That’s Not Your Code!
Jacobsen v. Katzer and Open Source Licensing

Significant Developments in Computer & Cyberspace Law
June 12, 2009

Diana J.P. McKenzie
Partner and Chair – Information Technology Practice Group

Cell: (847) 370-1121
dmckenzie@huntermaclean.com
Roadmap

- Open source software
  - Open source on the rise
  - Open Source Initiative
  - Open Source Definition
  - Open standard requirements
  - Types of open source licenses

- Jacobsen v. Katzer
  - Basic Facts
  - Decision
  - Reasoning
  - Significance

- Mitigating the Risks
  - Information gathering
  - Legal review and analysis
  - Open source submission forms
  - Open source policies
Open Source Software On the Rise

- $3 Billion in venture capital invested in 163 open source firms between 1997 and 2008
- Nearly 40% of US firms surveyed between Dec. 2008 and Feb. 2009 are implementing or expanding use of open source software
  - Another 25% are piloting or considering use of open source software
- Red Hat grew by 18% in the Q1 2009, despite (or because of?) recession
Open Source Initiative

- California public-benefit corporation founded in 1998
- Sets the community-recognized rules for open source licenses
  - “Open Source Definition” (OSD)
- Reviews and approves licenses as OSD-conformant
- 65 licenses approved as of June 2009
Open Source Definition

- Free redistribution under the same license
- Source code distributed or available
- Permit modifications and derivative works, and allow them to be distributed under same license
- Distribute license with the program
- No discrimination
- Right to extract the code
- No restrictions on other software
- No restrictions on technology or interface
Open Standard Requirements

- Standard = detail necessary for interoperable implementation
- Publicly available under royalty-free terms
- No royalties for patents essential to implementation
- No requirement for execution of an agreement
- Must not require technology that fails to meet criteria
Additional Restrictions

- Each individual public license may place additional restrictions
  - Beyond those required under the OSI definition
  - Each license is different
- Example: the GNU General Public License (GPL) prohibits distribution with software components licensed under the Mozilla Public License
Types of Open Source Licenses

- Hereditary or “strong copyleft” licenses
  - GNU General Public License (GPL)

- Permissive or “non-viral” licenses
  - Berkeley Software Distribution (BSD)
  - Apache Software License
  - MIT (X11) License

- Hybrid or “weak copyleft” licenses
  - GNU Lesser General Public License (LGPL)
  - Mozilla Public License (MPL)

- “Otherware”
  - Beerware
  - Catware
  - Postcardware
Jacobsen v. Katzer: Basic Facts

- Ct. of Appeals for the Federal Circuit – August 2008
- Jacobsen and Katzer develop and distribute software for controlling model railroads
  - Robert Jacobsen
    - Physics professor at University of California, Berkeley
    - Makes his software for the Java Model Railroad Interface Project available under an open source license
  - Matthew Katzer
    - Owns Kamind Associates, Inc. and sells commercial products
    - Copies some of Jacobsen’s software and uses it in his proprietary product
Jacobsen v. Katzer: Basic Facts

- Open source license at issue: Artistic License version 1.0
  - Rarely used; poorly worded
  - Originally written to provide rights to PERL programming language interpreter
  - Permissive license; requires:
    - Copyright and license notices
    - Identification of changes to original source code
    - Allows alternatives to relicensing of modifications on open source terms

- Katzer violated the terms of use
  - Failed to reference the original source of the code
  - Failed to identify how the original source code was modified
Jacobsen v. Katzer: Decision

- Real issue – can the licensor sue the licensee for copyright infringement? Generally:
  - If the breached term is only a contractual “covenant” → NO, can only sue for breach of contract
  - If the breached term is also a “condition” on the license → YES

- Longstanding issue in the open source community

- Why it matters: remedies
  - Since open source software is distributed for free, quantifiable damages under breach of contract most likely equals zero
  - But copyright infringement allows for injunctive relief, statutory damages, costs and attorney’s fees
Jacobsen v. Katzer: Holding

- **Holding:** The terms of the Artistic License are enforceable copyright conditions; Jacobsen can sue for copyright infringement.

- **Implication:** A software developer who uses open source code in violation of its license can be held liable for copyright infringement.
  - A copyright owner who grants a nonexclusive license waives his right to sue for copyright infringement, unless the license is conditioned on compliance with its terms.

- **Note:** case does not hold that open source licenses are not contracts (this was not at issue).
Jacobsen v. Katzer: Reasoning

- **Language of the Artistic License**
  - Preamble: “the intent of this document is to state the **conditions** under which” the code may be copied
  - Terms allow users to modify and distribute code, **provided** that certain conditions are met
  - “Provided that” denotes conditional language under California law

- **Substantively, the conditions in the Artistic License are vital to enable Jacobsen to benefit from the work of downstream users**
  - This “downstream benefit” is Jacobsen’s consideration for sharing his code
Jacobsen v. Katzer: Significance

- Case viewed as a victory for open source licensing
- Court of appeals followed guidance of the open source community on a key legal issue
- No real surprises
  - Most lawyers assumed violation of an open source license carries risk of copyright infringement
- Publicity of the case focused attention on importance of risk mitigation for using open source software
- Ironies
  - Obscure open source license
  - Ct. of Appeals
  - Originally a patent claim
Mitigating the Risks

- Two steps
  - Information gathering
  - Legal review and analysis
Mitigating the Risks

- Information gathering
  - Provenance checking (cheaper)
    - Record and follow precisely what version of what code is used
    - Examine how code is used
    - Determine which license the code is subject to
  - Code scanning (more expensive)
    - Use of automated tools to find third party code
      - Simple: search for copyright notices
      - Complex: search for matching algorithms
    - Hire an open source consultant that specializes in code scanning
Mitigating the Risks

- Legal review and analysis
  - Develop an open source submission form
  - Develop an open source software policy and apply it
    - Still controversial, but probably a best practice (if followed)
    - Most policies classify open source licenses into 3 categories:
      - Always approved – usually applied to permissive licenses
      - Never approved
      - Requires further legal review – usually applied to GPL and LGPL
Mitigating the Risks

- **Open source Submission Forms**
  - A form for engineers to complete before using open source software
    - Name of software, release/version, how/where obtained
    - How software will be used/integrated
      - Static or dynamic linking?
    - To whom will the open source software ultimately be distributed?
  - Process involves purchasing and contract administration
    - Often avoided because no fee payment is required
  - Determine how you intend to comply if the license:
    - Contains attribution/notice requirements
    - Requires source code to be offered on request
    - Requires distribution of derivative works
Mitigating the Risks

- Open source policies
  - Still controversial, but probably a best practice (if followed)
  - Policies vary considerably
  - Contents typically include:
    - General approach
    - Categorized lists: acceptable, not acceptable, TBD
    - Notices
    - Interplay with patents
    - Tools for tracking
    - Procurement requirements
    - Modifications process
Mitigating the Risks

- Record-keeping is important
  - Critical in avoiding evaluating open source licenses multiple times
  - But note that re-evaluation may be necessary if open source code is used in a different product or used differently in the same product

- Watch out for client requests for opinion letters
  - Too many uncertain legal principles
Questions

Diana J. P. McKenzie
Partner and Chair, Information Technology Practice Group
dmckenzie@huntermaclean.com