Developing an Open Source Governance and Compliance Program at Bank of America

Tales from the Trenches

Bank of America

Agenda

- The Open Source Business Case
- Creating an Open Source Policy
- Open Source Management
- The Free and Open Source Software (FOSS) Management Program
- Getting Help
- Risk Acceptance Process
- Remediation
- Open Source Software (OSS) Distribution
- OSS Contributions
- Looking Forward

The Open Source Business Case

- Most, if not all, organizations use OSS
- To responsibly use OSS, an organization must actively manage OSS with formal (e.g. via a policy) or informal processes, toolsets and human resources
  - Does NOT need to be a heavy process – depends on the type of OSS usage and resulting risk vs. reward case
- Most organizations are committed to honoring the terms for using software – proprietary or OSS
  - If these orgs want to continue to enjoy the benefits (cost savings, tried and tested code, not reinvent the wheel) of OSS, then management must support responsible and conscientious use; therefore, typically the business case can be driven by the CTO and key FOSS users
  - For organizations where software is not the primary product, OSS is typically used on internal projects where the OSS is not modified

Creating an Open Source Policy

- Federal Financial Institutions Examination Council's (FFIEC) Guidance (Financial Institutions Letter (FIL)-114-2004) - Risk Management of Free and Open Source Software
  - "The federal regulatory agencies believe that using FOSS does not impose risks to institutions that are fundamentally different from risks presented by proprietary or self-developed software. However, acquiring and using FOSS necessitates that institutions implement unique risk-management practices."
- Key policy tenets
  - Risk level based
    - Look for specific higher risk events such as distribution or modification
  - Highly indexed to the Legal department
  - Sections: legal, acquisition, usage, support, management, partner
- Policy education
  - Created a webinar which was required training for developers
  - Conducted general educational sessions
- Policy maintenance
  - Updated annually by FOSS Center of Excellence members
  - Recent revisions include OSS Contribution tenets

Open Source Management

- Why manage OSS?
  - To realize the benefits while mitigating risk
  - Benefits
    - Solid, secured, tried and tested code
    - At the right price – including support and functional gap development
    - Competitive leverage
  - Risks
    - IP infringement
    - IP loss
    - Effect to brand

- How to tie into the Corporate process?
  - Use existing processes
  - For example: the Enterprise Architecture (EA) process
    - All software is subject to EA approval
    - FOSS process is part of the overall EA process

The FOSS Management Program

- FOSS management program is an ‘electronification’ of FOSS policy requirements
  - Governance Charter
    - Define and establish consistent policy and best practices for Open Source software operations in order to manage Open Source legal, operational, and strategic risk
  - Advocacy Charter
    - Promote innovative Open Source solutions which enable cost reduction and a speed to market
    - Foster internal collaboration to create communities, improve experience of using Open Source and reduce variation
  - Open Source Management Portal (OSMP)
    - Request FOSS usage, download, view FOSS Metadata, scan for FOSS
- FOSS team structure
  - Handful of associates
    - Half focused on licensing and compliance
    - Half engaged in internal development work using Open Source
  - Serve several thousand technical associates

Getting Help

- Support types
  - Self-support or generalist support for developer frameworks
  - Specialty support for infrastructure or critical software
- Self-support software (depends on end-user capability)
  - Examples: IDEs, libraries, frameworks, non-mission critical apps
- Generalist support (e.g. from OpenLogic) software
  - Examples: IDEs, libraries, frameworks, non-mission critical apps
- Specialty support (e.g. from Red Hat) software
  - Examples: Operating systems, databases, application servers
- When do you need support?
  - Company culture drives support requirement; however, most medium to large sized companies will need to augment self-support with some commercial support
- At the bank we use all three types of support options

Risk Acceptance Process

- Generally, to meet business requirements it is often not necessary to engage in high risk FOSS activities (e.g. distribution, modification)
- Risk management approach
  - Rate risk usage as high, medium or low depending on:
    - FOSS components (e.g. licenses, maturity)
    - FOSS application usage (e.g. internal use only, or distribution)
  - Risk acceptance commensurate with risk rating; for example,
    - Higher ratings to be reviewed and accepted by senior management
    - Lower ratings assumed to be accepted by application managers
  - Use the Open Source Management Portal (OSMP) to initiate and track risk acceptance
  - Tie this process with the overall Enterprise Architecture (EA) software request process
  - Plans are to move to a process which relies less on “volunteer” requests and more on automatic discovery and auto-generated risk calculations and high risk escalations

Remediation

- Existing use
  - For those applications which have FOSS usage dated prior to the policy's implementation

- Approach
  - Create an application Open Source inventory repository
    - Track via your own manual list or a vendor tool (e.g. OpenLogic)
  - Document your organization's application list
  - Request application teams to review their current FOSS usage and compare it to the application Open Source inventory repository
  - Ask application teams to update the Open Source inventory
  - Have an annual or semi-annual certification exercise
    - Attest that the Open Source inventory is accurate
  - Work to directly integrate the FOSS process with routine developer activities
    - Less reliance on "volunteer" compliance

Distributing & Contribution to OSS

- These types of activities may result in a distribution
  - Consumer/Commercial distributions
  - Mobile
  - Innovation
    - Establish prior art
    - Orientate/direct industry to out-of-the-box solutions
  - Divestitures
  - Feature enhancements (maybe bug fixes) to existing projects
  - Non-software transfer community interactions (e.g. business Intellectual Property (IP))
- Derive distribution & OSS IP contribution and community interaction policy statements
  - As compared to internal use (inventorying), spend more time analyzing distributions and contributions
  - Distinguish by types of contributions (software vs. business; contribution to existing community vs. running your own)
  - Manage conflict of interests, brand impact, community obligations

Looking Forward

- Tie identification and governance tasks to the typical daily activities of the developers
  - If possible, don’t ask them to do approval requests
  - Instead make that automatic
    - Tie into tools – Software Configuration Management repositories, Maven, CPAN, build tools/processes
  - Bring critical issues to the developer’s attention
    - Example: Internal unmodified use of known and approved licenses simply needs to be documented
    - Example: Use of licenses which have not been vetted can become critical
- Keep in mind
  - A journey, not a destination; think “gradual” consistent progress
  - Often you are managing perception - e.g. fear and uncertainty
  - Make it palatable and the preferred choice to use OSS in your org.
    - Get appropriate support and coverage (e.g. indemnification)
    - Govern OSS as well if not better than commercial
    - Focus the attention to the benefits and rewards, not the Fear, Uncertainty and Doubt (FUD)

Bank of America

Corporate signature
Do Not print this page.
For projector presentations only.