

From: Kevin McFadden [e-mail redacted]
Sent: Tuesday, September 28, 2010 4:04 AM
To: Bilski_Guidance
Cc: [e-mail redacted]
Subject: No to software patents

In brief:

- software is a combination of art and science, both stand on the shoulders of giants
- no one writing software is innovating with the explicit goal of patenting their code
- innovation is stifled when bureaucracy interferes in favor of the rich and powerful, the process is long and complicated for both application and especially defense
- almost all software is like plumbing as there are many standard approaches
- patents last significantly longer than their usefulness since the software industry changes so fast

Software patents hurt individuals by taking away our ability to control the devices that now exert such strong influence on our personal freedoms, including how we interact with each other. Now that computers are near-ubiquitous, it's easier than ever for an individual to create or modify software to perform the specific tasks they want done -- and more important than ever that they be able to do so. But a single software patent can put up an insurmountable, and unjustifiable, legal hurdle for many would-be developers.

*The Supreme Court of the United States has never ruled in favor of the patentability of software. Their decision in *Bilski v. Kappos* further demonstrates that they expect the boundaries of patent eligibility to be drawn more narrowly than they commonly were at the case's outset. The primary point of the decision is that the machine-or-transformation test should not be the sole test for drawing those boundaries. The USPTO can, and should, exclude software from patent eligibility on other legal grounds: because software consists only of mathematics, which is not patentable, and the combination of such software with a general-purpose computer is obvious.*

Sent from my iPad