Allocation of the Risk

by Lawrence Rosen

A license serves to allocate risks between the licensor and the licensee. This is an essential purpose of license warranty provisions.

The BSD license, for example, contains the following clause:

> This Software is provided by the copyright holders and contributors "AS IS" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services, loss of use, data, or profits, or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

(This clause was originally in all caps to make sure every licensee reads it. I find all caps text to be unreadable, so I rewrote it in upper/lower case.)

The important thing to notice is that all risks under the BSD license are borne by the licensee. No matter how dreadful the software, no matter what damage it causes to your computer or your business, no matter what promises the licensor advertised about the usability or functionality of the software – the licensee accepts the software "AS IS" without any warranty.

All other open source licenses I've read contain similar disclaimers of warranty and limitations of liability. The licensee bears all the risks associated with using the software and creating derivative works from it.

I think that is a fair allocation of risks, except for one risk that isn't expressly mentioned in the BSD license. I'm concerned about the risk that the licensor isn't really authorized to grant the license to the software in the first place. The BSD license (because of the "including, but not limited to" language) excludes this *warranty of non-infringement*.

Without a warranty of non-infringement, if the licensor doesn't actually own the copyright to the software he licenses, or isn't acting under the authority of a license from someone who does own the copyright, the licensee, not the licensor, is accepting the risk of a lawsuit for copyright infringement when he accepts the software.

Here's how I corrected that problem in a new license I'm writing called the Academic Free License:

> Licensor warrants that the copyright in and to the Software is owned by the Licensor or is distributed by Licensor under a valid current license. Except as expressly stated in the immediately preceding sentence, the Software is provided by the Licensor, distributors and copyright owners "AS IS", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall the Licensor, contributors or copyright owners be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the Software.

In the first sentence of that warranty clause, the Academic Free License allocates the risk of copyright infringement to the licensor rather than to the licensee.

Here's why I did that: As between the licensor and the licensee, I'm convinced that the licensor is in a far better position to know whether he owns the copyright or has the software under a license that allows him to license it to others. The licensee, on the other hand, has no information on which to base a decision whether to accept the risk of non-infringement. The risk is allocated to the only party that is capable of determining the degree of risk.

For example, if the licensor knows he wrote the software himself, he owns the copyright. If he knows he received the software under the GPL or some other open source license, he is assured that he is distributing the software "under a valid current license." But if he merely copied the software from someone else, or created the software while working as an employee for another company, he can't honestly warrant against non-infringement.

Licensees may be unwilling to accept open source software without some assurance from the licensor that they are not licensing stolen intellectual property. The absence of such assurances may inhibit the acceptability of open source software. With a warranty of non-infringement like the one I wrote for the Academic Free License, licensees can at least reasonably rely on the license language to protect themselves from accusations that they didn't care about who actually owned the intellectual property.

What would happen if the licensor gave that warranty of non-infringement but didn't actually own the copyright or didn't actually have a valid license to distribute the software? Damages for breach of warranty can be substantial. In appropriate situations, a licensee can recover for any loss resulting from the breach, the difference between the value of the software accepted and the software delivered, and even incidental and consequential damages.

I'd be very interested in your thoughts about whether open source licenses should include warranties of non-infringement. Please send your comments via email to lrosen@rosenlaw.com. I'll report on the results of this informal survey in a later column.

Lawrence Rosen is an attorney in private practice, with offices in Los Altos and Ukiah, California (www.rosenlaw.com). He is also executive director and general counsel for Open Source Initiative, which manages and promotes the Open Source Definition (www.opensource.org).

