From: David Barnard [e-mail redacted]
Sent: Saturday, September 25, 2010 7:40 PM
To: Bilski_Guidance
Subject: Software Patents

I'm a small, independent software developer and live in constant fear that I might spend months and thousands of dollars implementing a feature (or entire app) that was somehow patented by another developer or patent troll. Even if their patent is weak or tangental, there's no way for me to even consider putting up a fight.

After releasing my first app (a mileage log called Trip Cubby) for the iPhone in 2008, I almost immediately received a threat from a company that said it owned a software patent on tracking mileage via GPS and reporting it via the Internet. My app didn't have that feature, but it's something I was planning to add at some point in the future. I still haven't implemented that feature, and the company that emailed me still hasn't created an iPhone app. Software patents stifle innovation and intimidate the small independent developers making some of the most fun, productive, cutting edge software.

Even though I could patent some of the innovations I've come up with, I've made the conscious choice to not pursue patents. I fully agree with the EFF's stance on the matter:

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.

Thanks,

David Barnard
App Cubby
(512) 212-4860