



Michael Wigley

Critical need to protect software IP

MINIMISING THE RISK of software IP problems is important for organisations, including ensuring ongoing availability of software through ownership or licence. But organisations often don't nail this, relying on slim arrangements or vendor-friendly agreements.

Government recently announced it would change the "commissioning" rules in the Copyright Act, to

The problem would have been avoided if the customer had made sure ownership rights were documented clearly, (ideally in an agreement but, at least, in an exchange of email).

So it's not wise to rely on the default commissioning rules. Document it.

Many customers will instinctively think it is best that they own the software, not the devel-

to a better road map for new releases with quicker and better bug-fixes, etc. Encouraging the developer to have a broad customer base, will make for more robust and up-to-date software. This is the way that ERP providers such as SAP and Oracle operate. They incorporate code cut for one client in future releases of software (where that improves the base product). The base product is already improved by customised code previously built into the standard offering. It's an ongoing cycle of development; enabled because the ERP provider, which was paid for the customised solutions, uses them for other clients. There's no detriment to the customer in this. Rather there is benefit. So it's better for the ERP provider to own the code. The provider gives a licence to the customer.

By the way, the idea of joint ownership between developer and customer is too messy a solution. There are better ways.

So, often all that the customer needs is a wide licence to use the software, not actually ownership. This option has also recently been recognised in New Zealand Government guidelines.

But doesn't ownership reduce risk as the organisation gets control and also the source code? Often not. Ownership can be illusory in terms of reducing risk.

The developer typically will have cut code, which operates on top of existing software that the developer owns and provides to other customers. The additional

code can be useless without the underlying code. Also, for complex or badly-drafted code, having the source code can be hard to utilise when the developer has failed or the relationship has ended.

That highlights the desirability of looking at options such as escrow, and also ensuring the software is developed using best practice standards (so it is easier for other developers to run with it).

That's easier said than done. Vigilance is needed. For example, escrow agreements must be robust. Many aren't. Also, the organisation needs reassurance that up-to-date source code, documentation, etc, is actually lodged in escrow (and continuously updated).

There are so many factors in software IP (proprietary software, open source, library code, bespoke code, etc) that cookie cutter IP clauses in contracts aren't usually sufficient. Each situation needs thought and care. ■

There are so many factors in software IP that cookie cutter IP clauses in contracts aren't usually sufficient. Each situation needs thought and care.

standardise the approach. Where an organisation commissions another to develop IP, the default position will be that the party creating the IP owns the IP.

If this becomes law, it changes the reverse default position for software. Under current law, the default position is that the customer owns the IP in the software.

However, these are just default rules that can be overridden by oral or written arrangements.

A discussion or an email can give ownership to an unintended party instead.

That's a real problem, with disputed evidence of "who said what" taking one case to the Court of Appeal, to decide if the software developer or its customer owned the software (ie whether the default commissioning rule was overridden).

oper. However, that's not the only, or even the best, solution in all situations. After all, most organisations want the software to work long-term in their own businesses. They won't on-sell it. So do they really need to own the IP?

Developers can commercialise it. If they own the code cut specifically for a customer, they can use it to make more money down the track by selling to others. On-going development of code, built on previous code, makes sense and is standard anyway. It's 'standing on the shoulders of giants'. If the developer can reuse the code because it owns the code, the price to the customer could reduce as well, to its benefit.

More importantly, developing similar software for others (except the customer's competitors) leads

Michael Wigley is the Principal of Wigley & Company, a law firm specialising in ICT. He can be reached at michael.wigley@wigleylaw.com

If there is a question you would like Michael to answer in relation to IT issues, please forward it to dparedes@fairfaxbm.co.nz