enhancements, and maintenance projects that leverage a DevOps Mode 2 Operating Model. The table below illustrates the SDLC for DevOps Mode 2 that must be followed for all application development, enhancements and maintenance projects under this standard.

Project Type		DevOps Mode 2 Requirements
1) Application Custom Development in DevOps Mode 2 (New Application, New Enhancement or new product) OR Migration of Existing Application into DevOps Mode 2		One-Time Activities: 1. Refer to AIG Agile Standard Operating Procedure for guidance on applicability of Investment Governance Lifecycle activities based on total project spend and strategic alignment 2. Complete DevOps Mode 2 Standard Release Certification Process 3. Establish DevOps Mode 2 Delivery Pipeline Capabilities and Conventions Ongoing Activities: 1. Follow **Error! Not a valid result for table.** Process 2. Adhere to DevOps Mode 2 Maintenance Requirements
2) Enhancing, changing or maintaining applications that are engaged in the DevOps Mode 2 Operations	**Standard Change - Recurrent / Well Known / Follow a pre-defined, relatively risk-free path**	Ongoing Activities: 1. Follow **Error! Not a valid result for table.** Process 2. Adhere to DevOps Mode 2 Maintenance Requirements
	Normal Change	Follow the "Normal" Change Management Standard[1]

[1] AIG Application Change Management Standard is defined in a separate document

GFs and BUs must also adhere to requirements detailed in the AIG Project, Program, Portfolio Management Policy, AIG Project Management Lifecycle Standard and by their respective Project Governance Board (PGB).

When product customization, custom application development, enhancements, and maintenance activities are undertaken by contracted third parties for AIG, GFs and BUs must require and confirm that contracted third party's application development practices adhere to the AIG Standards or adopt AIG accepted application development practices.

DevOps Mode 2 Standard Release Certification Process

This process aligns with AIG's documented deployment methods. All DevOps Mode 2 deployments are, by definition, recurrent, well-known, relatively risk-free and follow a pre-defined path. This aligns with the 'Standard' Change classification. As such, all DevOps Mode 2 deployments will follow the Standard deployment process as defined within this SOP[2]. BF and BU senior management is responsible for determining which applications qualify for DevOps Mode 2 Releases. Note that each DevOps Pipeline version must also be certified for use as a production deployment capability.

5. DevOps Mode 2 Delivery Pipeline Capabilities and Conventions

1. Continuous Integration Platform

   At AIG, the DevOps Engineering team has established a suite of integrated tools that make up the Continuous Integration (CI) platform. The following is a depiction of the Commercial IT CI ecosystem, provided for example purposes.

[2] A Standard Change is categorized by the following:

- Standard Changes are low risk changes that follow an established and approved process, and can thus be considered "pre-approved"
- Standard Changes include common enhancements such as periodic updates of reference information, application website content and styling changes, and certain types of application or operating system patches that have a well-understood impact and little to no risk
- To be considered Standard, these Changes must follow an approved deployment pattern (such as an approved DevOps Pipeline)
- Apart from the automated tests included within the approved deployment pattern, Standard Changes do not require any additional external approvals before deployment

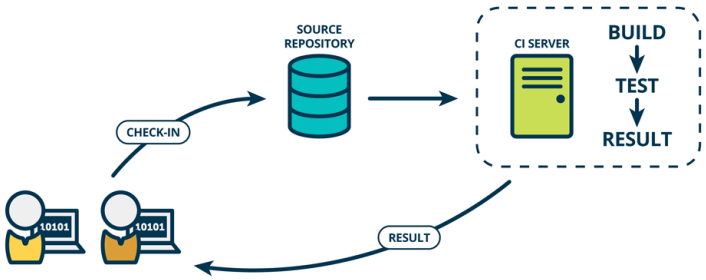


Figure 1: DevOps Mode 2 Continuous Integration (CI)[3]

When implemented correctly, the entire CI lifecycle is automated and delivers the following improvements to the overall SDLC:

a. Higher project quality
b. Higher code and product quality
c. Finds and corrects code problems faster
d. Better aligned with delivering software ready for release
e. Greater product release stability and reliability

See Terms and Definitions for further information on the CI Platform

2. Continuous Delivery Platform

The DevOps Engineering team has also established a suite of integrated tools that make up the Continuous Delivery (CD) platform. The following is a depiction of the Commercial IT CD ecosystem, provided for example purposes.

[3] https://www.gocd.org/2017/07/05/product-manager-guide-continuous-delivery.html

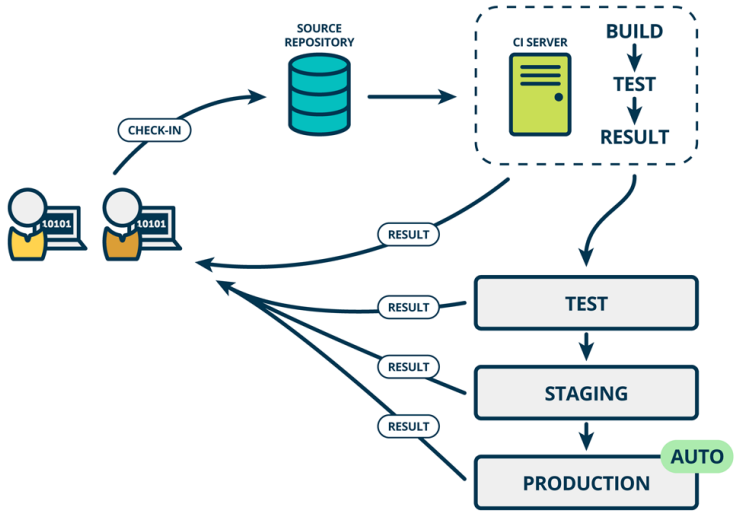


Figure 2: DevOps Mode 2 Continuous Delivery (CD)[4]

Continuous Delivery at AIG consists of a platform of integrated tools, like CI, that enables development teams the capability to define delivery pipelines based on their technology stack, testing practices and deployment requirements. Currently this consists of a mix of automated and manual activities with a trend towards complete automation over time. Complete automation within a CD process is commonly referred to as Continuous Deployment (see below). The DevOps Mode 2 Standard Release process is AIG's implementation of a Continuous Delivery practice.

3. Continuous Deployment Extension
   Continuous Deployment, as described above and in Terms and Definitions, is the complete automation of all activities executed by a Continuous Delivery pipeline.

4. Standard Naming Conventions

[4] https://www.gocd.org/2017/07/05/product-manager-guide-continuous-delivery.html

DevOps Engineering follows all relevant AIG design and development conventions and naming standards. For those not addressed in AIG's standard policies (see Appendix 1 – Security Scanning Workflow and Requirements

Static Scanning Workflow

Three distinct capabilities exist for static scanning within the Veracode platform. These are:

1. Veracode Greenlight Integrated Development Environment (IDE) plugin
2. Veracode Developer Sandbox
3. Veracode Policy Static Analysis

Figures 6-8 below provide an overview of where and how static scanning is performed for a DevOps-enabled applications.

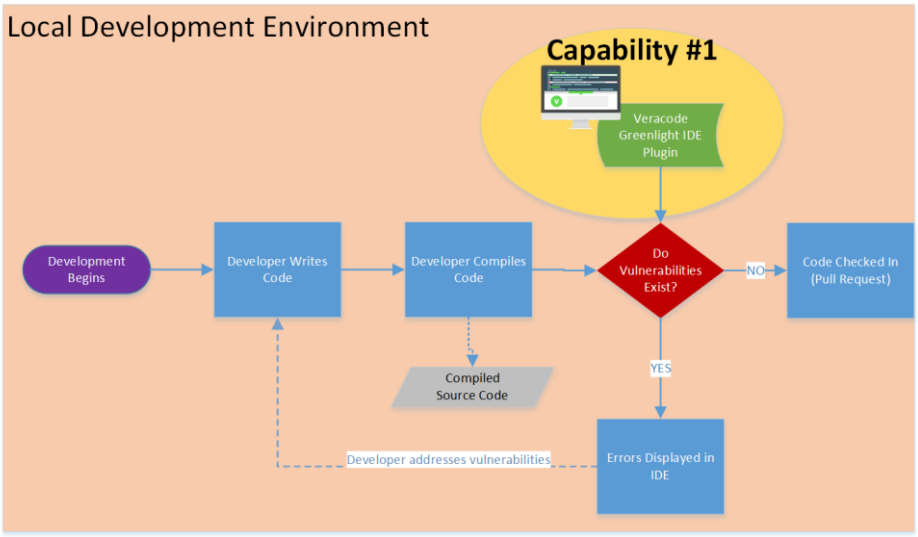


Figure 6: Veracode Greenlight IDE Plugin

Veracode Greenlight finds security defects in application code and provides contextual remediation advice to help developers fix issues in seconds, right in their IDE. This effectively pulls security scanning as far "left" as possible, making information about potential vulnerabilities available as early as possible in the development cycle. Developers do not need to provision any servers or tune the engine to use Greenlight. It simply scans in the background and provides accurate and actionable results, without taking up resources on a developer's local machine.

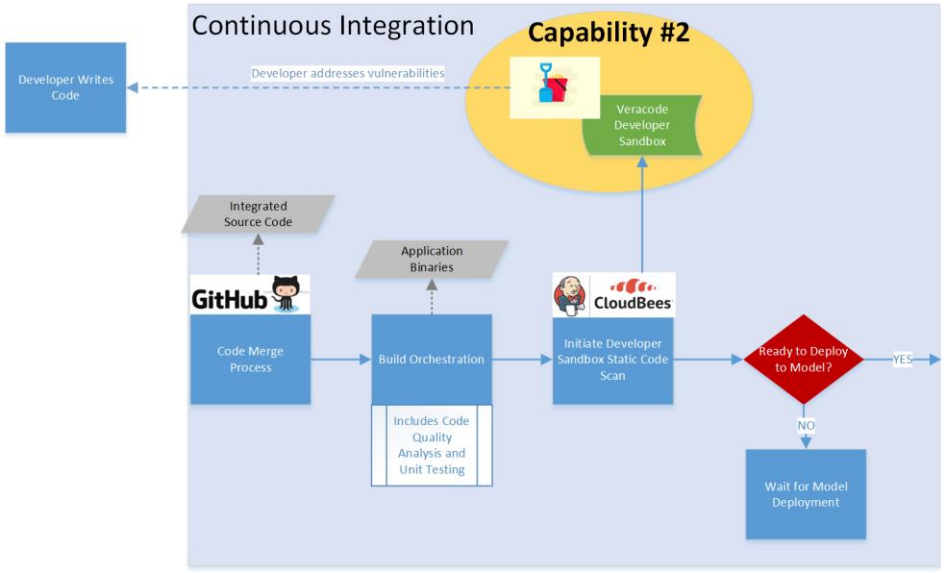


Figure 7: Veracode Developer Sandbox

During the Continuous Integration (CI) phase of the DevOps Mode 2 Standard Release Process, the Veracode Developer Sandbox is called to initiate static scanning. This capability leverages a static analysis scanning platform that assesses the security of micro services, web, mobile and desktop applications. Results from the Developer Sandbox are not formally reported, but are made available to developers to inform vulnerability remediation early in the release cycle. No action is required when security vulnerability is found at this stage, however vulnerabilities that are still present and that exceed the tolerable security risks as defined by security policy (see *Static Scanning Requirements*) must be addressed prior to a production deployment.

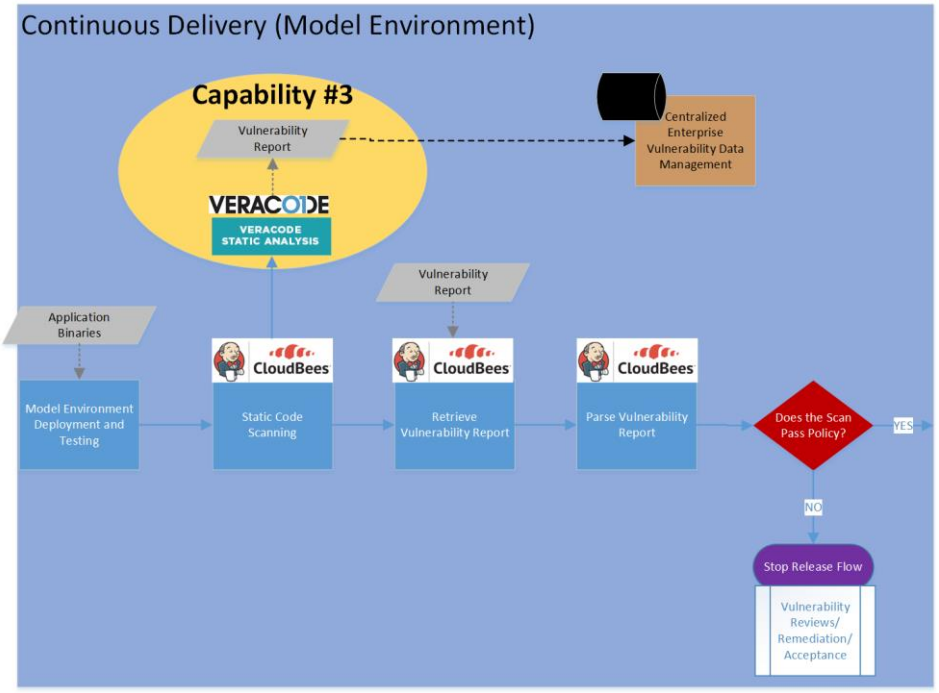


Figure 8: Veracode policy Static Analysis

During the Continuous Delivery (CD) phase of the DevOps Mode 2 Standard Release Process, Veracode policy Static Analysis is called to initiate static scanning. This scan is executed in parallel with the Model environment deployment, which acts as a pre-production environment and is a high probability release candidate for production. Note that this is the same underlying capability as the Developer Sandbox scan and that the exact same artifacts (e.g., application binaries) are scanned, so we expect to see the same results in terms of identified vulnerabilities. The difference at this step is that the output of the scan is reported and formally logged and tracked for the application.

The DevOps pipeline will automatically stop the release flow if vulnerabilities that are found exceed the risk thresholds set by security policy (see *Static Scanning Requirements*).

Static Scanning Requirements

The following static scanning requirements apply to DevOps-enabled applications:

1. Formal Static scanning is required prior to any Production release
2. DevOps Pipelines must stop the automated release flow if one or more Very High or High vulnerabilities is found during Policy Static Analysis
    a. Optionally, application teams may choose to prevent automated releases when a Medium vulnerability is present.

Any vulnerabilities that stop the release flow (per requirement #2) will initiate a manual review, remediation, and/or acceptance process that occurs outside of the DevOps Mode 2 Standard Release Process. If vulnerabilities are remediated, the process will begin again with the process as represented in **Figure 6: Veracode Greenlight IDE Plugin.**

Dynamic Scanning Workflow

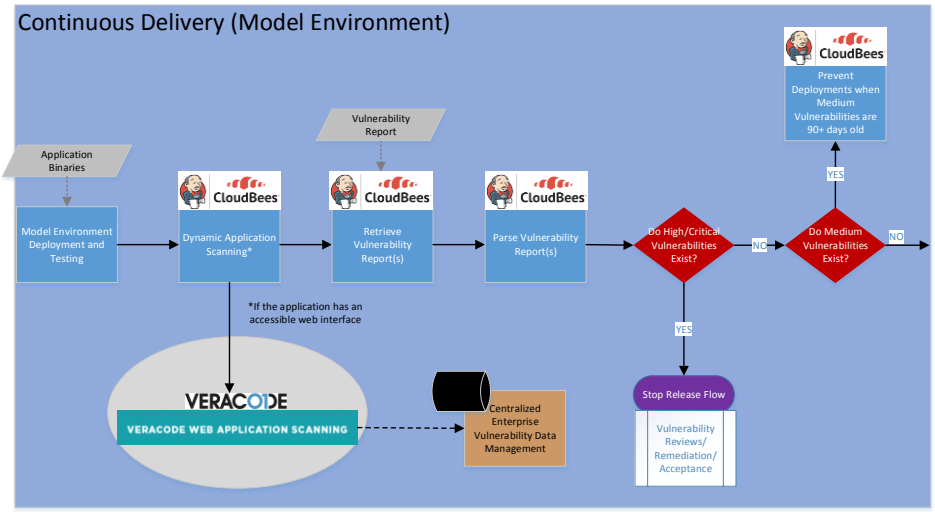


Figure 9: Dynamic Scanning Workflow

Applications with a web interface or a web service that can be invoked over the Internet are subject to dynamic scanning requirements. During the Continuous Delivery (CD) phase of the DevOps Mode 2 Standard Release Process, Veracode Web Application Scanning is leveraged for dynamic scanning. This scan is executed against the Model environment deployment, which acts as a pre-production environment and is a high probability release candidate for production.
The DevOps pipeline will automatically stop the release flow if vulnerabilities that are found exceed the risk thresholds set by security policy (see *Dynamic Scanning Requirements*).

Dynamic Scanning Requirements

The following dynamic scanning requirements apply to DevOps-enabled applications with a web interface or a web service that can be invoked over the Internet:

1. Dynamic scanning is required for a production release.
2. DevOps Pipelines must stop the automated release flow if one or more Very High or High vulnerabilities is found during Dynamic Scanning.
    a. Optionally, application teams may choose to prevent automated releases when a Medium vulnerability is present

Veracode Scan Results Review Process

Application teams must work with the SCSA team (SCSA-SourceCodeSecurityAssessmentTeam@aig.com) if results from security scans are believed to include false positives.

Additional Detail on Security Scanning Onboarding

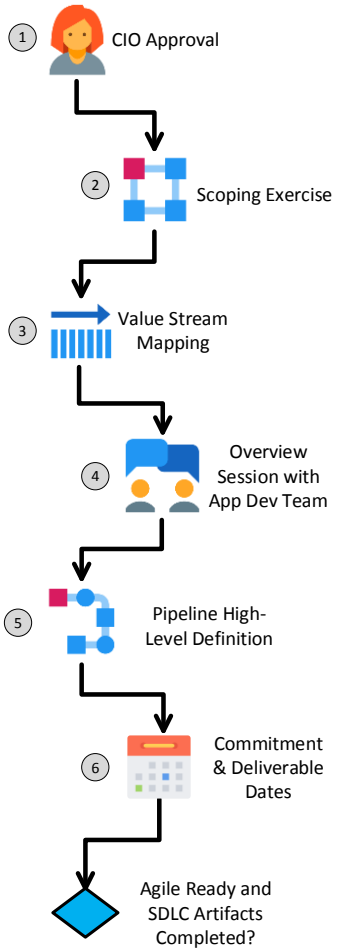
Onboarding to automated application static and dynamic security scanning capabilities (as described further in *Security Scanning Requirements*) will follow this process:

1. Commercial DevOps Engineering Team requests a new application for Veracode scanning through the SCSA team. This should be an application with active new or maintenance development.
    - As part of the request, DevOps Engineering provides application details (e.g., GEAR ID and Application Name) and a list of the application team developers (names and emails).
2. SCSA Team creates accounts for the developers within the Veracode platform and associates them with the application. Developers are emailed their credentials.
3. Application developers log into Veracode and generate API key credentials. Then developers install Veracode Greenlight IDE by following download instructions that are provided by the SCSA team. Application developers copy the newly created API key credentials into Greenlight. Note: API key credentials are time sensitive.
4. Application developers review Veracode training materials provided by SCSA team.
5. SCSA team enables Jenkins jobs to include scanning of the new application. Additional configuration work may be required for dynamic scanning to be properly configured for applications.
6. Commercial DevOps Engineering Team will work with application stakeholders to validate that users have access and that the scanning capabilities are working as expected.

Appendix 2 – Policy Alignment Mapping) DevOps attempts to follow emerging industry practices and conventions and are documented on the DevOps Engineering Confluence website.

DevOps Mode 2 Onboarding Process

Preparation and Decision Phase



1 – CIO Approval

Each application must receive CIO approval to onboard to a DevOps Lite or DevOps Complete operating model.

2 – Scoping Exercise

The DevOps Engineering team works with onboarding candidates to determine the target state operating model that best supports the application and business needs.

3 – Value Stream Mapping

Value Stream mapping provides an end-to-end, flow-based perspective of the current state application delivery metrics and capabilities. This also helps to identify and prioritize transformation activities.

4 – Overview Session with App Dev Team

Results of the Value Streaming will be discussed and each application team will learn more about the upcoming transformation activities.

5 – Pipeline High-Level Definition

The DevOps Engineering team works with the application stakeholders to define a high-level pipeline design based on the technical and operational components of the application. This includes determination of the end-state environments, and target technology components.

6 – Commitment & Deliverable Dates

Application teams are presented with a roadmap for the transformation, including details on the resource and time commitments that will be required to achieve the transformation vision.

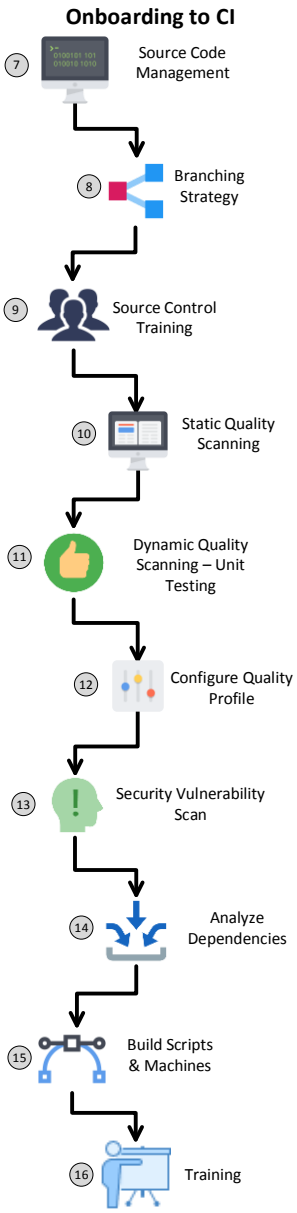
Agile Ready?

Teams should be utilizing Agile/Scrum/Kanban to manage their development and testing activities and be positioned to deliver product to Production frequently. If not, they

will not be able to continue on-boarding onto the DevOps pipeline until they have transitioned to Agile practices.

All required Project Initiation activities/artifacts from the SDLC process must also be complete at this point.

If steps 1-6 have been completed successfully, and the application team has committed to the transformation, then we can begin the DevOps journey.



7 – Source Code Management

Establishment of an integrated Source Code Management (SCM) to track, compare, store, and merge code revisions to form a release candidate.

8 – Branching Strategy

The DevOps Engineering team works with application teams to establish best practice branching and merging strategies taking advantage of the pipeline capabilities.

9 – Source Control Training

Application Teams are exposed to new strategies for effectively controlling source control in a DevOps operating model.

10 – Static Quality Scanning

Static Code Quality capabilities are established and baselined for the application team.

11 – Dynamic Quality Scanning - Unit Testing

The DevOps Engineering team works with the application team to establish a baseline of Unit Tests that will integrate into the automated CI build cycle. These tests provide rapid validation that a deployment doesn't break core application capabilities.

12 – Configure Quality Profile

Application teams must establish the quality gates for code coverage and defects that will be used to control flow from each stage/environment of the pipeline

13 – Security Vulnerability Scan

Security scanning is integrated into developer environments and the CI pipeline to provide static security scanning. Additional detail in Appendix 1.

14 – Analyze Dependencies

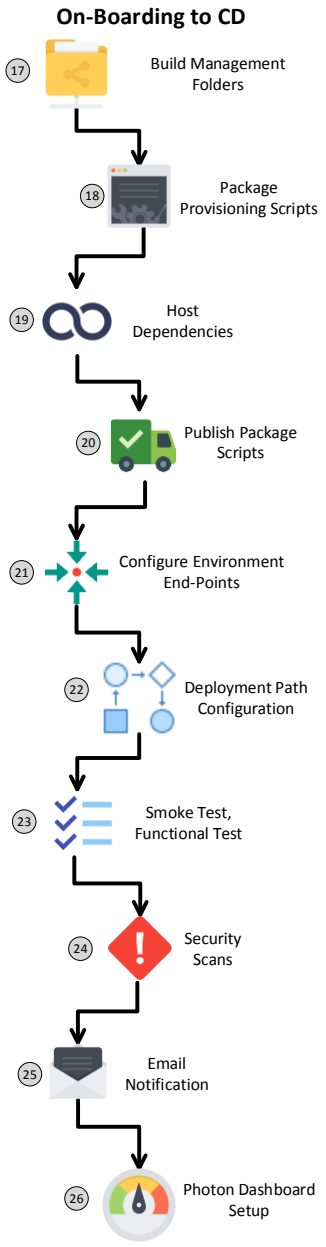
The DevOps Engineering team works with the application team to define and analyze deployment dependencies so that these can be automated.

15 – Build Scripts & Machines

Establish build automation scripting and supporting technologies to execute these tasks dynamically.

16 – Training

Application teams are provided multiple training opportunities and access to the AIG DevOps community.



17 – Build Management Folders

Build Management Folders must be established to effectively manage artifacts and versions for deployments.

18 – Package Provisioning Scripts

The DevOps Engineering team develops scripts for deployments of the application binaries and packages.

19 – Host Dependencies

The DevOps Engineering team works with the application team to identify and define host dependencies.

20 – Publish Package Scripts

Scripts developed in step #18 are published to the production pipeline capabilities so that they are accessible for upcoming deployments.

21 – Configure Environment End-Points

Application end-points that require additional configuration actions are identified and the configurations are documented for automated activities.

22 – Deployment Path Configuration

Application deployment paths are set up within the CD pipelines. This includes setting the appropriate target environment locations and parameters.

23 – Smoke Test, Functional Test

The application team establishes automated smoke and functional testing capabilities to check the health of a deployment made through the CD pipeline.

24 – Security Scans

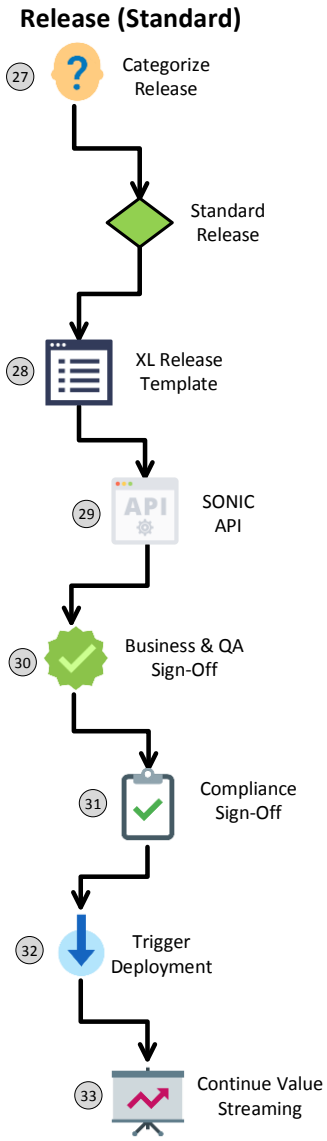
Security configuration work enables capabilities such as dynamic application scanning to be initiated automatically from the CD Pipeline after a deployment has completed.

25 – Email Notification

DevOps Engineering Team sets up the CD Pipeline to automatically generate notification emails concerning scheduled deployments and deployment initiation activities.

26 – Photon Dashboard Setup

The DevOps Engineering team creates a Photon dashboard for the application. This dashboard provides key operational and pipeline metrics to the team on an on-going basis.



27 – Categorize Release

Application teams must categorize the release as Standard or Normal. This includes identifying risk level of the associated Change by using a Consequence and Likelihood matrix. Follow this link for more information.

Standard Release

Standard Releases are recurrent, well-known, relatively risk-free, and follow a pre-defined path (such as a CD pipeline)

28 – XL Release Template

XL Release serves as the release orchestration engine. Application teams log into the tool and schedule a release. Once initiated, XL Release orchestrates manual and automated tasks required for a production deployment.

29 – SONIC API

The CD Pipeline automatically creates, updates, and closes a SONIC Change Ticket so that the application team does not need to perform this task manually.

30 – Business & QA Sign-Off

Releases must be approved by allowed QA and Business (OO) approvers. This approval activity is highly streamlined to support frequent releases.

31 – Compliance Sign-Off

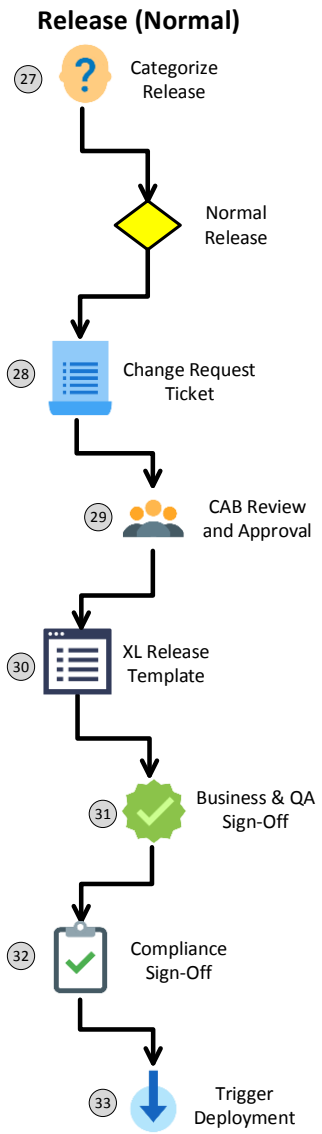
The Compliance sign-off activity provides a highly streamlined check for required artifacts, approvals, and a validation of the application's security/vulnerability status.

32 – Trigger Deployment

The CD Pipeline triggers an automated deployment during the scheduled deployment window. This deployment leverages production account controls, and includes automated smoke testing to validate a successful build.

33 – Continue Value Streaming

Value Streaming is an important component of all DevOps releases, continually providing feedback on performance improvements and additional focus areas for ongoing transformation activities.



27 – Categorize Release

Application teams must categorize the release as Standard or Normal. This includes identifying risk level of the associated Change by using a Consequence and Likelihood matrix. Follow this link for more information.

Normal Release

Normal Releases are unique, have an uncertain outcome, and are associated with possible risk.

28 – Change Request Ticket

Application teams must manually create Change Request tickets for Normal Releases. This initiates a manual review and approval workflow.

29 – CAB Review and Approval

Normal Changes may be subject to review by the Change Advisory Board (CAB) based on risk and other technology factors.

30 – XL Release Template

XL Release serves as the release orchestration engine. Application teams log into the tool and schedule a release. Once initiated, XL Release orchestrates manual and automated tasks required for a production deployment.

31 – Business & QA Sign-Off

Releases must be approved by allowed QA and Business (OO) approvers. This approval activity is highly streamlined to support frequent releases.

32 – Compliance Sign-Off

The Compliance sign-off activity provides a highly streamlined check for required artifacts, approvals, and a validation of the application's security/vulnerability status.

33 – Trigger Deployment

The CD Pipeline triggers an automated deployment during the scheduled deployment window. This deployment leverages production account controls, and includes automated smoke testing to validate a successful build.

Additional Controls for Production Deployments

DevOps Mode 2 Releases apply Privileged Access Management (PAM) best practices to ensure that production-affecting actions are secure, controlled and highly auditable[5]. For example, service accounts used for production deployments may be subject to a password refresh policy that. Mode 2 Release pipelines will retrieve the current production service account credentials from an AIG approved PAM solution (e.g., the TPAM system) during the production deployment window. This allows for "just in time" validation of privileges/permissions to execute a deployment, and helps to ensure that production service accounts are only used for approved deployment actions.

The workflow in **Figure 3** below provides more detail on the TPAM integration for DevOps Mode 2 Release Pipelines. This integration initiates within the Release Orchestration phase of the DevOps Mode 2 Standard Release flow. This ensures that service account credentials are only retrieved when a compliant release candidate is ready for production deployment. The workflow identifies multiple interaction points, which include passwords and key files. These sensitive data elements are protected by access restrictions to prevent unauthorized use.

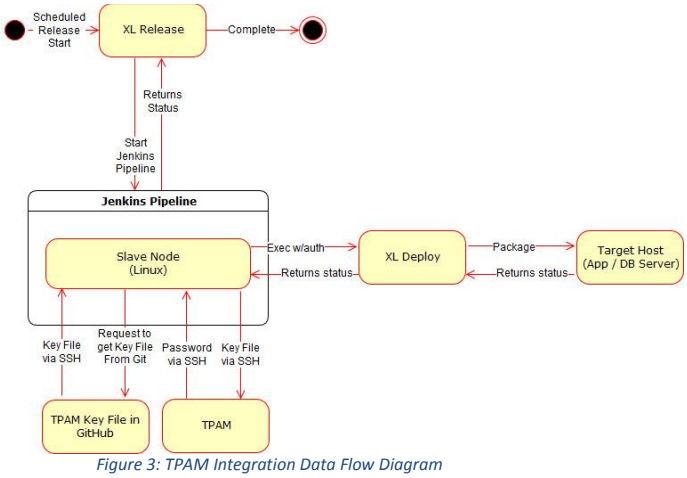


Figure 3: TPAM Integration Data Flow Diagram

[5] PAM controls are only required for Windows-based deployments, as Linux-based deployments achieve this control through the use of SSL Certificates.

6. DevOps Mode 2 Standard Release

DevOps Mode 2 Standard Releases follow a controlled path from Initiation through Production. This flow includes checkpoints and controls to validate that all compliance and security requirements are completed. ***Figure 4: DevOps Mode 2 Standard Release Flow*** below provides a visual representation of this flow, with examples for specific tasks within the stages of a Release. Note that the tools and technologies included in this diagram are provided as examples, and that the actual technical components of this flow may vary.

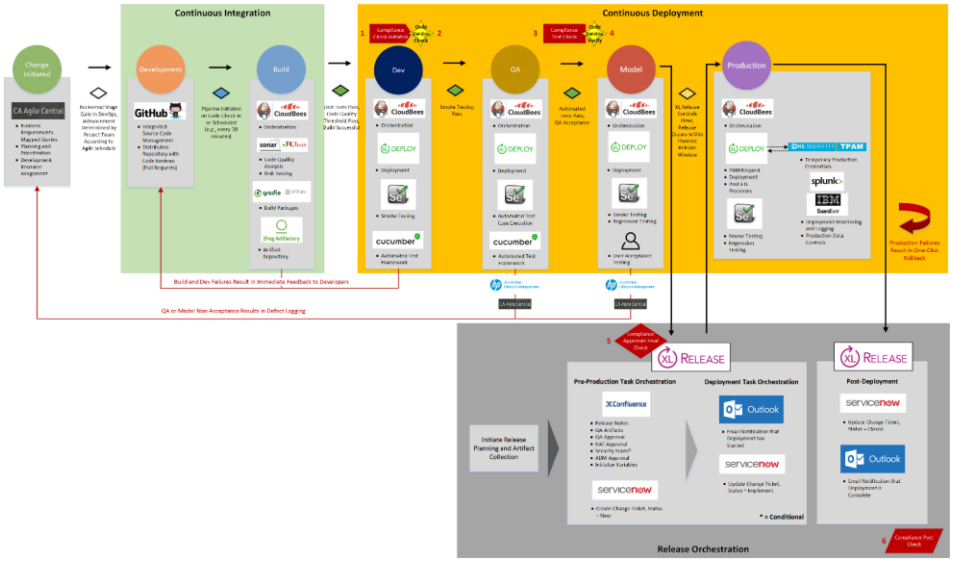


Figure 4: DevOps Mode 2 Standard Release Flow

Note: Tools/Technologies shown in ***Figure 4: DevOps Mode 2 Standard Release Flow*** above are included for representative purposes only.

The DevOps Mode 2 Standard Release Flow begins when a new Change is initiated by an application team. This includes the identification of new business requirements, mapping of those requirements to development work, planning and prioritization of that work, and assignment of resources to begin development.

The Continuous Integration phase of the release flow includes Development and Build activities. The outcome of the Continuous Integration phase is that version controlled artifacts are stored in a central artifact repository.

The Continuous Deployment phase of the release flow includes Dev, QA, Model, and Production activities. This phase includes Compliance-related activities and release orchestration activities. The outcome of the Continuous Deployment phase is a validated and approved production deployment.

DevOps Mode 2 Release Orchestration

Release Orchestration is the phase within the DevOps Mode 2 Release Process that enforces all pre-production requirements (e.g., approvals, compliance artifact creation/maintenance, and change record creation via SONIC). ***Figure 5: Release Orchestration Flow*** below provides a visual representation of the phases and tasks involved in the Release process. Note that the specific tasks and sequence are subject to change.

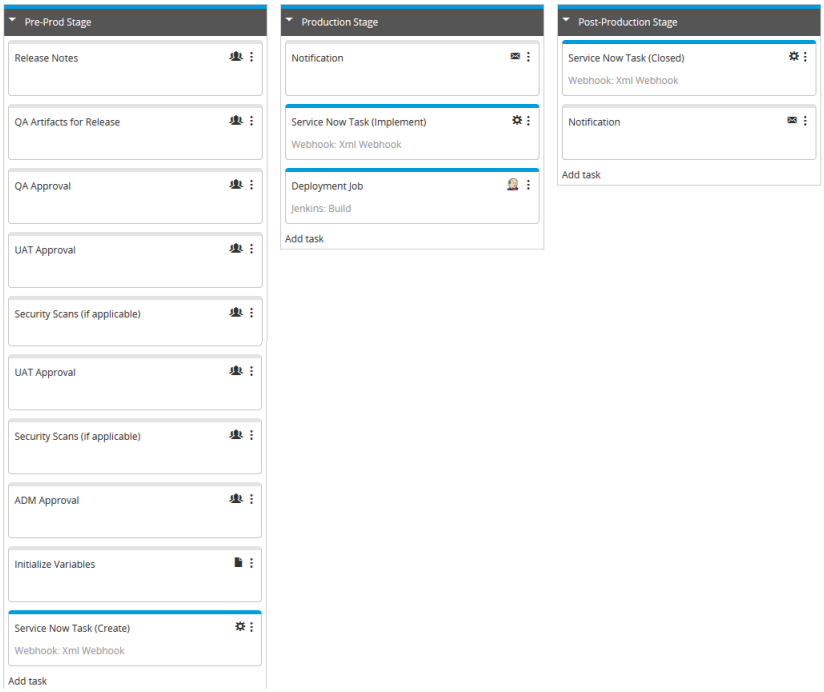


Figure 5: Release Orchestration Flow

DevOps Mode 2 Release Dictionary

Stage	Task	Task Type[6]	Description
Pre-Prod Stage	Release Notes	Manual	Upload/link the detailed release notes for the deployment. This would include a list of all code check-ins, with comments, that are included in the release.
	QA Artifacts for Release	Manual	Upload/link the QA Test artifacts including test results and defect logs
	QA Approval	Manual	Upload/link QA Approval
	UAT Approval	Manual	Upload/link UAT Approval
	Security Scans*	Manual	When required, this task ensures that Security Scans (Static, Dynamic, Penetration Testing) are completed in the Release Cycle
	ADM Approval	Manual	This includes confirmation of basic initiation artifacts/approvals. It also ensures actions are underway on longer runway activities like vulnerability reviews (when required).
	Service Now Task (Create)	API Request	This programmatically creates a new Change Record.
Production Stage	Notification	Notification	This task emails the application team about an upcoming deployment.
	Service Now Task (Implement)	API Request	This programmatically updates the status of an existing Change Record.
	Deployment Job	Pipeline Job	This calls the CD Pipeline Deployment processes to initiate the actual production deployment.
Post-Production Stage	Application Verification Process	Automated Test	This step validates that the application deployed successfully.
	Service Now Task (Closed)	API Request	This programmatically closes an existing Change Record.
	Notification	Notification	This task emails the application team about the completion of a deployment.

*Conditional Requirement – These tasks are only required if they are relevant for a specific release. It is the responsibility of the Application team to work with BF/BU compliance and determine the applicability of these tasks for their DevOps Mode 2 Standard Release.

[6] Task type is accurate as of Sep 2017. As the process and technology of Release Orchestration matures, manual tasks will be replaced with automation such as scripting and API calls.

7. DevOps Mode 2 Maintenance Requirements

As with most new capabilities, technological and process enhancements are expected within the DevOps Mode 2 Operating Model. These may result from additional enterprise integrations, new technologies, or the iterative improvement of existing capabilities. GFs and BUs that leverage a DevOps Mode 2 Standard Release process must conform to these updates and improvements as they are made available for general consumption. This will ensure that all application releases will stay current with enterprise requirements and controls.

8. DevOps Mode 2 Security and Compliance

DevOps Mode 2 Standard Releases allow for streamlining of Security and Compliance through automation and self-inspection. This process can be consistently managed across all change and project activity.

Security Scanning Requirements

Commercial DevOps Mode 2 Standard Release applications are subject to static and dynamic security scanning requirements as defined by AIG Application Security Testing. These requirements ensure that application changes do not introduce vulnerabilities and threats to AIG systems or data. Together, static and dynamic scanning produce a holistic approach to identifying application vulnerabilities such that can be remedied prior to releasing an application change to a production environment. The approved application scanning platform for static and dynamic scanning is Veracode. Continuous Delivery (CD) pipelines provide integration with the Veracode platform to ensure that applications can automatically align with AIG application security policies.

Additional detail on the specific components, workflows, and detailed requirements of Static and Dynamic Scanning for a DevOps Mode 2 Standard Release can be found in **Appendix 1 – Security Scanning Workflow and Requirements.**

Compliance Integration within DevOps Mode 2 Release Process

Dedicated Compliance tasks are embedded within the Mode 2 DevOps Release Process as indicated in ***Figure 4: DevOps Mode 2 Standard Release Flow***. These activities include:

1. Compliance Check Initiated – This includes confirmation of basic initiation artifacts/approvals. It also ensures actions are underway on longer runway activities like vulnerability reviews (when required)
2. Code Version Check – This check ensures code has been checked into a repository, and protected from additional developer modifications prior to production after tests are completed.
3. Compliance Test Check – This check aligns to final QA test check, once the stories are locked down. It also includes project manager and quality assurance manager sign-off unless the QA function is automated to occur within the pipeline.
4. Code Version Verify – This check provides traceability between what's been retained under #2 and what's deployed to PROD.
5. Compliance Approvals Final Check – This includes final business UAT and ready to implement signoffs and risk acceptance on outstanding vulnerability issues.

6. Compliance Post Check – This includes confirmation of ticket closure, a content traceability check, and the completed change closure report that confirms the business received what they expected.

DevOps Mode 2 Compliance Checklist

A single checklist, with a few conditional artifacts, will be used across all DevOps Mode 2 Standard Release deployments.

- GF and BU Compliance teams will verify artifacts and information in the application teams' tool/repository
- As tools/repositories mature, compliance teams will be kept up to date with current location information
- As automation becomes available, the compliance teams will stop manual reviews and focus on non-compliant issues found through automation

Mandatory Requirements	***DevOps Mode 2 Standard Release Artifact***
Vulnerability/Security GREEN Status	Validated Compliance/Security Status*
Compliance Inspection Template (Visualization, Low Risk, etc.)	Compliance Inspection Completed*
Change/Release Request and Delivery PM approval	Addressed by SONIC "DevOps Standard" Application Change Request Workflow
Business (OO) Request Approval	Approved/Validated Stories
IT/QA Test Results (with Defect Information) Approvals	Evidence of Testing Performed and Test Results from Release Process*
UAT Test Results with Business signoff	Business Approval indicating Fixes/Enhancements included in Release
Implementation Instructions present	Addressed by SONIC "DevOps Standard" Application Change Request Requirements
Back-out plan uploaded	Addressed by SONIC "DevOps Standard" Application Change Request Requirements
Production Implementation Approvals	Addressed by SONIC "DevOps Standard" Application Change Request Workflow
Change Request Closure Record	Addressed by SONIC "DevOps Standard" Application Change Request Requirements
Change Request Closure business post-implementation input	Post-Production Business Acceptance
Completed Project Checklist	Completed Project Checklist

Conditional Requirements	
Application Run Book	Updated when required.
Retention and Preservation Checklist for Data Migration	If required, DevOps Mode 2 Standard Release is not an allowable Release Model.
Electronic Information Disposition	If required, DevOps Mode 2 Standard Release is not an allowable Release Model.
Software Security Assessment (SSA)/MASA	If required, DevOps Mode 2 Standard Release is not an allowable Release Model.
Application Scan	If required, addressed by Dynamic Security Scanning as described in this document.
Source Code Scan	If required, addressed by Static Security Scanning as described in this document.
Penetration Testing	If required, DevOps Mode 2 Standard Release is not an allowable Release Model. Applications may conduct these scans outside of the Standard Release flow and re-enter the DevOps Mode 2 Standard Release process when complete.
Approved Risk Acceptance for security/vulnerabilities	If required, DevOps Mode 2 Standard Release is not an allowable Release Model. Applications may conduct these scans outside of the Standard Release flow and re-enter the DevOps Mode 2 Standard Release process when complete.

*Requirement may be fulfilled by manual or automated tasks, or a combination thereof.

Required Artifacts for DevOps Mode 2 Standard Release

Based on the above checklist, the artifacts required for a DevOps Mode 2 Standard Release are aggregated and listed below. The format and location of each artifact may vary, but must be accepted by the affected GF/BU Compliance group. Also note that these artifacts may be fulfilled manually, via automation, or a combination thereof.

Artifact	Responsible Party
1. SONIC DevOps Application Change Record – Enforces data requirements and appropriate workflow	Application PM
2. Approved/Validated Stories	Application PM
3. Evidence of Pre-Production Testing Performed and Test Results from Release Process	Application PM
4. Business Approval indicating Fixes/Enhancements included in Release	Application Operational Oversight (OO)
5. (Conditional) Application Run Book – If affected by Release	Application PM
6. Compliance Approval – This is the outcome of Steps #1-5 of the Compliance Integration within DevOps Mode 2 Release Process described above	GF/BU Compliance
7. Completed Project Checklist	GF/BU Compliance

After completing the Production deployment, the following additional artifacts must also be collected in order to finalize the Release process:

Artifact	Responsible Party
8. Post-Production Business Acceptance	Application Operational Oversight (OO)
9. Compliance Post Check – Aligned to Step #6 of the Compliance Integration within DevOps Mode 2 Release Process described above	GF/BU Compliance

Alignment with Corporate and Divisional Policies

Appendix 2 – Policy Alignment Mapping provides a detailed alignment of the DevOps Mode 2 Release Model to existing AIG policy requirements.

9. Roles and Responsibilities

Overall responsibilities for the SOP are defined below.

Global Functions (GFs) and Business Units (BUs)

Responsible for:

- Implementing the Standard
- Complying with the Standard
- Developing GF or BU-specific SDLC standards and procedures if GFs and BUs need to extend the baseline requirements defined within the Standard
- If leveraging a third-party vendor for application development and maintenance services:
    - Managing overall project and providing oversight
    - Reconciling roles and responsibilities between third-party vendor and GF/BU
    - Ensuring the completion of required actions and artifacts GFs and BUs will have to designate individuals or map an existing role to the designated application development and project management roles. When role designations other than the ones given in this standard are used, a mapping of the roles will be maintained.

Responsibility for Compliance

AIG GFs and BUs are responsible for adhering to and complying with this SOP. GFs and BUs may develop ***additional*** GF or BU-specific SDLC procedures. If GFs and BUs need to extend or realign the baseline requirements defined within this procedure, GFs and BUs are responsible for developing such additional SDLC procedures where required by their local leadership.

Terms and Definitions

- **DevOps** - DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.[7]

[7] https://aws.amazon.com/devops/what-is-devops/

- **Bimodal** – Bimodal IT is the practice of managing two separate, coherent modes of IT delivery, one focused on stability and the other on agility.[8]
  - **Mode 1** – Mode 1 is traditional and sequential, emphasizing safety and accuracy.
  - **Mode 2** – Mode 2 is exploratory and nonlinear, emphasizing agility and speed Mode 2 works in small self-managing teams, focused on the outcome, chooses own tools, no fixed rules, adaptive to context, and low ceremony. Even in Mode 2, AIG applications are subject to controls and stage gates that have been established to align with all AIG compliance, regulatory, security and audit policies.
- **Continuous Integration** – Continuous integration is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run. Continuous integration most often refers to the build or integration stage of the software release process and entails both an automation component (e.g. a CI or build service) and a cultural component (e.g. learning to integrate frequently). The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.[9]
- **Continuous Delivery** – Continuous delivery is a DevOps software development practice where code changes are automatically built, tested, and prepared for a release to production. It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage. When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that has passed through a standardized test process.
  With continuous delivery, every code change is built, tested, and then pushed to a non-production testing or staging environment. These tests may include UI testing, load testing, integration testing, API reliability testing, etc. This helps developers more thoroughly validate updates and pre-emptively discover issues.[10]
- **Continuous Deployment –** Continuous deployment is an extension of continuous delivery, where the push to production happens automatically without explicit approval.
- **Normal Change** – Refer to AIG Change Management Standard[11]
- **Standard Change** – Standard Changes are low risk changes that follow an established and approved process, and can be pre-approved. This includes common application enhancements such as periodic updates of reference information, website content and styling changes, and certain types of application or operating system patches that have a well-understood impact. The change proposer does not require approval before deploying the change, and change deployments can be completely automated and logged for traceability.
- **Static Scanning** - Static scanning refers to analysis that can be performed without actually executing programs. In most cases the analysis is performed on some version of the compiled application code or application binaries. Typically a static analysis tool will inspect program code for all possible run-time behaviors and seek out coding flaws, back doors, and potentially malicious code[12].

[8] http://www.gartner.com/it-glossary/bimodal
[9] https://aws.amazon.com/devops/continuous-integration/
[10] https://aws.amazon.com/devops/continuous-delivery/
[11] AIG Application Change Management Standard is defined in a separate document
[12] https://www.veracode.com/blog/2013/12/static-testing-vs-dynamic-testing

- **Dynamic Scanning** - Dynamic scanning is executed while a program is in operation. A dynamic test monitors system memory, functional behavior, response time, and overall performance of the system. This method is similar to the way a malicious third party may interact with an application[13].
- **Test-Driven Development (TDD)** – TDD is a software development process that relies on the repetition of a very short development cycle: Requirements are turned into very specific test cases, then the software is improved to pass the new tests, only.[14]
- **Behavior-Driven Development (BDD)** – BDD includes the practice of writing tests first (from TDD), but focuses on tests which describe behavior, rather than tests which test a unit of implementation. BDD is largely facilitated through the use of a simple domain-specific language (DSL) using natural language constructs (e.g., English-like sentences) that can express the behavior and the expected outcomes.[15]

Related Content

- AIG Acquisition of Goods and Services Policy
- AIG Application Change Management Policy and Standard
- AIG Application Development Policy
- AIG Application Development Standard (Agile)
- AIG Technology and Application Portfolio Management Standard
- AIG Application Quality Assurance/Testing Policy and Standard
- AIG Data Migration Standard
- AIG Enterprise Data Management Policy and Standards
- AIG Enterprise Information Model
- AIG External Website Lifecycle Management Policy and Standard
- AIG Global Business Continuity Management Policy and Standards
- AIG Global Compliance Group Protocols for Review of Proposed External and Internal Use of New Technologies
- AIG Global Economic Sanctions Compliance Policy
- AIG Global Information Handling Policy
- AIG Innovation Gateway Process
- AIG IT Security and IT Risk Policies and Standards
- AIG Project Management Policy
- AIG Project Management Lifecycle Standard
- AIG Records and Information Management Policy and Records Retention Schedules
- AIG Software and Hardware Product Standards
- AIG Standards for the Disposal of Structured Data
- AIG Vendor Risk Management & Governance Policy and Standards

[13] https://www.veracode.com/blog/2013/12/static-testing-vs-dynamic-testing
[14] https://en.wikipedia.org/wiki/Test-driven_development
[15] https://en.wikipedia.org/wiki/Behavior-driven_development

Appendix 1 – Security Scanning Workflow and Requirements

Static Scanning Workflow

Three distinct capabilities exist for static scanning within the Veracode platform. These are:

4. Veracode Greenlight Integrated Development Environment (IDE) plugin
5. Veracode Developer Sandbox
6. Veracode Policy Static Analysis

Figures 6-8 below provide an overview of where and how static scanning is performed for a DevOps-enabled applications.

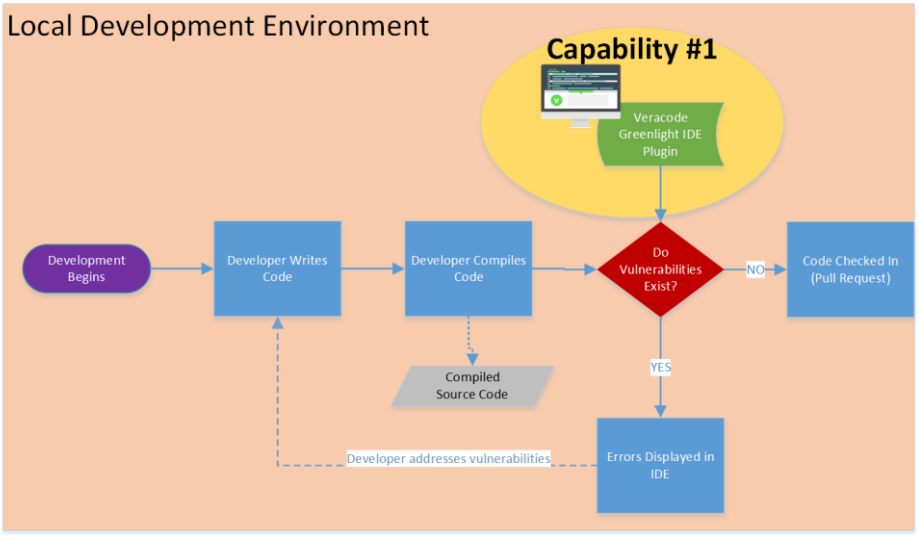


Figure 6: Veracode Greenlight IDE Plugin

Veracode Greenlight finds security defects in application code and provides contextual remediation advice to help developers fix issues in seconds, right in their IDE. This effectively pulls security scanning as far "left" as possible, making information about potential vulnerabilities available as early as possible in the development cycle. Developers do not need to provision any servers or tune the engine to use Greenlight. It simply scans in the background and provides accurate and actionable results, without taking up resources on a developer's local machine[16].

[16] https://www.veracode.com/products/greenlight

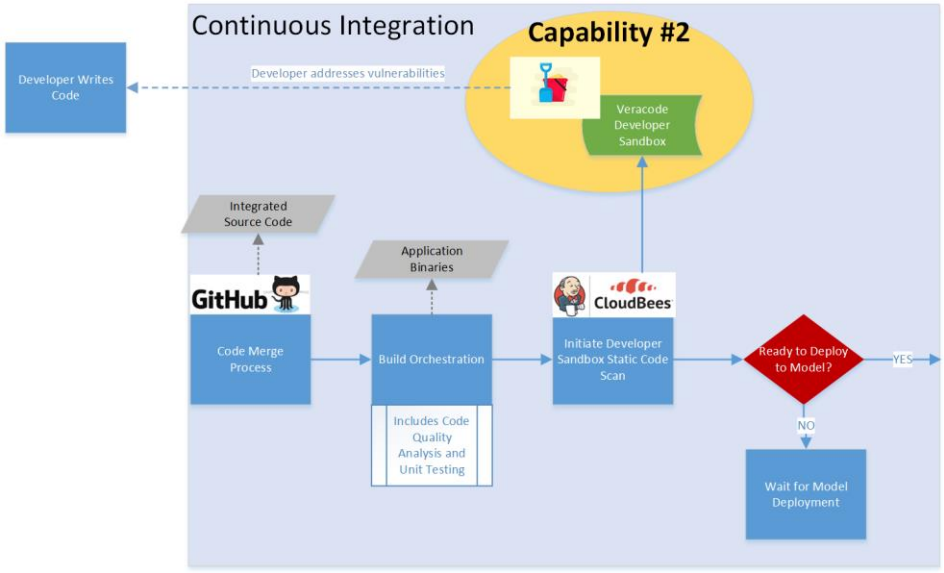


Figure 7: Veracode Developer Sandbox

During the Continuous Integration (CI) phase of the DevOps Mode 2 Standard Release Process, the Veracode Developer Sandbox is called to initiate static scanning. This capability leverages a static analysis scanning platform that assesses the security of micro services, web, mobile and desktop applications. Results from the Developer Sandbox are not formally reported, but are made available to developers to inform vulnerability remediation early in the release cycle. No action is required when security vulnerability is found at this stage, however vulnerabilities that are still present and that exceed the tolerable security risks as defined by security policy (see *Static Scanning Requirements*) must be addressed prior to a production deployment.

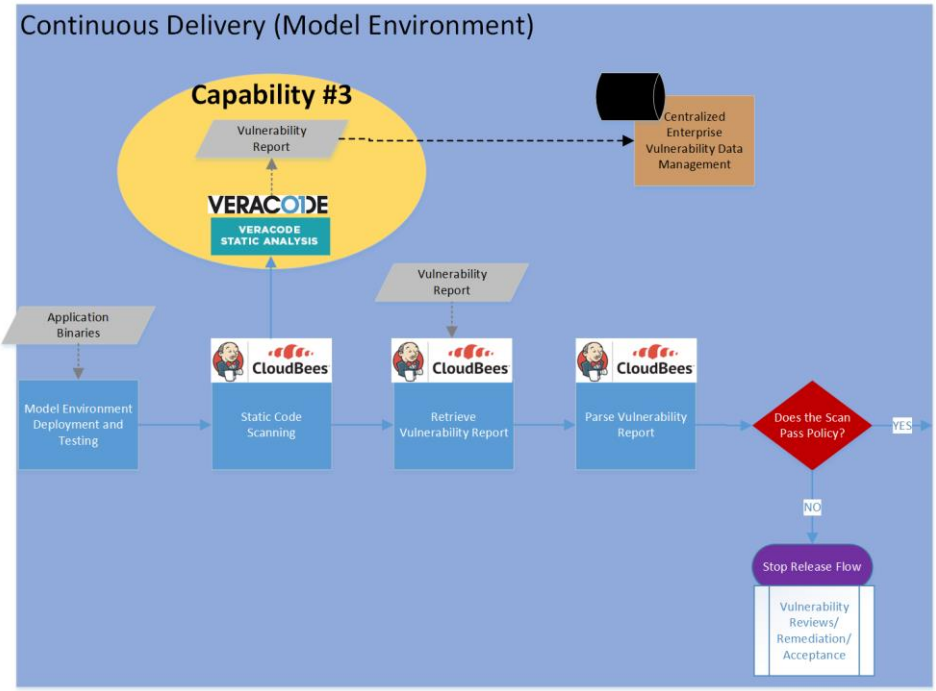


Figure 8: Veracode policy Static Analysis

During the Continuous Delivery (CD) phase of the DevOps Mode 2 Standard Release Process, Veracode policy Static Analysis is called to initiate static scanning. This scan is executed in parallel with the Model environment deployment, which acts as a pre-production environment and is a high probability release candidate for production. Note that this is the same underlying capability as the Developer Sandbox scan and that the exact same artifacts (e.g., application binaries) are scanned, so we expect to see the same results in terms of identified vulnerabilities. The difference at this step is that the output of the scan is reported and formally logged and tracked for the application.

The DevOps pipeline will automatically stop the release flow if vulnerabilities that are found exceed the risk thresholds set by security policy (see *Static Scanning Requirements*).

Static Scanning Requirements

The following static scanning requirements apply to DevOps-enabled applications:

3. Formal Static scanning is required prior to any Production release
4. DevOps Pipelines must stop the automated release flow if one or more Very High or High vulnerabilities is found during Policy Static Analysis
    a. Optionally, application teams may choose to prevent automated releases when a Medium vulnerability is present.

Commented [WC1]: Will need to discuss this with the risk team to finalize

Any vulnerabilities that stop the release flow (per requirement #2) will initiate a manual review, remediation, and/or acceptance process that occurs outside of the DevOps Mode 2 Standard Release Process. If vulnerabilities are remediated, the process will begin again with the process as represented in **Figure 6: Veracode Greenlight IDE Plugin.**

Dynamic Scanning Workflow

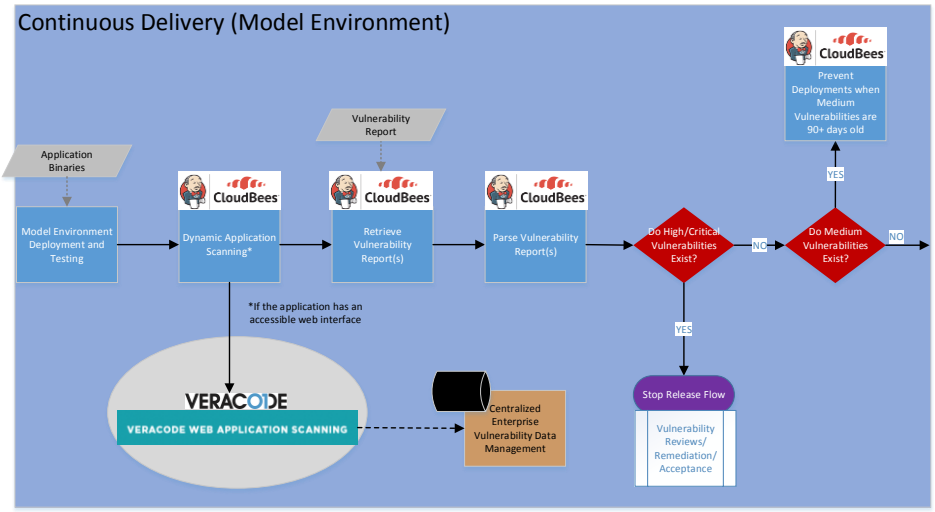


Figure 9: Dynamic Scanning Workflow

Applications with a web interface or a web service that can be invoked over the Internet are subject to dynamic scanning requirements. During the Continuous Delivery (CD) phase of the DevOps Mode 2 Standard Release Process, Veracode Web Application Scanning is leveraged for dynamic scanning. This scan is executed against the Model environment deployment, which acts as a pre-production environment and is a high probability release candidate for production.
The DevOps pipeline will automatically stop the release flow if vulnerabilities that are found exceed the risk thresholds set by security policy (see *Dynamic Scanning Requirements*).

Dynamic Scanning Requirements

The following dynamic scanning requirements apply to DevOps-enabled applications with a web interface or a web service that can be invoked over the Internet:

3. Dynamic scanning is required for a production release.
4. DevOps Pipelines must stop the automated release flow if one or more Very High or High vulnerabilities is found during Dynamic Scanning.
    a. Optionally, application teams may choose to prevent automated releases when a Medium vulnerability is present

Veracode Scan Results Review Process

Application teams must work with the SCSA team (SCSA-SourceCodeSecurityAssessmentTeam@aig.com) if results from security scans are believed to include false positives.

Additional Detail on Security Scanning Onboarding

Onboarding to automated application static and dynamic security scanning capabilities (as described further in *Security Scanning Requirements*) will follow this process:

7. Commercial DevOps Engineering Team requests a new application for Veracode scanning through the SCSA team. This should be an application with active new or maintenance development.
    - As part of the request, DevOps Engineering provides application details (e.g., GEAR ID and Application Name) and a list of the application team developers (names and emails).
8. SCSA Team creates accounts for the developers within the Veracode platform and associates them with the application. Developers are emailed their credentials.
9. Application developers log into Veracode and generate API key credentials. Then developers install Veracode Greenlight IDE by following download instructions that are provided by the SCSA team. Application developers copy the newly created API key credentials into Greenlight. Note: API key credentials are time sensitive.
10. Application developers review Veracode training materials provided by SCSA team.
11. SCSA team enables Jenkins jobs to include scanning of the new application. Additional configuration work may be required for dynamic scanning to be properly configured for applications.
12. Commercial DevOps Engineering Team will work with application stakeholders to validate that users have access and that the scanning capabilities are working as expected.

Appendix 2 – Policy Alignment Mapping

Policy Source	Requirement Category	Applicable Requirement	Met?	Mechanism(s) (Example)	Notes/Comments
AIG Information Security Policy	3.2 Information Security Framework	Quality assurance procedures must be defined and managed.	Yes	- Jenkins	The DevOps Pipeline (e.g., Jenkins) provides a structured and consistent application to validate all release requirements, including quality assurance procedures. The Pipeline itself is version controlled and managed by the GF or BU DevOps Teams.
AIG Information Security Policy	3.4 Secure Management of Assets	Information assets must be managed based on the classification level of the asset as defined in the AIG Global Information Handling Standard.	Yes	- Artifactory	All application assets (e.g., binaries, executables) are maintained in an approved repository (e.g., Artifactory). All application assets can only be published via the standard pipeline workflow; no individual resources/users have permissions or authorization to publish artifacts. DevOps pipelines do not handle actual application data, so pre-existing controls within the application design will address those requirements.
AIG Global Information Technology Governance and Strategy Policy	3.2.1 Technology Management	All technology used or proposed for use must be assigned a lifecycle stage and maintained according to that assignment.	Yes	- Pipeline Toolchain	All applicable DevOps Toolchain components must obtain SSA sign-off and are approved for use as live production services. Technology Services manages the Pipeline Toolchain and aligns their lifecycle management stages with the current policy.
AIG Global Information Technology Governance and Strategy Policy	3.2.1 Technology Management	All technologies in the Technology Portfolio must have defined and managed lifecycles.	Yes	- Pipeline Toolchain	All applicable DevOps Toolchain components must obtain SSA sign-off and are approved for use as live production services. Technology Services manages the Pipeline Toolchain and aligns their lifecycle management stages with the current policy.
AIG Global Information Technology Governance and Strategy Policy	3.2.4 IT Release Management	A process that controls the life cycle management (i.e., planning, scheduling and deployment) of an IT system or service must be established, managed, measured and maintained.	Yes	- Pipeline Toolchain	All applicable DevOps Toolchain components follow standard enterprise life cycle management processes.
AIG Global Information Technology Governance and Strategy Policy	3.2.4 IT Release Management	The IT Release Management Process must plan, schedule, and control the build, test and deployment of IT releases within agreed IT service levels.	Yes	- XL Release	Releases are orchestrated (e.g., by the XL Release platform) in such a way that the releases adhere to all agreed IT service levels for DevOps applications

Policy Source	Requirement Category	Applicable Requirement	Met?	Mechanism(s) (Example)	Notes/Comments
IT Change Management Standard	3.2 IT Change Management Framework	Lead times for all IT change types, including emergency and expedited IT changes, must be defined to allow time for proper approvals.	Yes	- XL Release	When required, proper approvals are captured (e.g., within the XL Release flow). These approvals are not time-limited, so there is no scenario where an approval would not occur simply because there was not enough time. The Mode 2 standard release workflow can pause indefinitely until required approvals are uploaded and/or evidenced.
IT Change Management Standard	3.3 IT Change Request	IT change tickets must be logged and stored in a controlled repository throughout IT change lifecycle.	Yes	- ServiceNow	All DevOps releases automatically create a SONIC Change ticket as defined and managed by the Change Management team.
IT Change Management Standard	3.3 IT Change Request	IT change tickets must capture all appropriate data and references including IT change description, business justification, impact analysis, and associated risks.	Yes	- ServiceNow	All DevOps releases automatically create a SONIC Change ticket as defined and managed by the Change Management team. All SONIC Change tickets contain the required content.
IT Change Management Standard	3.3 IT Change Request	IT changes must be categorized based on an established categorization schema.	Yes	- ServiceNow	DevOps Standard Change SONIC Tickets are "pre-sorted" according to AIG Change Management standards. High risk changes will not follow the Standard Change flow as defined. Only certified and approved applications will adopt the Mode 2 Standard Release process as defined by Compliance.
IT Change Management Standard	3.3 IT Change Request	Appropriate IT change tickets must be submitted within the timeframe outlines in the IT change management processes to allow for review by the IT change approval board or emergency IT change approval board.	N/A		DevOps releases do not require pre-emptive reviews by IT change approval boards. DevOps changes are still subject to any post-deployment reviews as deemed necessary by IT change approval boards.
IT Change Management Standard	3.3 IT Change Request	Design, build, and test for IT changes must be coordinated with technical teams (external to IT change management).	Yes	- Jenkins	DevOps Pipeline (e.g., Jenkins) provides framework for all design, build, and testing activities to be executed in a consistent and measurable process.
IT Change Management Standard	3.3 IT Change Request	IT changes must be tested according to IT change management processes and procedures.	Yes	- Sonar - Selenium - Cucumber	DevOps Pipeline Testing platforms (e.g., Sonar and Selenium) provide automated and manual test definition and execution capabilities that are coupled with the software deployment processes.

Policy Source	Requirement Category	Applicable Requirement	Met?	Mechanism(s) (Example)	Notes/Comments
IT Change Management Standard	3.3 IT Change Request	IT change documentation must include back out procedures.	Yes	- ServiceNow	DevOps Standard Change SONIC Tickets include references to automated back out procedures. Execution of this capability is simplified to a "one-click" rollback capability due to the version history and environmental control enabled within a Continuous Delivery deployment model.
IT Change Management Standard	3.4.2 IT Change Scheduling	Staging for IT changes must be coordinated for release management.	Yes	- XL Release	Releases are orchestrated (e.g., by the XL Release platform) to include release staging for appropriate Change Windows as defined by the application team based on resources, environmental and usage factors, and any external dependencies.
IT Change Management Standard	3.6 IT Change Monitoring/Tracking	Schedule of IT changes must be centrally maintained.	Yes	- XL Release	Releases are orchestrated (e.g., by the XL Release platform), via a central control point for all DevOps projects. Application teams cannot release outside of the XL Release flow.
IT Change Management Standard	3.6 IT Change Monitoring/Tracking	IT change request status must be updated throughout the lifecycle of an IT change.	Yes	- Jenkins - ServiceNow	DevOps Standard Change SONIC Tickets are created via the DevOps Pipeline (e.g., Jenkins), and their status is updated to "Implementation" programmatically with an additional API call.
IT Change Management Standard	3.6 IT Change Monitoring/Tracking	Any in-flight IT change request which did not meet the agreed IT change schedule must follow escalation procedures.	N/A		DevOps Changes will not occur outside of the approved standard flow. If a failure/incomplete step/missing approval in the DevOps flow caused a change to not meet the agreed change schedule, then that change will be closed and a new change will be created for a future change schedule.
IT Change Management Standard	3.6 IT Change Monitoring/Tracking	Open IT changes must be monitored to ensure IT changes are closed in a timely fashion.	Yes	- Jenkins - XL Release	All DevOps changes are monitored closely with the DevOps Pipeline (e.g., Jenkins) and release orchestration platform (e.g., XL Release). Any change that is not moving forward in a timely fashion will have high visibility and will be addressed.
IT Change Management Standard	3.7 IT Change Closure	The IT change management systems, tools and/or configuration management system must be updated to reflect IT changes.	Yes	- Jenkins - ServiceNow	DevOps Standard Change SONIC Tickets are created via the DevOps Pipeline (e.g., Jenkins), and their status is updated to "Closed" programmatically with an additional API call.
IT Change Management Standard	3.7 IT Change Closure	IT change requests must be closed with a closure code as defined in IT change management guidelines and procedures.	Yes	- ServiceNow	DevOps Standard Change SONIC Tickets are created via the DevOps Pipeline (e.g., Jenkins), and their status is updated to "Closed" programmatically with an additional API call.

Policy Source	Requirement Category	Applicable Requirement	Met?	Mechanism(s) (Example)	Notes/Comments
Configuration Management Standard	3.2 Configuration Management Framework	Configuration management procedures must be documented and maintained throughout the lifecycle of the process.	Yes	- Manual - Jenkins (future)	DevOps Pipeline (e.g., Jenkins) provides framework for programmatically executing configuration management procedures, to include such activities as automatically registering new server Cis within a CMDB, and associating Cis with a business purpose.
Configuration Management Standard	3.2 Configuration Management Framework	Relationships and linkages between Configuration Items (CI's) must be established (e.g., applications, server, database and owner relationships) as part of deployment to production and maintained through the life cycle of the CI to facilitate impact assessment.	Yes	- Manual - Jenkins (future)	DevOps Pipeline (e.g., Jenkins) provides framework for programmatically executing configuration management procedures, to include such activities as automatically registering new server Cis within a CMDB, and associating Cis with a business purpose.
Configuration Management Standard	3.2 Configuration Management Framework	Configuration management measures must be maintained.	Yes	- Manual - Jenkins (future)	DevOps Pipeline (e.g., Jenkins) provides framework for programmatically executing configuration management procedures, to include such activities as automatically registering new server Cis within a CMDB, and associating Cis with a business purpose.
Configuration Management Standard	3.2 Configuration Management Framework	Quality assurance procedures must be defined and managed.	Yes	- Manual - Jenkins (future)	DevOps Pipeline (e.g., Jenkins) provides framework for programmatically executing configuration management procedures, to include such activities as automatically registering new server Cis within a CMDB, and associating Cis with a business purpose.
Configuration Management Standard	3.3.3 Configuration Control	Configuration changes must be authorized in accordance with IT change management processes.	Yes	- Jenkins	All changes in the DevOps model, whether application code or configuration related are treated the same. The DevOps Pipeline (e.g., Jenkins) ensures that configuration control processes and their associated

Policy Source	Requirement Category	Applicable Requirement	Met?	Mechanism(s) (Example)	Notes/Comments
					requirements are maintained for all environments.
Configuration Management Standard	3.3.3 Configuration Control	Configuration change processes must comply with IT change management processes.	Yes	- Jenkins	All changes in the DevOps model, whether application code or configuration related are treated the same. The DevOps Pipeline (e.g., Jenkins) ensures that configuration control processes and their associated requirements are maintained for all environments.
Configuration Management Standard	3.3.3 Configuration Control	Configurations must be maintained in a secure location.	Yes	- Artifactory	All changes in the DevOps model, whether application code or configuration related are treated the same. The artifact repository (e.g., Artifactory) provides a secure, version controlled location for all configuration files to be stored.
IT Release Management Standard	3.2 Release Management Framework	Release management scope must be established.	Yes	- XL Release - ServiceNow	Releases are orchestrated by the XL Release platform, which serves as a central control point for all DevOps projects. All release management activities are performed within this platform. ServiceNow serves as the AIG system of record for releases.
IT Release Management Standard	3.2 Release Management Framework	Release management systems and tools must be identified and maintained.	Yes	- ServiceNow - Jenkins	All release management procedures are maintained by ServiceNow. DevOps releases do not allow for variances/escalations outside of the approved flow. This is controlled via the checkpoints and stage gates of the DevOps Pipeline (e.g., Jenkins).
IT Release Management Standard	3.2 Release Management Framework	Release management procedures, including escalation points must be maintained throughout the lifecycle of the process.	Yes	- ServiceNow - Jenkins	All release management procedures are maintained by ServiceNow. DevOps releases do not allow for variances/escalations outside of the approved flow. This is controlled via the checkpoints and stage gates of the DevOps Pipeline (e.g., Jenkins).
IT Release Management Standard	3.2 Release Management Framework	Release management measurements must be maintained.	Yes	- ServiceNow - Jenkins	All release management measurements are maintained by ServiceNow. DevOps releases do not allow for variances/escalations outside of the approved flow. This is controlled via the checkpoints and stage gates of the DevOps Pipeline (e.g., Jenkins).
IT Release Management Standard	3.2 Release Management Framework	Quality assurance procedures must be defined and managed.	Yes	- ServiceNow	All release management procedures are maintained by ServiceNow, including the maintenance of quality assurance procedures.

Policy Source	Requirement Category	Applicable Requirement	Met?	Mechanism(s) (Example)	Notes/Comments
IT Release Management Standard	3.4 Release Planning	Releases must be tracked in a central system integrated with the version control system(s).	Yes	- Jenkins - Enterprise GitHub	The DevOps Pipeline (e.g., Jenkins) controls all releases and is directly integrated with the DevOps Version Control system (e.g., Enterprise GitHub)
IT Release Management Standard	3.4 Release Planning	Release packages must be defined and only authorized changes can be assigned to release packages.	Yes	- Jenkins - Enterprise GitHub	The DevOps Pipeline (e.g., Jenkins) and the associated process ensures that only authorized changes can become release packages. No alternate flows are enabled.
IT Release Management Standard	3.4 Release Planning	Release and back-out (roll back) plans must be defined, reviewed for quality assurance and agreed upon with stakeholders prior to promoting changes into production.	Yes	- ServiceNow - Jenkins	DevOps Change tickets include roll-back instructions, and the procedure to execute this is validated with stakeholders during DevOps onboarding activities.
IT Release Management Standard	3.5 Release Build	An approach for building, testing and maintaining controlled quality assurance (QA) and production environments must be established.	Yes	- Jenkins - XL Deploy	All environments (not just production) are deployed and validated with the XL Deploy platform following a controlled, repeatable process that is fully reportable and completely version controlled.
IT Release Management Standard	3.5 Release Build	All build and test activities must be communicated in advance and appropriate resources must be assigned.	Yes	- DevOps Pipeline	Build and Test activities are automated to the furthest extent possible from the DevOps Pipeline (e.g., Jenkins). This includes build automation (e.g., XL Deploy), and test automation (e.g., Sonar and Selenium). When required, manual actions are orchestrated (e.g., via XL Release).
IT Release Management Standard	3.5 Release Build	Release components must be assembled, integrated, and unit-tested to confirm that they meet business requirements without fault.	Yes	- DevOps Pipeline	DevOps Pipeline (e.g., Jenkins) validates that all release components have been tested and meet acceptance thresholds as defined by business.
IT Release Management Standard	3.5 Release Build	Release documentation must be updated.	Yes	- XL Release	DevOps release orchestration (e.g., XL Release) documents all release activities.
IT Release Management Standard	3.5 Release Build	All releases must include test plan(s) or acceptance criteria.	Yes	- XL Release - Jenkins	DevOps release orchestration (e.g., XL Release) and DevOps Pipeline (e.g., Jenkins) provide links to test plans and acceptance criteria.
IT Release Management Standard	3.6 Release Validation	Non-emergency releases (planned releases) and release components must be tested and approved prior to promotion to production environments. Emergency releases must also be tested and approved prior to	Yes	- XL Release - Jenkins	DevOps release orchestration (e.g., XL Release) and DevOps Pipeline (e.g., Jenkins) validate all testing and require approval prior to production.

Policy Source	Requirement Category	Applicable Requirement	Met?	Mechanism(s) (Example)	Notes/Comments
		promotion to production environments.			
IT Release Management Standard	3.6 Release Validation	For non-emergency releases, user acceptance testing (UAT) must be performed prior to promotion to production. For emergency releases, UAT must also be performed before promotion to production.	Yes	- XL Release - Jenkins	DevOps release orchestration (e.g., XL Release) and DevOps Pipeline (e.g., Jenkins) validate all testing and require approval prior to production, to include UAT phases.
IT Release Management Standard	3.7 Release Turnover	Release packages must only include authorized changes by a change authority.	Yes	- XL Release - Jenkins	DevOps release orchestration (e.g., XL Release) and DevOps Pipeline (e.g., Jenkins) ensure that release packages follow the approved path, and that no non-authorized changes are included.
IT Release Management Standard	3.7 Release Turnover	An approval must be received from appropriate business owners or key stakeholders for promoting the releases into production.	Yes	- XL Release	DevOps release orchestration (e.g., XL Release) includes a task to obtain business approval prior to production release.
IT Release Management Standard	3.7 Release Turnover	Release packages must be tested and verified before placing in the definitive media library (DML).	Yes	- XL Deploy - Jenkins - Artifactory	Release packages built (e.g., from XL Deploy) are tested from the DevOps Pipeline (e.g., Jenkins). This occurs prior to deployment. The DML for AIG is an approved artifact repository (e.g., Artifactory).
IT Release Management Standard	3.7 Release Turnover	Access to promote changes into production must be segregated from development personnel and restricted to authorized users.	Yes	- XL Deploy - XL Release - Jenkins	DevOps releases are managed by the DevOps teams, who has access to the key Toolchain components that enable and orchestrate the production promotion process. Application developers do not have access to or control over these resources.
IT Release Management Standard	3.7 Release Turnover	Releases to production environments must use the same versions that were tested in QA environments.	Yes	- Artifactory	All deployments to environments use the exact same binaries from the artifact repository (e.g., Artifactory).
IT Release Management Standard	3.7 Release Turnover	Release must be deployed from the DML to the production environments by following agreed release plan and schedule.	Yes	- XL Release	DevOps release orchestration (e.g., XL Release) ensures that all releases, including DML changes, follow agreed release plan and schedule.

Policy Source	Requirement Category	Applicable Requirement	Met?	Mechanism(s) (Example)	Notes/Comments
Source Code Protection Standard	3.3 Source Code Protection Requirements	Security reviews must be conducted in accordance with source code protection processes on software source code based on application sensitivity and exposure to threats (e.g., internet facing applications). The source code security review techniques used must include, but not be limited to, static and/or dynamic analysis.	Yes	- Manual - Jenkins (future) - Veracode (future)	All DevOps changes are subject to security scanning as required by security policy. This includes static and dynamic scanning.
SDLC Agile Standard Operating Procedure	Baseline Software Development Lifecycle Requirements for Agile Development Methodology	AGILE projects must be developed in line with PMLC requirements. Project Management must show adherence to artifact and approval retention throughout the project development into release to production and to decommission.	Yes	SDLC DevOps SOP	SDLC DevOps SOP Document Provides the Artifacts and Compliance Alignment for Mode 2 Standard Releases.
SDLC Waterfall Standard Operating Procedure	Baseline Software Development Lifecycle Requirements for Waterfall Development Methodology	Waterfall projects must be developed in line with PMLC requirements. Project Management must show adherence to artifact and approval retention throughout the project development into release to production and to decommission.	Yes	SDLC DevOps SOP	SDLC DevOps SOP Document Provides the Artifacts and Compliance Alignment for Mode 2 Standard Releases.

