

From: Greg McKeever [e-mail redacted]
Sent: Wednesday, October 06, 2010 8:31 AM
To: Bilski_Guidance
Subject: Software Developer's Perspective

To whom it may concern:

I would like to begin by apologizing for missing the September 27, 2010 deadline for submissions regarding requests for comments pertaining to software patents. I have been very busy with my software development activities as I approach crunch time on my project and I only learned of the USPTO request for comments yesterday. I would also like to state that I welcome your fresh approach of requesting comments from people in the software development industry before making such important decisions that will affect how an entire industry conducts its business. It is refreshing to see an organization appeal to its constituents directly for comments and I hope you have are gaining valuable information from my fellow software developers.

I would like to provide a quick background about my interest in software patents. I have been developing software for nearly a decade and from the beginning of my career I have realized the monumental impact patents have had on the software industry. Over the years, I've constantly read about large companies battling each other in court over trivial software patents. Worse yet are the large companies that drive small companies out of business by accusing them of infringing one of their software patents. These lawsuits cost million of dollars to defend and they are often based on dubious claims. While patents on physical inventions provide invaluable protection to the inventors who drive the technology industry, they have the opposite effect on software. In this letter, I hope to differentiate the effect of patents on physical inventions versus software and how patents on software are unnecessary and often detrimental to our industry.

As I understand patents, they exist to protect concrete inventions and not abstract ideas. They provide an inventor a government-granted temporary monopoly on an invention in return for divulging the details of how the invention works. The inventor benefits from the patent via

the protection it provides for all of the time spent researching and developing the invention. The detailed information within the patent provides our society with the insight of how to implement the invention without having to perform the research and development all over again.

In the world of physical inventions, patents are based on major scientific breakthroughs and tireless efforts to apply those breakthroughs to create useful technology. These patents contain valuable diagrams and descriptions that provide all of the important details about how to overcome the previous limits of biology, chemistry, and physics to create useful devices. The details in these patents provide a blueprint which can be used by other engineers in the field to build the invention with minimal effort once the patent has expired.

In contrast, software development is not hampered by the physical limits of our world, but by the limits of our imaginations. Software isn't developed through a small number of major scientific discoveries, but by a myriad of tiny discoveries that are made during the process of writing the code that transforms the abstract concepts into concrete applications. While software patents require design diagrams and written descriptions, these artifacts fall far short of describing the details of transforming the abstract into the concrete. Those details remain firmly planted within the source code, which is never embedded within the patent. Without the details of the source code, the entire abstract concept gets patented rather than the invention itself. This is extremely detrimental to software engineers because there can be hundreds of ways of implementing an application, but a software patent prevents all of those methods from being realized because the description and diagrams are so vague that they cover all possible implementations.

Some would argue that we should keep software patents, but require more detailed diagrams and descriptions of how the software works. While this may seem adequate on the surface, it makes much less sense when it is viewed within the context of typical software development. Software is commonly developed by establishing the requirements, creating a design, writing the source code, and testing the compiled

product. Out of these four phases, the most time is spent writing the source code and testing it. These are the two phases that require the most protection. However, software patents only cover written descriptions, usually taken from the requirements, and diagrams that are pulled from the software design. This means that the software patent covers nothing of substance since no code has been written let alone tested. Therefore the software patent only covers an abstract idea and the provided descriptions and diagrams are worthless since they likely have not even been implemented yet to be proven whether or not they are viable solutions. This means that the patents lacks any substantial information about how the software can be implemented and it prevents anyone else from creating a working implementation of their own.

While many software developers oppose software patents, we are not proposing to get rid of all intellectual property rights over software. The valuable source code that we compose to bring our creations to life is afforded ample protection through copyright laws that cover the source code, the compiled binary objects, and all supported project files and graphics that are contained within the software. Since copyright already protects the most important tools of our industry, software patents are superfluous and only get in the way of our ability to deliver quality products.

In conclusion, I appreciate your time and I hope you will consider these comments and all of the comments from other software developers. I'm sure there are some companies with large software patent portfolios who have an interest in preserving software patents to maintain a high barrier of entry for competitors. I hope you focus your attention on the comments of the people who are actually developing software and who overwhelming support the abolition of software patents. Let's put an end to patents that are stifling competition in the software industry so that we can focus on doing what we do best: developing useful software.

Sincerely,

Gregory McKeever