

OPEN SOURCE LANGUAGE AUTOMATION

BLUEPRINT for SUCCESS

ActiveState®

Decrease costs and risks of managing open source languages.
Implement a systematic and automated workflow.

01 DEFINE POLICIES

Programmatically enforce policies to ensure compliance across entire organization.

- Form Usage Criteria
- Set Package Controls
- Create License & Security Criteria
- Document Policies

02 CENTRALIZE DEPENDENCIES

Track languages and packages across DevOps to assess open source usage.

- Analyse Usage
- Define Terminology
- List Package Criteria
- Record Audit Results

03 AUTOMATE BUILDS

Reduce vulnerabilities & increase application quality; automatically create builds with a repeatable process organization-wide.

- Set Consumption Pace
- Define Build Certification
- Define Resolution Process
- Define Workflow

04 DEPLOY & MANAGE ARTIFACTS

Automatically update all of your test, stage & production servers with the appropriate and latest open source language builds.

- Set Deployment Mechanism
- Report Runtime Performance
- Update Based on Policy
- Optimize Build, Certify, Resolve

GET THE SERIES

01 Define Open Source Language Organization-Wide Policies

Programmatically enforce policies to ensure corporate policy compliance. Define what type of code usage is acceptable at each stage of your software development process & how it can be used by stakeholders in the company. Set organization-wide open source language policies, version controls and triggers correspondingly. Achieve this through the following activities:

Form Usage Criteria

List open source languages, versions and packages that can be used, changed and updated organization wide. Include trusted sources & usage restrictions based on environment or application.

Set Package Controls

Record vetted packages by languages, detail the test suite required for acceptable builds, record the acceptable modifications and sources for packages.

Create License & Security Criteria

Agree on license use by open source languages and environments (e.g. research vs production) as well as threat levels and those that require immediate action.

Document Policies

Include roles & privileges, expiration of information archives, and who has access to "secrets". Secrets are keys, certificates, and passwords required to access shared resources in a secure manner.



02 Centralize Open Source Language Dependencies

Track languages and packages across DevOps to provide understanding of open source language usage. Obtain a single source of truth for open source language and build information including packages, version, security and licenses deployed across environments. Achieve this through the following activities:

Analyse Usage

Assess open source language usage across your organization from build, to deploy, to monitoring and updates. Understand corporate usage to determine improvements.

Define Terminology

Set naming conventions for open source components, including versioning, and how each stage in the software development process will be termed organization-wide along with trigger-events to move from one stage to another.

List Package Criteria

Determine how open source code is vetted and approved for its addition to the organization's open source language repo. Include approval levels by environment, stage and use. e.g. what's validated versus experimental.

Record Audit Results

Include inventory listing all open source language builds running along with package, security & license information. Aggregate all open source language information in one place.



03 Automate Open Source Language Builds

Reduce vulnerabilities and increase application quality. Automatically create builds across your organization with a systematic, repeatable build process. Establish build, certify and resolve as the three lifecycle stages for your open source languages. Achieve this through the following activities:

Set Consumption Pace

Decide how, when and who can update open source language builds. Determine triggers that force a build update. e.g. time, major / minor language build, etc.

Define Build Certification

Establish workflow for vetting and pushing open source language builds to production. Set corporate definition for certification of open source language builds. Include certification levels as per environment (eg dev vs production), application or customer use case.

Define Resolution Process

Implement organization-wide process for how open source language builds are resolved for dependencies, licenses and security.

Set Tracking Parameters

Establish organization-wide open source language tracking parameters. Minimum set should include consumption pace, certification and resolutions against policy and results: frequency, success, fails.



04 Deploy & Manage Open Source Language Artifacts

Automatically update all of your organization's test, stage and production servers with the appropriate and latest open source language builds. Inform corporate policies with code-specific open source knowledge that considers security and performance updates. Achieve this through the following activities:

Set Deployment Mechanism

Set the mechanism by which you deploy code and define how open source language build deployment is enabled: who accesses the builds; who distributes, who pushes to production.

Report Runtime Performance

Monitor open source language builds and report based on open source language policies. Implement alerts and notifications based on **Phase 2: Centralize Open Source Language Dependencies**. Share insights as to what's in language packages and where language packages are deployed.

Update Based on Policy

Ensure continuous resolution & updates of open source language builds are realized according to policies and what's been reported. e.g. as updates and vulnerabilities occur. Automatically update test, stage & production servers with the latest open source language builds.

Optimize Build, Certify, Resolve

Run continuous build-certify-resolve cycles to improve corporate Open Source Language Automation policies. Use code-specific knowledge to improve management, trust and performance of applications.



Are You Ready to Implement Open Source Language Automation?

Have all relevant stakeholders agreed that the organization will have a single source of data for all open source binaries?



Have you defined the steps within each stage of your software development process and what is required to move from one stage to another?



Have you set the guidelines for the use, verification and versioning of code at every step of your SLDC through to continuous delivery?



Schedule a call to assess readiness and how we can help. Go to: www.activestate.com/OSLA-readiness



**Schedule a
consultation**