Quality Pedigree Programs: Or How to Mitigate Risk and Cover Your Assets

Presented by Marc Visnick, Susan Courtney, & Barb Frederiksen
What We’re Going to Talk About…

• Technology / legal level-setting
• Good pedigree practice: definition
• Good pedigree practice: implementation
What is 3rd Party Code?

- Code your company didn’t write
  - Commercially-licensed IP (binaries, source)
  - Open Source software
  - Samples from books, articles, Internet
- Code your company wrote as Work for Hire
- Code your employees wrote for past employers
You Have a Problem With 3rd Party Code if:

- You don’t know you are using
- You don’t know what you are using
- You don’t know the license terms
- You haven’t identified where you are using it or whether you are distributing
Why Should You Care?

• Litigation risks (©, trade secret, patent, breach of license)
• Valuation implications
• Community stigma & market erosion
• Security / support concerns
Technology Level-setting

- How and where 3rd-party software interacts with your software, matters. A heck of a lot. Truly.

- This question relevant to all types of potential litigation risks
  - Most open source entanglements appear to relate to © infringement and/or breach of license
  - Don’t ignore patent / TS risks
Technology Level-setting: Compiling

Source Code → Compiler → Object Code
Technology Refresher - Linking

• How to create an executable program?
  – Static Linking
  – Dynamic Linking
Why This Matters…

• Have I created a derivative work?
  – Static Linking
  – Dynamic Linking
The Distribution Trigger

- Many license terms kick in on distribution
  - What about internal use?
  - Distribution of product to customers?
  - Web-hosted applications?
What is Open Source?

• A software licensing model generally predicated on certain “Freedoms:”
  – Freedom to run a program, for any purpose
    • No discrimination against people or technologies
  – Freedom to study how a program works
    • Access to source code
  – Freedom to redistribute original source code
  – Freedom to make & distribute derivative works
Open Source License Types

• Two general categories of open source licenses:
  – Permissive Licenses
  – Reciprocal ("copyleft") Licenses
Permissive Licenses

• MIT, BSD, Apache License, *etc.*
• Focus is usually on downstream attribution
Reciprocal Licenses

• “Copyleft,” “Share-Alike,” “Viral”
• Examples: GPL, LGPL, Mozilla Public License
• Use my stuff if you’d like, but then you must “share-alike”
An endless litany of non-compliance…

- Hundreds of M&A reviews, have evidenced countless examples of:
  - Failure to comply with license terms
    - Open source and commercial terms
  - Failure to document use of open source
  - Code (license) laundering
  - Use of “trial” versions of commercial software in production code
  - Failure to consider implications of using open source in certain parts of product
Process & Documentation

- Code Audit
- License Review
- Business Decision
- Compliance Status
- Process & Documents
Define a Policy

• Have a published policy for use of source code that defines where and how Open Source / 3rd Party code can be used
  – Define criteria for approved and unapproved 3rd party software/licenses.
  – Define scope of the policy: internal development, ICs, vendors, etc.
  – Define the approval processes and the process owners
  – Establish required artifacts for traceability and compliance
Establish Accountability

• Form a core compliance team:
  – Legal
  – Open Source / 3rd party code Compliance person
    • Overall responsibility
    • Final approval authority
  – Technical / engineering management / architect
    • help identify OSS code
  – Business stakeholder
Modify Development Lifecycle

• Re-model software development life cycles to include 3rd party code review checkpoint
  – E.g. Buy vs. build scenarios
• At the point a “coding solution” is proposed / known is the appropriate time for the OSS/3rd party review
Establish Traceability

- Document 3rd party / Open source code used
  - Code version and origin
  - Maintain a copy of unmodified original code
- Trace 3rd party / OS code to products/packages
- Maintain copies of licenses
- Use Black lists / white lists / gray lists
  - Same code may be white listed for internal use, black listed for distributed products
Embrace Pedigree Program

- Educate your employees and independent contractors
- Review, revise, communicate related notices
  - Keep compliance on the radar
- Audit compliance artifacts regularly
- Use an internal policing process
Code Audit

- Process & Documents
- License Review
- Business Decision
- Compliance Status

Code Audit
Code Audit

• A way to find out what you own
• Audit code to identify 3rd party materials, how they are used, and where they came from
• Relies on automated tools and visual inspection
  – “Trust but Verify” applies to any automated tool / process
• Identify technical issues
  – Scope of use, security vulnerabilities, lack of documentation
License / Legal Review

• Evaluate requirements and risks

• Assess license(s) to determine restrictions
  – External commercial use / Internal use / Non-commercial

• Identify other risks
  – Exposure to loss of TS
  – Patent issues
  – Other litigation risks
Compliance Status

- Code Audit
- License Review
- Process & Documents
- Business Decision
- Compliance Status
Compliance Status

• Are you compliant with license terms?
• If not, what actions must you take?
• Who will be involved in assessment?
• How can compliance requirements be met?
Business Decision

• What should we do to remediate a problem(s)?
  – Remove, replace or refactor?
  – Allow use & come into compliance?

• Any limits on how/where code can/should be used?
  – E.g., internal v. distributed?
Back to Process & Documentation
Questions?

Susan Courtney
susan@jli.com

Barb Frederiksen
barb@jli.com

Marc Visnick
marc@jli.com