KRAUSE LAW LLC
BUSINESS | STRATEGY | COUNSEL

SOFTWARE PATENTS, "ABSTRACT IDEAS," AND THE ENTREPRENEUR

This is an introduction to the topic of software patents written for the small business owner or aspiring entrepreneur. This brief resource discusses what a patent is, how it relates to software and "apps," and how to begin applying this information to your business' intellectual property ("IP").

What Is a Patent, And How Do I Get One?

The United States Patent and Trademark Office ("USPTO") grants patents. The USPTO defines a patent as "the right to exclude others from making, using, offering for sale, or selling the invention in the United States or importing the invention into the United States. What is granted is not the right to make, use, offer for sale, sell or import, but the right to exclude others from making, using, offering for sale, selling or importing the invention."[1] You might have also heard a patent described as a "legally sanctioned monopoly" or a "keep out" sign on your invention.

Next, how do you get a patent? The relevant statute, 35 U.S.C. Section 101 reads: "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title." There are many other conditions and requirements in the statute, but if your invention meets them all the USPTO will grant you a patent.

What Is the Difference Between "Validity" and "Infringement"?

These are important, but different questions. "Validity" addresses whether the patent should have been issued at all. For example, if the applicant was not truthful with the patent office or the invention does not actually meet the statute's requirements, the patent could be invalid. Infringement addresses whether someone is using the patent without permission.

Validity can be an issue for software patents.

Section 101, "Abstract Ideas," and Software Patents

A validly granted patent must fall into one of the categories mentioned above: process, machine, manufacture, composition of matter or any new and useful improvement thereof. Software inventions have a harder road, because in the landmark case *Alice Corp. v. CLS Bank International*[2], the Supreme Court ruled that the software patent in question was drawn to an abstract idea and did not fall into an eligible category and was thus invalid (not properly issued).

[1] https://www.uspto.gov/patents-getting-started/general-information-concerning-patents#heading-2
[2] *Alice Corp. v. CLS Bank International*, 573 U.S. __, 134 S. Ct. 2347 (2014)

Livestock Exchange Building 1600 Genessee St, Suite 460, Kansas City, MO 64102
816-735-0222 | www.krauselawkc.com

Since then, Courts have clarified the "abstract ideas" category to include "mathematical algorithms,[3] as well as fundamental economic and conventional business practices… even if performed on a computer."[4]

Case Comparison: Valid and Invalid Software Patents
To illustrate, here is a brief overview of two similar cases with different outcomes: *Versata v. SAP America*[5] and *Enfish v. Microsoft.*[6]

In *Versata*, the dispute involved a patent for a software program that generated multi-level pricing tables based on product and organizational groups. The patent "reduce[d] the need for large data tables by arranging customers (purchasing organizations) into a hierarchy of customer groups and products." This patent did not survive the validity challenge.

Enfish also dealt with a software that generated a data table. However, the Court wrote "the claims at issue in this appeal… are directed to a specific improvement to the way computers operate, embodied in the self-referential table… the claims here are directed to an improvement in the functioning of a computer."[7] This patent did survive the validity challenge. "In contrast, the claims at issue in *Alice* and *Versata*… simply [add] conventional computer components to well-known business practices."[8]

So, in *Versata*, the invention organized data, and in *Enfish* the invention improved computer functioning through data organization. That distinction is a subtle one, but important. If the invention doesn't affect the computer itself, or solve a problem that we can't do without a computer (see *DDR Holdings*[9]), the software app might be an implementation of an abstract idea.

How Do I Evaluate My Software Invention?
So the question becomes: what does your software do, and is it more *Versata* or *Enfish*? Does it simply use a computer, or does it affect the way the computer (or something else) operates?

For perspective, I will share some wisdom from a recent seminar: "If your software tells the computer how to move a robot arm, it might be patent eligible, but if it only tells the computer how to record the arm's movements, it's probably not."

---

[3] *DDR Holdings v. Hotels.com et al.,* Fed. Cir. 2014 at 15., quoting *Gottschalk v. Benson, 409 U.S. 63 (1972)*.
[4] *Enfish, LLC v. Microsoft Corp.,* 2016 U.S. App. LEXIS 8699, 2016 WL 2756255 (Fed. Cir. May 12, 2016).
[5] *Versata Development Group, Inc. v. SAP America Inc.* 2015 U.S. App. LEXIS 11802.
[6] *Enfish, LLC v. Microsoft Corp.,* 2016 U.S. App. LEXIS 8699, 2016 WL 2756255 (Fed. Cir. May 12, 2016).
[7] *Enfish* at 8710.
[8] *Enfish* at 8710.
[9] An in-depth analysis of the topic would include *DDR Holdings v. Hotels.com* decision, in which the software patent was upheld. However, that case turned on the idea that the software solved a problem "unique to the internet," and the case didn't bring about a significant change in Examiner guidance at the USPTO. The "unique to the internet" concept is rather complex, and *DDR Holdings* has for that reason been left out of this resource.

Should I Just Forget About a Software Patent?
No. None of the foregoing should imply that a software patent is a waste of time or money. There are plenty of strategic and legal reasons to pursue patent protection. For example, a cost-effective solution is often to file a provisional patent application[10] and test the waters with potential investors before investing the larger funds in a non-provisional application. Being able to market your app as "patent pending" will carry weight with some investors.

- *James Cole*

[10] Provisional patent applications work as placeholders. They're shorter and hold your place in the patent line for a year while you decide whether to file a non-provisional patent application. However, if you don't file a non-provisional application, the provisional expires after a year.