Open Source Compliance: Reworking Internal Processes

Aida Rivas, Senior Program Manager
Meng Chow, PhD, PMP, Staff Program Manager
August 2018
An Introduction

20 year-old enterprise software company
Over 100 products across 10 product lines
More than 6,000 developers in six development centers:
  • Sofia, Bulgaria
  • Bangalore & Pune, India
  • Beijing, China
  • Bellevue, WA
  • Palo Alto, Ca (HQ)
Open Source Compliance Framework

**Shifting Mindset & Sustaining Cultural Shift**

- Leadership Support
- Select Early Adopters
- Broaden Engagement
- Assess Compliance Culture
Open Source is Everywhere

Average codebase that is open source:

- 57% in 2018
- 36% in 2017

Applications now contain more open source than proprietary code.

Source: Synopsys 2018 Open Source Security and Risk Analysis Report
Product Team Challenges
Driving Change

Maintain accurate inventory of open source usage
Reduce late discovery of issues in a release
Ensure ongoing license compliance
Reduce rework
Leadership Support
Shaping Open Source Compliance Culture

Senior Management Role
• Drive business imperative

Open Source Program Office
• Increase productivity
• Drive efficiency and best practices
• Enable faster time to market

Middle Management Role
• Coach teams
Early Adopters
Selecting and Supporting

Selecting Early Adopters

Criteria: Assess product teams’ commitment to automating their development processes & implementing best practices

Supporting Early Adopters

Demonstrate empathy
- See from product team’s perspective

Showcase progress
- Validate small units of change with product team

Roll sleeves up
- Identify the space that no one is owning

Transparency
- Objectives, priorities, assumptions
Capture Failures

“Failure isn’t fatal, but failure to change might be” (John Wooden)

Validating all assumptions
- Assumption: availability of given infrastructure
- Problem: requirement was not communicated

Considering broader, long-term perspective
- Product Scope: API requirements lacked representation of the various build environments
- Problem: API could not be integrated in all development build environments

Results: created rework, impacted schedule
Assess Open Source Compliance Culture
Plan-Do-Check-Act (PDCA) Model for Continuous Improvement

Plan-Do-Check-Act (PDCA) Model for Continuous Improvement

W. Edwards Deming, author of PDCA model
Open Source Compliance Framework

Shifting Mindset & Sustaining Cultural Shift

- Leadership Support
- Select Early Adopters
- Broaden Engagement
- Assess Compliance Culture
Leverage early adopters

Fit / gap analysis
- Address pain points
- Align with desired business outcomes of product team

Gaps
- New requirements to bridge gap
- Progress => future engagement opportunities
Each product team is different

Business priorities
- Product lifecycle, market situation
- DevOps transformation
- Acceptable risks

Team culture
- Product history
- Organic home grown vs Mergers & Acquisition
Challenges to change

People
- Functionality vs process; different business priority
- Compliance and Security training
- Limited resources, SMEs

Process
- Development methodologies, DevOps journey
- Product planning and risks mitigation; start early
- Change initiative; assessment and prioritization

Technology
- Different languages, different build tools
- Data reporting; false positives, false negatives
- Automation; early detection, consistency
Address challenges via holistic approach

Empower teams to innovate
Enhance cross-team collaboration
Improve engagement opportunities with product teams

DevOps Ecosystem

Access points to engage each team, integrate with product release pipeline
Summary

Approach
Start small
Engage stakeholders at all times
Experiment: plan, do, check, act

Be cognizant
Product team priorities
Business risks, risks appetite
Product team culture
I am always doing what I cannot do yet, in order to learn how to do it.

Vincent van Gogh
Thank you