

**From:** [e-mail redacted] **On Behalf Of** Hunter Freyer  
**Sent:** Monday, September 27, 2010 11:00 AM  
**To:** Bilski\_Guidance  
**Cc:** [e-mail redacted]  
**Subject:** Please strongly reconsider the implications of software patents

If you ask the biggest inventors and innovators in the space of software engineering (heretofore referred to as "hackers"), software patents are a menace which at best can be ignored, and at worst can be the undoing of some really important innovation. Often the fear of patents kills projects before they can even get off the ground.

Now, the natural response to this might be, "Of course those people are upset - the government is telling them they can't profit off of other people's intellectual creations." While I'm sure this is true of some dissenters, the fact of the matter is that most hackers are upset with software patents because they provide utility to neither the inventor, nor the industry at large.

Let's consider the cost/benefit analysis of software patents in general. The trade-off for patents is that we want to risk potentially stifling innovation in a certain space for a limited time (20 years), in order to encourage innovators to publish their findings without fear of getting "ripped off". Three facts about software patents specifically tip the scales of this trade-off in a negative direction:

- \* Twenty years ago, the web didn't even exist. Any innovation developed now will be meaningless in 20 years, probably even in 5 years.
- \* Software companies deal in trade secrets, not patents. If a company has a piece of software that really contributes to their ability to succeed (heretofore referred to as "secret sauce"), they are going to keep that software under wraps. Partially, this is because the patent office can't really help that well in ensuring a monopoly on some "secret sauce". If Google were to publicly patent the most important aspects of its search engine, and Bing's quality went up, how could they determine what caused the improvement? Maybe they could win in court, but getting caught up in a legal battle over such things just sucks up money and time which could otherwise be used for further innovation. And why bother going to court to enforce a 20 year monopoly when the secret sauce goes stale after just a few years? The fact that legitimate insights go unpatented means that...
- \* Most software patents that companies obtain protect intellectual property most consider "obvious". While this shouldn't happen according to the goals of the USPTO, unfortunately, it does. This is simply because software engineering is a very new field, and potential patents can't be evaluated for originality by a patent office without enough knowledge of the software domain. It's okay. It's hard. I don't blame you.

My argument is not that software patents are inherently bad, but in their current form they do much more harm than good, and even a system with no software patents at all would be greatly preferable to the system we have now.

kthxbai,  
Hunter Freyer