

## IT Analysis – New Issues in Software Compliance

By Clive Longbottom, Service Director, Quocirca Ltd

For many companies, software compliance is just about making sure that all copies of a particular application in use have a valid vendor licence. There's plenty of software around to try and ensure that all licences are valid and all users covered, to stop organisations like the Federation Against Software Theft (FAST) from claiming damages on behalf of its members for fraudulent use.

However, something that is far more insidious and dangerous is being seen as developers start to use chunks of reusable code in creating their own applications, composite applications and mash-ups.

The problem lies within the various licencing agreements that there are around software made available for public usage. As well as the main ones, such as the General Public Licence (GPL), the Apache Software Licence (ASL) and the Eclipse Public Licence (EPL), there's a whole host of others, such as the Common Development and Distribution Licence (CDDL), the Fair Licence and the Beer Licence (where you undertake to buy the licensor a beer if you ever meet him).

The majority of the public use licences are not particularly restrictive, but there are conditions in some of them which can lead to nightmares for companies further down the line. Let us take as an example the GPL licence in its current version 2 form. Essentially, under section 2 of the Terms and Conditions of Copying, Distribution and Modification, if you happen to utilise a piece of code that has been distributed under the GPL—just one piece—then the whole of the final released code that includes that GPL code in it also has to be distributed under the GPL.

Consider the ramifications of this—as a commercial entity, you want to create functional software as rapidly as possible. Functions which are deemed as "commodity" are available from the web, and it makes sense to utilise these functions to speed up overall development. Your developers then come up with the next "killer application", and you start selling it on the open

market. Just as you're going for IPO, someone in the community points out that 100 lines of code used within your 1 million lines were licenced under the GPL. All of a sudden, all your 1 million lines of code have to be covered by GPL—the code has to be made available to anyone, and you cannot apply restrictions on the redistribution of such code—or its modification. You still own the copyright—but that's no use when your customer base and those who you were hoping would invest in you are deserting you in droves.

Even if you are in the Open Source movement yourself, you may want to utilise a different licence for the final code—but you may find that you are forced into a specific one through the code chunks that have been utilised in one part of the overall application. As the move towards utilising more of these publically available pieces of functional code to speed up development accelerates, there's the risk that we spend more time either checking through all the different licences that have been used—or fighting various actions in court.

So, what can be done? Prevent your developers from using pre-written reusable code chunks? Not really—such usage means that time to capability and to market is much enhanced. Ensure that all code is held against a copy of the licence, and then distribute each chunk under its own licence separately? Not viable, and anyway, the developers won't read the licences.

A small US company, Black Duck, seems to have a solution. It provides a capability for code to be searched at both a string matching level and at a pattern matching level to identify code that has come from an environment where the code is licenced. It can then flag all of these pieces of code and ensure that the developer or legal department is aware that this may raise issues. The developer and/or organisation concerned can then make a decision as to how this is all rolled up—does the overall code go out under a specific licence, does the intellectual property code get packaged separately to the open code so as to maintain the fidelity of the commercial code, or is the code that has been utilised to be

replaced with in-house code so as to by-pass the possible ramifications of the other licenced code?

All of these are valid options, and knowing what code is problematic, what all the various licences are, and what the ramifications are means that time to capability and market are not compromised, while the business value of the code is maximised. Also, end users are safeguarded against claims against licence misuse—a point that has taxed many who were originally caught up in SCO's patent and licence claims around Linux.

For many, running such a capability at the end of development may be sufficient, but Black Duck's tooling provides the best return by integrating it into the whole development process, so that a licence audit trail can be kept—and so that any possible issues can be dealt with as soon as possible.

Black Duck could take this approach further to look at areas such as digital rights management (DRM) around picture, sound and video, increasingly sensitive areas as such content proliferates on the internet. Although the domain skills would need to be built up, the approach of scanning files for direct copyright notices and for pattern matching against known copyright content will be similar. That Black Duck has chosen to focus for the moment, and not to stretch itself too thin is probably sensible, but OEM deals for the underlying technology should not be too far off.

All told, the area is a minefield for the unwary. For software developers, whether in the ISV community, within end-user development groups or the open source community itself, such a facility has solid value, and a product like Black Duck's is well worth considering.

## About Quocirca

Quocirca is a primary research and analysis company specialising in the business impact of information technology and communications (ITC). With world-wide, native language reach, Quocirca provides in-depth insights into the views of buyers and influencers in large, mid-sized and small organisations. Its analyst team is made up of real-world practitioners with first hand experience of ITC delivery who continuously research and track the industry and its real usage in the markets.

Through researching perceptions, Quocirca uncovers the real hurdles to technology adoption – the personal and political aspects of an organisation's environment and the pressures of the need for demonstrable business value in any implementation. This capability to uncover and report back on the end-user perceptions in the market enables Quocirca to advise on the realities of technology adoption, not the promises.

Quocirca research is always pragmatic, business orientated and conducted in the context of the bigger picture. ITC has the ability to transform businesses and the processes that drive them, but often fails to do so. Quocirca's mission is to help organisations improve their success rate in process enablement through better levels of understanding and the adoption of the correct technologies at the correct time.

Quocirca has a pro-active primary research programme, regularly surveying users, purchasers and resellers of ITC products and services on emerging, evolving and maturing technologies. Over time, Quocirca has built a picture of long term investment trends, providing invaluable information for the whole of the ITC community.

Quocirca works with global and local providers of ITC products and services to help them deliver on the promise that ITC holds for business. Quocirca's clients include Oracle, Microsoft, IBM, Dell, T-Mobile, Vodafone, EMC, Symantec and Cisco, along with other large and medium sized vendors, service providers and more specialist firms.

Details of Quocirca's work and the services it offers can be found at

<http://www.quocirca.com>