From: John Ries [e-mail redacted]
Sent: Saturday, September 25, 2010 11:32 PM
To: Bilski_Guidance
Cc: [e-mail redacted]
Subject: Software Patents

My name is John Ries and I have been employed full time as a computer programmer for the past 18 years. I have for 17 years been employed by Salford Systems, a small statistical consulting and software development firm based in San Diego, California. As I am writing on my own behalf, I am doing it from my personal e-mail account, rather than my work account. My opinions on this subject are not necessarily those of my employer.

I am given to understand that the USPTO is seeking guidance on how to implement the recent Supreme Court decision Bilski v Kappos. While I am not a lawyer and have no legal training, the issue of software patents affects me and every other software developer in the United States.

My employer has already been threatened once with a patent infringement lawsuit over code implementing an algorithm which we developed internally without ever having heard of the supposed invention referenced in the patent (which is, as far as I am concerned, direct evidence of the triviality of the patent). My employer is small (approximately 30 employees; much too small to have a legal department), so I ended up spending many hours searching my e-mail for evidence that we developed our algorithm before the patent application was filed (which in fact, we did). This is time that I would otherwise have spent doing the development, analysis, and database management work which are my normal functions. I suspect that others of my colleagues ended up doing similar research as we prepared our defense. This is a productivity drain that can be tolerated once every few years (maybe) but if it were to become a frequent occurrence (and there are now any number of firms whose sole business is buying obscure software patents for the purpose of suing developers and users) then my employer would surely be driven out of business. The situation would have been far worse had we been a recent startup, as that single threat would likely have been sufficient to cause my employer to close permanently.

In Bilski v Kappos, the U.S. Supreme Court held unanimously that abstract concepts are not patentable. The business of writing computer software is fundamentally the business of implementing mathematical algorithms, which are abstract concepts by definition. Dressing up the patent application by claiming that a general purpose computer running a program implementing a particular algorithm constitutes a patentable invention does not change that fact and is a bit of legal sophistry that neither the courts nor the USPTO should ever have tolerated. The time to put an end to it is now.

The problem is compounded by the fact that many thousands of seemingly trivial patents are granted each year. They are accumulated by large corporations, as well as non-practicing entities and used as legal weapons against both small companies and independent developers, who can ill afford to pay lawyers to defend themselves (especially in a far away district court, as often happens). This has the potential effect of

greatly reducing competition in the computer software industry and working against the "advancement of science and the useful arts", promotion of which is why the US Constitution authorizes Congress to grant copyrights and patents in the first place. It should be noted that the great flowering of software development in the 1970s and 1980s occured *before* the legal precedents were set that made software patentable. I don't think that any reasonable person can argue that the practice of granting them has in any way encouraged further innovation in the software industry.

I strongly believe that software patents have been bad for the industry and pose a threat to the livelihood of every software developer in the United States who is not employed by either a large corporation or governmental entity. The problem is compounded by the ease of obtaining such patents. At a minimum, the proper response to Bilski vs Kappos would be the implementation of proper controls to insure that patents are not granted for trivial inventions, inventions based on abstract concepts, or for overly broad claims. In regard to that, it would be well if patent applications were reviewed by people who have real expertise in the field of knowledge that the supposed invention claims to advance (if fees have to be raised significantly to permit the hiring of such experts, then so be it). As computer software consists solely of implementations of mathematical algorithms, which are by their very nature abstract concepts, it is my opinion that they are not patentable at all, but if they are to be granted, then at least let them be limited to a few real advances on the state of the art that are not likely to be infringed upon by accident (which is what I believe is what the Framers intended).

Finally, while I have written this message at the behest of the Free Software Foundation, it should be stressed that the opinions expressed here are entirely my own and are likely at variance with those of the FSF, with which I have no affiliation Also, I do not believe that the closed source proprietary model of software development so strongly opposed by the FSF and its founder Richard Stallman to be in any way immoral. I am employed by a proprietary software developer and believe that my employer's products are well worth licensing by statisticians and data analysts. That said, I believe that the free software movement has performed an enormous public service by opening up a software market that had become largely noncompetitive and making high quality software available to those who could not possibly have afforded it otherwise.. It deserves to continue without the threat of lawsuits over software patents.

Sincerely yours...

John L. Ries
1733 N. Main St.
Cedar City, UT 84721