ROOTS TO BITS: HOW THE HISTORY OF PLANT PATENTS CAN SHAPE SOFTWARE'S FUTURE

TREVOR BERVIK*

Software patents and patent applications have become inundated with subject matter eligibility concerns. The Supreme Court, as well as the Federal Circuit, have attempted to provide structure to subject matter eligibility jurisprudence, but their attempts have often failed to provide the necessary guidance. While this legal issue may seem novel, due to the relatively recent emergence of software technology, courts and legislators have previously grappled with a similar issue in plant breeding. Novel varieties of plants were previously considered not eligible for patent protection. However, in response to the common law of the time, Congress enacted targeted legislation that afforded patent protection to plants. This Note will explore the history of this plant patent legislation, specifically the Plant Patent Act of 1930 and the Plant Variety Act of 1970, and argue that similar legislation is necessary for software.

* J.D., University of Colorado, Class of 2019; B.S., Colorado School of Mines, 2013.

INTRODUCTION

There is a war in the world of software patents. This war rages within splintered factions and encompasses many issues, but perhaps the most contentious is whether improvements in software technologies should be considered patent-eligible subject matter.

Current law considers "any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof" appropriate subject matter for patent protection.[1] Courts have traditionally interpreted this list rather expansively and afforded patent protection to "anything under the sun that is made by man."[2] However, there are limits on this expansive definition. Courts have held basic scientific principles—specifically the laws of nature, physical phenomena, and abstract ideas—not patentable, in order to prevent would-be patent holders from monopolizing "the basic tools of scientific and technological work."[3]

The last of these three exceptions, the abstract idea, is at the core of the debate surrounding software subject matter eligibility.[4] Though the Supreme Court has a long history of applying the abstract idea concept, the Court has never explicitly defined the term, relying instead on earlier precedent.[5] The lack of a definition

[1] 35 U.S.C. § 101 (2012).

[2] Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980).

[3] Gottschalk v. Benson, 409 U.S. 63, 67 (1972).

[4] *See, e.g.*, Alice Corp. v. CLS Bank Int'l, 573 U.S. 208, 212–14 (2014) (finding that the patent at issue, disclosing a scheme for mitigating settlement risk by using a computer system, was drawn to a patent-ineligible abstract idea).

[5] U.S. PATENT & TRADEMARK OFFICE, U.S. DEP'T OF COMMERCE, MANUAL OF PATENT EXAMINING PROCEDURE (MPEP) § 2106.04(a) (9th ed. Rev. 8, Jan. 2018),

has resulted in confusion about the application of the abstract idea exception, particularly in relation to non-physical innovations such as software.[6] In recent decisions the Supreme Court has refused to define the precise boundary that separates patent-eligible software from patent-ineligible software, instead generally holding that "the mere recitation of a generic computer cannot transform a patent-ineligible abstract idea into a patent-eligible invention."[7]

Because the Supreme Court has not answered the basic question of whether all software is patentable, competing arguments have proposed conflicting solutions to the software patent problem and, more generally, have produced dueling scholarship seeking to clarify the underlying purpose of the American patent system.[8]

Although the subject matter eligibility of software patents has been only relatively recently considered by the courts, general subject matter eligibility concerns have previously arisen in other industries. In one such industry, early 20th-century jurisprudence largely considered plants, even those artificially bred, as not suited for patent protection due to being products of nature and therefore not products of man.[9] Echoes of the arguments made both for and against the patentability of plants are emerging once again in the software field. A thorough analysis of these past arguments, along with the resulting solutions, can shed light on modern policy

https://www.uspto.gov/web/offices/pac/mpep/s2106.html#ch2100_d29a1b_13a9e_2dc [https://perma.cc/UL7A-BJ7K] ("The abstract idea exception has deep roots in the Supreme Court's jurisprudence . . . . Despite this long history, the courts have declined to define abstract ideas.").

[6] *See, e.g.*, Mark A. Lemley et al., *Life After* Bilski, 63 STAN. L. REV. 1315, 1317 (2011) ("The patentability of software and business methods has a long and tortured history.").

[7] *Alice*, 537 U.S. at 226 (listing a subset of well-known pieces of computer hardware, like generic processors, memory, and drives, as not providing sufficient structure to widely-known economic principles).

[8] For an example of the viewpoints from these two sides, look to the reactions from the Supreme Court's landmark opinion in *Alice*, which many interpreted as a serious blow to software patents. *Compare* Daniel Nazer, *Happy Birthday Alice: Two Years Busting Bad Software Patents*, ELECTRONIC FRONTIER FOUND. (June 20, 2016), https://www.eff.org/ deeplinks/2016/06/happy-birthday-alice-two-years-busting-bad-software-patents [https://perma.cc/2P8Z-D26A] (arguing that the *Alice* decision has harmed software "trolls," thereby benefitting the software industry at large), *with* Gene Quinn, *A Software Patent Setback:* Alice v. CLS Bank, IPWATCHDOG (Jan. 9, 2015), http://www.ipwatchdog.com/2015/01/09/a-software-patent-setback-alice-v-cls-bank/id=53460/ [https://perma.cc/AY9E-3STP] (providing a critical look at the *Alice* decision and the underlying *Mayo* framework). For a comparison with patent eligibility in Europe, see Dan L. Burk, *The Inventive Concept in* Alice Corp. v. CLS Bank Int'l, 45 INT'L REV. INTELL. PROP. & COMPETITION L. 865 (2014) (discussing the similarities of existing subject matter eligibility requirements in Europe and the decision in *Alice*).

[9] F.K. BEIER ET AL., BIOTECHNOLOGY AND PATENT PROTECTION: AN INTERNATIONAL REVIEW 26 (1985).

proposals.[10]

Section I of this paper will provide the relevant background surrounding plant and software technologies, which is necessary to illustrate the technology-specific issues that directly implicate subject matter eligibility. This historical context will illustrate the similarities between these seemingly disparate technologies, which will, in turn, lay the groundwork for a software-specific patent type largely based on existing plant patent legislation. This section will conclude with a workable definition of a software patent, as the term is prone to many interpretations and an undefined term may cause confusion.

Section II will establish the need for a more permanent solution for software patents, using the Supreme Court's recent decision in *Alice Corp. v. CLS Bank* to center the discussion. In *Alice*, the Supreme Court announced a two-step test to determine subject-matter eligibility: first, whether a patent claim is directed to non-eligible subject matter, such as an abstract idea, and, second, whether that claim as a whole amounts to "significantly more" than the judicial exceptions.[11] Although *Alice* is highly controversial, it has controlled many of the recent decisions surrounding software patents and is therefore vital to any discussion concerning subject matter eligibility.[12] This section will discuss the *Alice* test, along with other similar doctrines, to highlight the need for new software patent legislation.

Section III will analyze the specific statutes in the Plant Patent Act of 1930[13] ("Plant Patent Act") and the Plant Variety Protection Act of 1970[14] ("Plant Variety Act") to show what lessons legislators should take from prior statutes as they work to develop a statutory solution for software patents. Specifically, this section will argue in favor of a statutory solution ensuring that software is patent-eligible while also providing limits in scope to facilitate information dissemination. The specific solution will incorporate tenets from both Acts to create a software patent regime that will seek to simultaneously reduce the number of patent litigation proceedings and mitigate uncertainty surrounding software patent.

[10] *See* discussion *infra* Sections I.A, III.B.

[11] *Alice*, 573 U.S. at 215.

[12] *Where Do We Stand One Year After* Alice?, LAW360 (June 17, 2017, 8:27 PM), https://www.law360.com/articles/668773 [https://perma.cc/UTF9-JFGB] (interviewing leading experts on the state of patent law one year after the *Alice* decision).

[13] 35 U.S.C. §§ 161–164 (2012).

[14] Plant Variety Protection Act of 24 December 1970, Pub. L. No. 91-577, 84 Stat. 1542 (1970) (codified as amended at 7 U.S.C. §§ 2321–2583 (2012)).

I. A BRIEF HISTORY OF PLANTS AND SOFTWARE

When taking a surface-level look at the technology surrounding plants and software, drawing significant parallels may seem difficult. After all, the explosion of code and computer technology is a recent phenomenon, whereas primitive plant breeding dates back to the dawn of civilization—when humans transitioned from hunting and gathering to raising animals and farming.[15]

Although ancient peoples did not fully understand the precise mechanisms of plant genetics, they nevertheless realized that certain, desirable traits could be passed down through the generations by careful breeding.[16] In contrast to ancient practices, modern plant breeders no longer simply wait for new plant varieties to develop.[17] Instead, they take a more proactive approach. Breakthroughs in plant crossing and genetic manipulation have enabled modern breeders to inject useful genes directly into a plant's genetic code.[18]

Software technology, on the other hand, is the set of programmed instructions that controls how computer hardware functions and, in contrast to plant breeding, enjoys a much more recent history.[19] In popular culture, even referring to something as "high-tech" almost always refers to trendy software start-up companies or established software giants.[20]

Against this backdrop, plant and software technologies appear to be entirely unrelated. A plant is a physical object that can be touched, moved, and eaten; whereas software is not physical, but instead exists as a collection of instructions and electrical impulses that are sent to smartphones, computers, or other electronic devices.

Though the differences are evident, both software and plants

[15] Evidence of artificial breeding dates back as early as 870 BCE. Caved depictions of masked Assyrian priests show the process of date palm pollination by hand. *See* MARK D. JANIS ET AL., INTELLECTUAL PROPERTY LAW OF PLANTS 11–12 (2014).

[16] *Id.*

[17] David H. Freedman, *The Truth About Genetically Modified Food*, SCI. AM., (Sept. 1, 2013) https://www.scientificamerican.com/article/the-truth-about-genetically-modifi ed-food/ [https://perma.cc/SXQ6-VKLJ].

[18] *Id.*

[19] Ada Lovelace created the first piece of software in the mid-19th century, though the field did not explode until a century later. *See* John Fuegi & Jo Francis, *Lovelace & Babbage and the Creation of the 1843 'Notes,'* 25 IEEE ANNALS HIST. COMPUTING 16, 19 (2003).

[20] *See, e.g.*, Chris Weller, *The 25 Most High-Tech Cities in the World*, BUS. INSIDER (Aug. 8, 2017, 11:34 AM) http://www.businessinsider.com/the-most-high-tech-cities-in-the-world-2017-8 [https://perma.cc/V9HV-ZF69] (ranking "high-tech" cities largely by the number of software companies within the city limits).

share a common core—data manipulation. With software, developers manipulate the code.[21] With plants, breeders manipulate the genetic code.[22] In each case, successive improvements involve tinkering with a pre-existing product to create something new. In plant breeding, breeders modify existing plants to create new plants that are hardier, have better yield, or are more disease resistant[23] and, in software development, developers control existing hardware to provide a user with a more useful visual, tactile, or auditory feedback.[24]

Data manipulation is the common thread that connects plants and software, but a brief background on the patent history of each further underscores how data manipulation relates to the patent landscape of each technology. Part A of this section will provide a brief history of plant patents, particularly as it relates to the Plant Patent Act and the Plant Variety Act. Part B will examine software patents as they presently exist.

A. Creation of the Plant Patent and Plant Variety Protection

Before the Plant Patent Act, two underlying beliefs generally disqualified plants from receiving standard patent protection.[25] First, "plants, even those artificially bred, were products of nature," which, in the eyes of the Court, made them ineligible for patent protection.[26] Second, because plants are largely visual and variations among similar plants may be in only color or smell, the Court thought it was impossible to adequately describe these features in text-based patents.[27]

These two beliefs first appeared in the doctrinal case *Ex parte Latimer*.[28] In *Latimer*, the applicant attempted to get two separate patents—one for the process of isolating a fiber from the needles of a pine tree and the other for the fiber material itself.[29] The patent

[21] *See, e.g.*, KEN KOCIENDA, CREATIVE SELECTION: INSIDE APPLE'S DESIGN PROCESS DURING THE GOLDEN AGE OF STEVE JOBS (2018).

[22] *See* JOHN PARRINGTON, REDESIGNING LIFE: HOW GENOME EDITING WILL TRANSFORM THE WORLD 1–3 (2016).

[23] *See* Freedman, *supra* note 17.

[24] *Definition of: Software*, PCMAG.COM: ENCYCLOPEDIA, https://www.pcmag.com/encyclopedia/term/51660/software [https://perma.cc/VBC8-542B].

[25] This belief has eroded over the years. Modern jurisprudence currently affords utility patent protection, in addition to plant patent protection, to plants. However, this is a relatively recent development and for much of American patent-history plants were not considered eligible for utility patent protection. *See* Philip Pardey et al., *The Evolving Landscape of Plant Varietal Rights in the United States, 1930–2008*, 31 NATURE BIOTECH. 25, 25 (2013).

[26] Diamond v. Chakrabarty, 447 U.S. 303, 311–12 (1980).

[27] *Id.* at 312.

[28] 1889 Dec. Comm'r Pat. 123.

[29] *Id.* at 125.

examiner granted the process patent, but rejected the fiber material patent because he considered the isolated fibers indistinguishable from those naturally produced by trees or plants.[30] The Commissioner for Patents affirmed the examiner's decision and opined that if the applicant obtained a patent on the fiber, "patents might be obtained upon the trees of the forest and the plants of the earth, which of course would be unreasonable and impossible."[31] Though seemingly innocuous, this piece of dicta rendered the entirety of plants patent ineligible, even those that would not have existed in nature if not for human-directed breeding efforts.[32]

Plant breeders were not pleased by this status quo—they believed themselves to be innovators whose efforts were worthy of patent protection.[33] The easily reproducible nature of plants makes novel plant varieties expensive and time-consuming to produce but, once created, easily and inexpensively duplicated through pollination or grafting.[34] While plant breeders could have created many new varieties of plants, such as roses, fruit trees, and other woody plants, through new innovations in cross-breeding techniques, they feared that without explicit patent protection they would lose their first-mover advantage and genetically identical copies of their innovations would quickly flood the marketplace.[35] In the early 20th century, the industry was experiencing a boom, but, without patent protection, early innovators feared this boom would quickly fizzle.[36]

In response to these fears, plant breeders brought several proposals for a dedicated plant patent before Congress, which eventually culminated in the Plant Patent Act.[37] The Plant Patent Act attempted to strike a compromise between those wanting patent protection for plants and those fearing monopolization.[38] Congress achieved this by protecting plants that reproduce asexually, such as roses or fruit trees, but omitting coverage for plants that reproduce sexually, such as grains or vegetables.[39]

[30] *Id.*

[31] *Id.* at 126.

[32] *See* Cary Fowler, *The Plant Patent Act of 1930: A Sociological History of Its Creation*, 82 J. PAT. & TRADEMARK OFF. SOC'Y 621, 622–25 (2000).

[33] *Id.* at 621 ("[P]roponents of the [Plant Patent Act] argued at the time that the Act [providing plants with patent protection] was warranted because changes in technology had made inventors out of plant breeders.").

[34] JANIS ET AL., *supra* note 15, at 2.

[35] Fowler, *supra* note 32, at 630 ("Once a [plant] variety 'escaped' from a single customer, it could be propagated legally and with impunity by others . . . .").

[36] *Id.* at 626.

[37] *Id.* at 622–40.

[38] *Id.* at 640–41.

[39] Congress made this compromise in response to fears that patenting food-providing plants could result in a monopoly on basic necessities. *Id.* at 635.

Congressional action in the field then stagnated for forty years, until Congress enacted the Plant Variety Act.[40] Congress created the Plant Variety Act to provide a sui generis plant breeders' rights system, including coverage for sexually reproducible plant types.[41] Although the Plant Variety Act is a type of intellectual property protection, it is outside the United States Patent and Trademark Office's (USPTO) administration and instead falls under the United States Department of Agriculture (USDA).[42]

While the Plant Patent Act and the Plant Variety Act share similarities, three major differences distinguish them. First, under the Plant Variety Act, the USDA has certain authority to safeguard the public interest by declaring an otherwise protected plant variety open for public use.[43] Second, the Plant Variety Act includes a "research exemption," which allows researchers to use any protected variety in experimentation without paying royalties.[44] Lastly, the Plant Variety Act allows farmers to save seed from protected varieties and to use in limited situations.[45]

The modern landscape surrounding the intellectual property of plants can be complex, but these two pieces of legislation form the backbone for congressional action in novel plant matter. When determining whether Congress should legislate specifically for software, we can use these two Acts as a guide for future software patent legislation. By looking both at the historical record and the eventual innovation that resulted in plant breeding after these Acts were enacted, Congress can tailor any future legislation in software appropriately—and learn from the lessons of the past.

B. The Rise of Software

In 1968, the USPTO issued the first U.S. software patent to Martin Goetz for a method of sorting without the use of external, dedicated hardware.[46] Since the grant of that patent, there has been a long debate as to whether software should be patentable.[47]

Those who argue against the patentability of software rarely invoke arguments about novelty or innovation and instead rely on

[40] Plant Variety Protection Act of 24 December 1970, Pub. L. No. 91-577, 84 Stat. 1542 (1970) (codified as amended at 7 U.S.C. §§ 2321–2583 (2012)).

[41] JANIS ET AL., *supra* note 15, at 90.

[42] *Plant Variety Protection*, U.S. DEP'T OF AGRIC., https://www.ams.usda.gov/services/plant-variety-protection [https://perma.cc/43P6-6NNX].

[43] 7 U.S.C. § 2404 (2012).

[44] *Id.* § 2544.

[45] *Id.* § 2543.

[46] *Should Patents be Awarded to Software?*, WALL ST. J. (May 12, 2013, 4:03 PM), https://www.wsj.com/articles/SB10001424127887323335404578444683887043510 [https://perma.cc/MYL8-TCBN] [hereinafter *WSJ Patent Debate*].

[47] *See supra* note 8 and accompanying text.

the notion that the patent process functions too slowly to create any actual public benefit.[48] Opponents also argue that the mere nature of software allows patent applicants to claim their innovations in a way that unfairly pushes out competitors.[49] The tech industry is extremely quick moving, so software innovators may reap most of the financial benefits before the USPTO can grant the patent, which can take years.[50] Furthermore, improperly granted patents could cover basic principles, which would prevent innovators from developing groundbreaking improvements and ultimately tie up entire industries in dense patent thickets.[51]

Those in favor of intellectual property protection for software argue that the innovative elements of software are not substantially different from that of physical apparatuses, and therefore software should be protected regardless of physicality.[52] Although proponents concede that the current software patent framework is flawed, they argue that since any piece of software can alternatively be created using hard-wired components, patent eligibility should not rest on the specific construction developers use.[53]

This Note assumes that software should have intellectual property protection in at least some form. Later, in Section III, this Note will discuss the aspects of software that deserve intellectual property protection but will also consider policy changes that will help avoid the problems that could result from overly broad software patents.[54]

C. What is a Software Patent?

Before we ask whether software should be patent-eligible, we need to answer the question "what is a software patent?" While this seems as though it would have a simple answer, describing what a software patent actually *is* is harder than it appears. One reason this question is difficult to answer may be because it incorrectly frames the issue.[55]

According to Goetz, a better question reads: "should an

[48] *Id.*

[49] *WSJ Patent Debate*, *supra* note 46.

[50] *Id.*

[51] Carl Shapiro, *Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard Setting*, *in* 1 INNOVATION POLICY AND THE ECONOMY 119, 121 (Adam B. Jaffe et al. eds., 2001).

[52] *WSJ Patent Debate*, *supra* note 46.

[53] *Id.*

[54] A similar compromise made in the Plant Patent Act. *See supra* note 39 and accompanying text.

[55] *WSJ Patent Debate*, *supra* note 46.

invention that is patentable when described as part of a digital circuit (hardware) be equally patentable when described as stand-alone software?"[56] In other words, if the software used in applications such as facial feature recognition, driverless cars, and computer graphics, would be patentable if described using physical components, should it be equally patentable when created solely as computer software? This new question appropriately considers the fact that software is just another, albeit virtual, way to create a physical hardware system. With enough time, patience, and space, anything that can be coded into a computer can also be created with physical hardware components, such as transistors, resistors, and switches.[57] For the remainder of this Note, the term "software patent" will refer to subject matter that can be built using physical components but is being expressed digitally.

While software patents can presently be used to protect inventions, complications often arise because of their perceived similarity to business method patents—which have spurred criticism in both academic and business communities.[58] Business method patents generally do not define a specific piece of software, but instead simply define "a method of doing business"—a rather tautological, but accurate, definition.[59]

Probably the most famous example of the business method patent is Amazon's "One-Click" patent.[60] The One-Click patent, which covered a user's purchase of an online item with only "one-click" and without entry of address or credit card information, did not identify a single piece of software, but protected the idea itself.[61] Though controversial, the USPTO ultimately upheld the patent as valid.[62]

Much of the distaste underlying software patents comes from a lack of knowledge about the differences between software patents and business methods patents. Although the distinction is not clear cut, there is a clear difference. The former defines a specific way of solving a computer-related problem, whereas the latter more generally involves a computer-implemented solution for a manner of doing business. Nevertheless, patents covering both types of

[56] *Id.*

[57] *See* Ronald D. Williams et al., *Teaching Computer Design Using Virtual Prototyping*, 46 IEEE TRANSACTIONS ON EDUC. 296, 296 (2003).

[58] *See* John R. Allison & Emerson H. Tiller, *The Business Method Patent Myth*, 18 BERKELEY TECH. L.J. 987, 1007 (2003).

[59] *Id.* at 1031.

[60] U.S. Patent No. 5,960,411 (issued Sept. 28, 1999).

[61] *Id.*

[62] Dennis Crouch, *Amazon One-Click Patent Slides Through Reexamination*, PATENTLYO (Mar. 10, 2010), https://patentlyo.com/patent/2010/03/amazon-one-click-patent-slides-through-reexamination.html [https://perma.cc/JU8H-JSGM].

subject matter commonly come before the courts.[63]

Those who lump software and business method patents together often point to a handful of infamous patents, such as the aforementioned One-Click patent. And indeed, they roll software claims reciting digital methods for solving a specific problem into their arguments. Under the current software patent standard, practitioners have repeatedly proven that they cannot easily distinguish between software and business method patents before costly litigation has started. To alleviate this confusion, any proposed software patent legislation must create a presumption in favor of the subject matter eligibility.

II. A NEED FOR STRUCTURE—A WORKABLE MODEL FOR SOFTWARE

Insofar as it would relieve pressure from both courts and parties to litigation, reducing uncertainty should be a priority in patent law. Perhaps the most persistent criticism against software patents is the lack of predictability in both patent prosecution and the subsequent litigation, particularly for issues arising under § 101.[64] Academics have repeatedly found a high correlation between legal uncertainty and a resulting increase in litigation rates.[65] With the amount of ongoing litigation in the technology industry, uncertainty in software patent law drives many of these disputes.[66]

Section 101 was never intended to be a high bar for applicants to pass.[67] David Kappos, the former director of the USPTO, has

[63] *See, e.g.*, Bilski v. Kappos, 561 U.S. 593 (2010) (involving a computer-implemented business method for hedging against the risk of price changes in the energy market); McRo, Inc. v. Namco Bandai Games Am., Inc., 23 F. Supp. 3d 1113 (2013) (involving a software patent for animating lip synchronization).

[64] Jeffrey A. Lefstin et al., *Final Report of the Berkeley Center for Law & Technology Section 101 Workshop: Addressing Patent Eligibility Challenges*, 33 BERKELEY TECH. L.J. 551, 593–94 (2018).

[65] *See* William M. Landes & Richard A. Posner, *Legal Precedent: A Theoretical and Empirical Analysis*, 19 J.L & ECON. 249, 269–70 (1976) (explaining that an increase in litigation would result when the likely outcomes of legal disputes are in question); Peter Siegelman & Joel Waldfogel, *Toward a Taxonomy of Disputes: New Evidence Through the Prism of the Priest/Klein Model*, 28 J. LEGAL STUDS. 101 (1999); *see also* Peter Siegelman & John J. Donohue, *The Selection of Employment Discrimination Disputes for Litigation: Using Business Cycle Effects to Test the Priest/Klein Hypothesis*, 24 J. LEGAL STUDS. 427 (1995).

[66] Keith N. Hylton, *Patent Uncertainty: Toward a Framework with Applications*, 96 B.U. L. REV. 1117, 1125 (2016).

[67] Ryan Davis, *Kappos Calls for Abolition of Section 101 of Patent Act* (Apr. 12, 2016 4:32 PM), https://www.law360.com/articles/783604/kappos-calls-for-abolition-of-section-101-of-patent-act [https://perma.cc/J4P3-5FZN] ("At the time Section 101 was written, those areas of the law were less well-developed, and the patent-eligibility requirement was designed to serve as a 'backstop' to prevent patents on basic concepts.").

used stronger language to describe the current state of patent subject matter eligibility jurisprudence, calling it "a real mess."[68] The extent of the confusion surrounding subject matter eligibility should compel Congress to reexamine the current state of the law and consider fixing the problem through legislation.

By explicitly making software inventions patent-eligible, Congress will ensure that courts no longer need to navigate the labyrinth of subject matter eligibility jurisprudence and can instead focus on the more settled areas of patent law—namely 35 U.S.C. §§ 102[69], 103[70], and 112[71], which cover the principles of novelty, obviousness, and clarity of the written description, respectively.

A statutory solution is the most effective way to accommodate software patents in our legal system. The remainder of this semester will examine the major shortcomings of current software patent laws, which often result in the high cost of software patent litigation and prosecution, and recommend a legislative solution, which will be tailored to address software-specific problems and thereby increase software patent quality.

In the literature, the discussions about this issue are lengthy and robust.[72] As such, rather than retread the entire debate, Part A of this section will instead briefly discuss the current jurisprudence surrounding subject matter eligibility and the need for a statutory solution for software. Part B will discuss whether different types of technologies should be treated differently under the patent system or whether a technology-blind system should be preferred.

A. Reasons for a Statutory Solution

Patent litigation has exploded in recent years.[73] Scholars have attributed a large portion of ongoing litigation to software[74] and

[68] As one author puts it, "[t]he Supreme Court is both obsessed with the law governing eligible subject matter and unable to identify a workable standard." David O. Taylor, *Amending Patent Eligibility*, 50 U.C. DAVIS L. REV. 2149, 2154 (2017); *see also id.*

[69] 35 U.S.C. § 102 (2012).

[70] *Id.* § 103.

[71] *Id.* § 112.

[72] *See* Lefstin et al., *supra* note 64.

[73] *See* Dennis Crouch, *US Patent Litigation New Filings by Year*, PATENTLYO (Dec. 19, 2016), https://patentlyo.com/patent/2016/12/patent-litigation-filings.html [https://perma.cc/D49N-TBB7]. *But see* David Pridham, *The Patent Litigation Lie*, FORBES (Apr. 13, 2017, 12:24 PM), https://www.forbes.com/sites/davidpridham/2017/04/13/the-patent-litigation-lie/2/ [https: //perma.cc/ZBD2-JJYZ] (arguing that the apparent rise in patent litigation is due to specific lobbying efforts from Big Tech rather than from all patent holders equally).

[74] Brian J. Love, *An Empirical Study of Patent Litigation Timing: Could a Patent Term Reduction Decimate Trolls Without Harming Innovators?*, 161 U. PA. L. REV. 1309,

many high-profile decisions have revolved around software technologies.[75] When looking at the number of patent suits filed, sharp increases from two time periods stand out—the first from 2011 to 2012 and the second in 2015.[76] From 2011 to 2012 the number of patent suits increased after the Supreme Court issued its decisions in *Bilski v. Kappos*[77] and *Mayo Collaborative Servs. v. Prometheus Labs.*,[78] which the Court handed down in 2010 and 2012 respectively. The second bump came after *Alice Corp. v. CLS Bank Int'l*,[79] which the Court decided in 2014. While it is unclear exactly how much litigation can be attributed to any specific case, these three cases appear to have at least contributed to the increase.

Bilski was the first modern Supreme Court decision to seriously consider which rules the Court should adopt to determine whether a particular piece of software is patent-eligible.[80] In *Bilski*, the Supreme Court analyzed the validity of a patent seeking to explain "how buyers and sellers of commodities in the energy market can protect, or hedge, against the risk of price changes."[81] The Court invalidated the claims at issue for merely claiming "the abstract idea of hedging risk in the energy market."[82]

The *Bilski* Court may have intended to reduce the uncertainty in software patentability—indeed, in the majority opinion, Justice Kennedy referenced numerous amicus briefs that argued, "[T]he machine-or-transformation test [as the sole criterion for determining subject matter eligibility] would create uncertainty as to the patentability of software."[83] In reality, however, the opinion did little to mitigate uncertainty. *Bilski* instead opened a Pandora's Box that the Court has yet to quiet. By giving parties the ability to attack subject matter eligibility, the Supreme Court allowed those seeking to invalidate patents near boundless creativity for their claims.

In retrospect, that the Supreme Court's decision would increase uncertainty seems like a likely result. Because courts could only use the machine-or-transformation test prior to the decision, parties knew the precise standard that a court would use

1344 (2013).

[75] *See, e.g.*, Alice Corp. v. CLS Bank Int'l, 573 U.S. 208 (2014).

[76] Crouch, *supra* note 73.

[77] 561 U.S. 593 (2010).

[78] 566 U.S. 66 (2012).

[79] 134 S. Ct. 2347 (2014).

[80] *Bilski*, 561 U.S. at 605–06 (rejecting the Federal Circuit's strict adherence to the machine-or-transformation test and opening the door for additional modes of analyses).

[81] *Id.* at 599.

[82] *Id.* at 612.

[83] *Id.* at 605.

to treat software claims. Under the *Bilski* approach, however, the number of available tests greatly increased, providing more tools for litigators to use in a much greater number of litigation strategies.

The tragedy of *Bilski* is not that the Supreme Court critiqued a strict reliance on the machine-or-transformation test—the test is ill-suited for treating software claims[84]—but rather that the Court failed to introduce an alternative way to determine what actually constitutes a patentable process.[85] As the Court did not settle the subject matter eligibility issue, it soon reappeared before the Court in *Mayo*.[86]

Patent practitioners have described *Mayo* as "the root of all the problems facing the industry relative to patent eligibility" and "probably the worst, most wrongly decided case by the Supreme Court in the patent field ever."[87] *Mayo* involved patent claims covering a process for determining the appropriate amount of thiopurine drugs that should be given to patients with autoimmune diseases.[88] The unanimous Court overturned the Federal Circuit and held the claims invalid, reasoning that the patent simply applied a recitation of a natural law.[89] In doing so, the Supreme Court imposed a new "inventive application" requirement, which set a higher bar for patentability than the prior "useful application of a scientific discovery" standard.[90] Industry reactions to the decision were mixed.[91] Stakeholders either welcomed the stricter standard with hopes that it would curtail future litigation or decried the loss of innovative protection.[92]

What is particularly perplexing about *Mayo* is the Court's reliance on § 101, since it appears that invalidation under § 102 (novelty) or § 103 (non-obviousness) would have been much more appropriate.[93] The dicta in *Mayo* suggests that the Supreme Court is apprehensive about invalidating patents based on prior art; it

[84] *See* Lemley et al., *supra* note 6.

[85] *Bilski*, 561 U.S at 612 ("The Court, therefore, need not define further what constituted a patentable process.") (internal quotations omitted).

[86] Mayo Collaborative Servs. v. Prometheus Labs., 566 U.S. 66 (2012).

[87] Gene Quinn, Mayo v. Prometheus*: A Lawless Decision by an Omnipotent Court Wreaking Havoc on Patents*, IPWATCHDOG (Jan. 23, 2017), http://www.ipwatchdog.com/2017/01/23/mayo-v-prometheus-lawless-decision-wreaking-havoc-patents/id=77438/ [https://perma.cc/FGZ9-UWHN].

[88] *Mayo*, 566 U.S. at 72.

[89] *Id.* at 77–78.

[90] *See id.* at 81.

[91] Lefstin et al., *supra* note 64, at 554.

[92] *Id.*

[93] *See Mayo*, 566 U.S. at 73 ("We find . . . the steps in the claimed processes . . . involve well-understood, routine, conventional activity previously engaged in by researchers in the field.").

instead prefers making broad statements about patentability principles, rather than tailoring them to the specific technology or patent subject matter at issue. Although such a strategy could provide deference to the technology experts, the realistic outcome is that it muddies the patent landscape.

Alice, the most recent case that has been thrown into the patent eligibility pit, is primarily built on *Mayo*'s inventive application limitation and solidified a two-step analysis for determining whether an innovation is tied to patent-eligible subject matter.[94] This two-step analysis, colloquially referred to as the *Alice-Mayo* framework, first asks whether a claim is "directed to a law of nature, a natural phenomenon, or an abstract idea"—the judicially recognized exceptions.[95] If the claim falls under one of these three categories, it asks whether "the claim recites additional elements that amount to significantly more than the judicial exception."[96] Under this framework, the Federal Circuit has held claims in fifty-nine cases to be patent ineligible while claims in only nine cases have survived a § 101 attack.[97]

1. Quality, Quality, Quality—Creating a Presumption of Validity for Software

A common criticism surrounding software patents, and utility patents in general, is that many are low-quality.[98] This criticism had become so pervasive in the patent landscape that patent quality became a major policy agenda item in former-USPTO director Michelle Lee's administrative goals.[99]

Although patent quality is a subjective concept, proper invention disclosure and lasting patent validity can help evaluate a specific patent's quality. This part will focus more specifically on whether creating a strict patent eligibility standard through invention disclosure perverts the patent scheme and incentivizes practitioners to hide the core innovation in their patent applications.

[94] Alice Corp. v. CLS Bank Int'l, 573 U.S. 208 (2014).

[95] 2014 Interim Guidance on Patent Subject Matter Eligibility, 79 Fed. Reg. 74,618, 74,621 (Dec. 16, 2014).

[96] *Id.*

[97] *Chart of Subject Matter Eligibility Court Decisions*, U.S. PAT. & TRADEMARK OFF., https://www.uspto.gov/sites/default/files/documents/ieg-sme_crt_dec.xlsx [https://perma.cc/4CM4-Y89A] (last updated Mar. 16, 2018).

[98] *See, e.g.*, Arti K. Rai, *Improving (Software) Patent Quality Through the Administrative Process*, 51 HOUS. L. REV. 503 (2013).

[99] Michelle K. Lee, Dir. U.S. Patent & Trademark Office, Keynote Address at the Patent Quality Conference (Dec. 13, 2016), https://www.uspto.gov/about-us/news-updates/remarks-director-michelle-k-lee-patent-quality-conference-keynote [https://perma.cc/79T6-FZB3].

Before the *Bilski* decision, patent examiners rarely used § 101 as a basis for a rejecting a patent application.[100] Indeed, § 101 accounted for just 8.5 percent of all rejections.[101] Post-*Alice*, this number ballooned to 12.2% and USPTO groups dealing entirely with software approached a 90% § 101 rejection rate.[102] When faced with a § 101 rejection, an applicant must either challenge the merits of the rejection or amend the claims to overcome the rejection.[103] In a post-*Alice* world, this means downplaying the software aspects in a patent application, which tends to increase uncertainty as applicants often substitute the technical words that clearly describe the product with convoluted jargon.

If software patents had a presumption of validity under § 101, applicants would not need to hide the ball and pretend that an innovation is unrelated to software. Rather, the applicants could shift their attention to drafting the subject matter related to their innovation, which provides more effective notice to competitors and other inventors.

Under the current patent regime, the pendulum sways away from software patents. Few examiners delve deeply into the rationale of the Court and instead persist with their own, personal status quo. A congressional statute removing the subject matter requirement would disrupt this status quo and allow applicants the freedom to describe their innovation in more directed terms.

2. Preventing the Blockbuster Patent Dispute

Since *Alice*, patent eligibility cases have flooded the courts.[104] These cases would likely not completely disappear if the Supreme Court interpreted § 101 differently, but a more favorable interpretation would be more prone to actually force litigants to find invalidating prior art.

In *Mayo*, the Supreme Court declared a process to be tied to patent-ineligible subject matter.[105] As part of the analysis, the Court claimed that "the steps in the claimed processes . . . involve well-understood, routine, conventional activity previously engaged

[100] James Cosgrove, *§ 101 Rejections in the Post-*Alice *Era*, IPWATCHDOG (Mar. 7, 2017), http://www.ipwatchdog.com/2017/03/07/101-rejections-post-alice-era/id=78635/ [https://perma.cc/8ATV-CNET].

[101] *Id.*

[102] *Id.*

[103] U.S. PATENT & TRADEMARK OFFICE, U.S. DEP'T OF COMMERCE, MANUAL OF PATENT EXAMINING PROCEDURE (MPEP) § 2666 (9th ed. Rev. 8, Jan. 2018), https://www.uspto.gov/web/offices/pac/mpep/s2666.html [https://perma.cc/V5CC-SWW4].

[104] *See* discussion *supra* Section II.A.

[105] Mayo Collaborative Servs. v. Prometheus Labs., Inc., 566 U.S. 66, 77 (2012).

in by researchers in the field."[106] For an opinion that so heavily relies on words like "routine" and "conventional," it is noteworthy that the Court did not use the novelty and non-obviousness provisions of the U.S. Code—instead relying on subject matter eligibility. Rather than rely on opaque doctrine, a better option would be requiring the party challenging the patent to produce an invalidating piece of prior art to demonstrate that the patented work already exists in the market.

Under the current legal framework, a party need not produce prior art nor prove that the innovation already exists on the market, so long as they can bring a § 101 challenge. That is, even if a software invention is useful and innovative, a party could challenge its patentability using the ambiguous rules of *Alice* and burden the patent holder with expensive litigation. If the Court allowed the doctrines of novelty and non-obviousness to control, uncertainty and litigation costs should both decrease.

B. Technology Concerns – Should Software Be Treated Differently?

Commentators Dan Burk and Mark Lemley have suggested that courts often apply principles of patent law differently to different areas of technology, particularly in the biotechnology and software sectors.[107] They have noted that while patent statutes generally do not distinguish between different technology types, courts' applications of these statutes varies wildly across industries.[108] For example, in the software industry, Burk and Lemley have found that patents must overcome high obviousness standards. Once the patent has overcome this standard, applicants must disclose "virtually nothing about the detailed workings of their invention."[109]

In light of the courts' inability to provide consistent guidance,[110] Congress should sever this sort of deference from judicial discretion and instead promulgate a legislative framework that more explicitly spells out the parameters for software patent eligibility.

[106] *Id.* at 73.

[107] Dan L. Burk & Mark A. Lemley, *Is Patent Law Technology-Specific?*, 17 BERKELEY TECH. L.J. 1115 (2002).

[108] *Id.* at 1156.

[109] *Id.* at 1173.

[110] *See* discussion *supra* Section II.

III. THE MODEL FOR REFORM—PLANT PATENT ACT AND PLANT VARIETY ACT

Although the technologies that drive innovations in plant breeding and software development are different, the arguments that call into question their patentability are often similar. This section will look to legislation involving the patentability of plants, underscore the similarities between plant and software patents, and present the specific statutory remedies that Congress should enact for software patents. To be sure, the issues surrounding subject matter eligibility are complicated, but there is no need to start from scratch—together, both the Plant Patent Act and the Plant Variety Act provide a template for Congress to enact legislation to handle software patent laws.

Part A of this section will briefly discuss how Congress structured the laws surrounding plant patents. Part B will provide an analysis of how Congress could apply this structure to software. Finally, Part C will consider how Congress could protect software outside of patent law—with a particular focus on the Plant Variety Act. Specifically, it will look to whether Congress could apply certain provisions from this Act to software.[111]

A. The Statutory Landscape of Plants

The Plant Patent Act contains only four short sections—§§ 161 to 164.[112] Of these four sections, §§ 161 and 162 are most relevant here because they have reframed the obstacles that had previously prevented courts from considering plants to be patentable objects.[113]

Prior to the Plant Patent Act, two perceived obstacles prevented courts from considering plants to be patentable subject matter. In *Chakrabarty*, the Supreme Court discussed these two obstacles.[114] The first was the common belief that plants were

[111] When discussing the patentability of plants, I will focus on the history of such patents within the United States. A curious reader may want to research such topics through an international lens. *See, e.g.*, F.K. BEIER ET AL., *supra* note 9, at 67 (discussing how plants have historically been treated under European patent law, specifically looking at Germany in the 1930s); Martin A. Girsberger, *The Protection of Traditional Plant Genetic Resources for Food and Agriculture and the Related Know-How by Intellectual Property Rights in International Law: The Current Legal Environment*, 1 J. WORLD INTELL. PROP. 1017 (1998) (discussing the legal landscape surrounding plant genetic resources for food and agriculture as it relates to international trade).

[112] Plant Patent Act of 1930, 35 U.S.C. §§ 161–164 (2012).

[113] Though not discussed in this Note, Section 163 discusses the grant of the plant patent, including the right to exclude others from using the patented plant, and Section 164 allows the President to request certain information from the Secretary of Agriculture on behalf of the Director of the Patent Office.

[114] Diamond v. Chakrabarty, 447 U.S. 303, 311–12 (1980).

products of nature and therefore fell squarely within the public sphere.[115] Congress explicitly rebutted and statutorily overturned this belief in § 161 of the Plant Patent Act, which reads:

> Whoever invents or discovers and asexually reproduces any distinct and new variety of plant, including cultivated sports, mutants, hybrids, and newly found seedlings, other than a tuber propagated plant or a plant found in an uncultivated state, may obtain a patent therefor, subject to the conditions and requirements of this title.[116]

The second obstacle summarized in *Chakrabarty* was the belief that patents could not adequately describe plants in text, as "new plants may differ from old only in color or perfume, [which made] differentiation by written description . . . impossible."[117] Again, Congress unambiguously overturned this doctrine in § 162, which reads:

> No plant patent shall be declared invalid for noncompliance with section 112 if the description is as complete as is reasonably possible.[118]

In using the above language, Congress acted on its belief that the work of a plant breeder was both innovative in nature and worthy of patent protection. Thomas Edison succinctly wrote why the patent system should be available to plant breeders, sending a letter to Congress which said, in part:

> Nothing that Congress could do to help farming would be of greater value and permanence than to give to the plant breeder the same status as the mechanical and chemical inventors now have through the patent law. There are but few plant breeders. This [Act] will, I feel sure, give us many Burbanks [a prolific plant breeder of the time].[119]

The recent increase in plant patent applications certainly appears to have vindicated Edison and the increase in these filings, which had risen to a maximum in recent years, tends to show that a renaissance of plant innovation occurred as a direct result.[120]

[115] *Id.*
[116] 35 U.S.C. § 161 (2012).
[117] *Chakrabarty*, 447 U.S. at 312.
[118] 35 U.S.C. § 162 (2012).
[119] 72 CONG. REC. 8392 (1930).
[120] Pardey et al., *supra* note 25, at 26.

B. From Plants to Software

Essentially, the Plant Patent Act ensured that certain types of plants receive patent protection, in direct contrast to court jurisprudence that governed when Congress enacted the law.[121] In this way, inventors and cultivators of new plants of this type could apply for a plant patent and avoid costly court battles as to whether plants should be patentable. Rather, the test for validity rested simply on novelty[122] and whether such a plant was known to have existed before the alleged invention.

Though the number of plant patent application filings is smaller than the number of software application fillings, there is still an appreciable lack of litigation in the space, particularly around questions of eligibility. In fact, there have only been a few precedential cases relating directly to plants since the enactment of the Plant Patent Act, all of which focused on substantive issues, rather than rehashing § 101 concerns.[123]

The statutory framework in the Plant Patent Act prevented, or at least dissuaded, parties from raising general, overreaching arguments concerning general written description principles. Software should be treated similarly to avoid similar distractions. Software, after all, is not a physical object that we can hold in our hands, like a printer or battery, and may require language entirely different from conventional patents.

A provision similar to 35 U.S.C. § 161 should exist in any future software legislation so parties focus on the specific patents at issue. A possible software statute in this vein may read as follows:

> Whoever invents any distinct and new computer code,

[121] *Ex parte* Latimer, 1889 Dec. Comm'r Pat. 123.

[122] And later non-obviousness, as codified in the Patent Act of 1952.

[123] *See* J.E.M. Ag Supply, Inc. v. Pioneer Hi-Bred, Inc., 534 U.S. 124 (2001) (agreeing with *Hibberd* that plants are eligible for utility patents); Asgrow Seed Co. v. Winterboer, 513 U.S. 179 (1995) (clarifying the manner in which a farmer can save seed under the Plant Variety Protection Act); Monsanto Co. v. McFarling, 488 F.3d 973 (Fed. Cir. 2007) (finding that patent exhaustion doesn't exist for second generation seeds not sold by patent holder); Imazio Nursery, Inc. v. Dania Greenhouse, 69 F.2d 1560 (Fed. Cir. 1995) (finding that actual evidence of copying is required to show infringement of a plant patent); Pioneer Hi-Bred Int'l, Inc. v. Holden Foundation Seed Inc., 35 F.3d 1226 (8th Cir. 1994) (holding that plants can also be protected under trade secret law); Yoder Bros., Inc. v. California-Florida Plant Corp., 537 F.2d 1347 (5th Cir. 1976) (clarifying the concept of novelty as it refers to plants, specifically requiring that a new plant has to be one that literally had not existed prior); Cole Nursery Co. v. Youdoth Perennial Gardens, Inc., 17 F. Supp. 159 (N.D. Ohio, 1936) (invalidating a burberry plant patent due to prior public use); *Ex parte* Hibberd, 227 USPQ 433 (Bd. Pat. App. & Interf. 1990) (holding that seeds and plants may be protected by utility patents in addition to plant-specific protections).

> including machine code, source code, and object code, other than for merely digitizing well-known principles, may obtain a patent therefor, subject to the conditions and requirements of this title.

Although the above proposal is not a perfect representation of what a software statute should look like, it may alleviate current judicial problems. If Congress decides to enact a similar proposal, it should properly balance the concerns raised in *Bilski* and *Alice* with the concerns of software patent owners.

C. *Intellectual Property Protection Outside the Patent System*

Up until this point, this Note has primarily discussed solutions to the subject matter eligibility issue using a patent-specific lens. While a patent-specific solution for statutorily ensuring that software is patent-eligible is one way to protect software innovation, there have been other suggestions to protect and encourage software innovation through non-patent incentives.[124] One proposal is for the government to provide increased grants, tax incentives, or prizes to researchers who are involved with creating cutting-edge software.[125] According to this proposal, prizes are the optimal way to foster software innovation since a clear, market-provided goal can incentivize innovators in a way that patents cannot.[126] Since patents do not necessarily convey market power, allowing the market to preemptively set what breakthroughs are valuable will prevent the USPTO-driven roulette of approving many patents and seeing what sticks.

In this arena we can also look to congressional treatment of plants, specifically through the Plant Variety Act, to consider how non-patent incentives have performed. The Plant Variety Act was created to extend intellectual property protection for plants that reproduced "asexually or sexually."[127] The policy behind this additional protection is "to afford adequate encouragement for research, and for marketing when appropriate, to yield for the public the benefits of new varieties."[128] While the policy rationale

[124] Lisa Larrimore Ouellette, *Patentable Subject Matter and Nonpatent Innovation Incentives*, 5 U.C. IRVINE L. REV. 1115, 1137–38 (2015).

[125] *Id.* at 1138–41.

[126] *Id.* at 1140.

[127] *Patent Law Revision Part 2: Hearings on S. 2, S. 1042, S. 1377, S. 1691, S. 2164, and S. 2597 before the Subcomm. On Patents, Trademarks & Copyrights of the Senate Comm. on the Judiciary*, 90th Cong. 637, 639 (1968).

[128] 7 U.S.C. § 2581 (2012); *see also Plant Variety Protection Act: Hearing on S. 3070 before the Subcomm. On Agric. Res. & Gen. Legis. Of the Senate Comm. on Agric. & Forestry*, 91st Cong. 47 (1970) (statement of Hon. Jack Miller, U.S. Senator, Iowa) ("The bill under your consideration is designed to encourage the development of new varieties

behind the Act certainly aligns with the policy rationale for utility patents in general, Congress decided to form the Plant Variety Act outside the patent system in order to tailor policy to better meet the technology-specific concerns of those in the agricultural industry.[129]

There are many notable differences between utility patents and the Plant Variety Act.[130] This section will focus on the limitations of plant variety protection rights and consider whether Congress could afford analogous limitations to software in any future legislation. Of these limitations on rights, the three most disputed exemptions are as follows: (1) the noncommercial use exemption;[131] (2) the plant breeder's exemption;[132] and (3) the saved-seed exemption.[133]

1. Applying the Noncommercial Use Exemption to Software

Those familiar with utility patents may remember the experimental use exception that a defendant can use as a defense to patent infringement.[134] The Federal Circuit's current jurisprudence treats the experimental use defense very narrowly—applying the exception only when there are no monetary interests. In *Madey*, the court found that Duke University's experimental use of patented technology did not constitute experimental use because the University had "legitimate business objectives" in using patented technology.[135] The experimental use defense described in *Madey* is not a creature of statute, but rather a common law doctrine created by the courts.

Commentators have criticized the narrow treatment of the experimental use exception but, in the absence of a congressional statute, it appears that this narrow treatment will continue.[136]

of sexually reproduced plants by providing protection for those who breed and develop them, thus promoting the growth and well-being of agriculture.").

[129] JANIS ET AL., *supra* note 15, at 92–93.

[130] For example, utility patents have a term of 20 years while under the Plant Variety Act the protection length depends on the specific plant. *See* 7 U.S.C. § 2483.

[131] *Id.* § 2541(e).

[132] *Id.* § 2544.

[133] *Id.* § 2543.

[134] *See, e.g.*, Madey v. Duke, 307 F.3d 1351, 1361 (2002) ("[T]he experimental use defense persists albeit in [a] very narrow form.").

[135] Specifically looking at Duke's interest in attracting student talent, keeping a reputation as a cutting-edge research institution, and holding an aggressive patent licensing program. *Id.* at 1362.

[136] *See, e.g.*, Janice M. Mueller, *The Evanescent Experimental Use Exemption from United States Patent Infringement Liability: Implications for University and Nonprofit Research and Development*, 56 BAYLOR L. REV. 917 (2004) (examining the emasculation of the experimental use exemption); Janice M. Mueller, *No "Dilettante Affair": Rethinking the Experimental Use Exception to Patent Infringement for Biomedical*

Therefore, a litigating party may be unsure about the scope of the defense or even its mere existence.

In contrast, the Plant Variety Act includes a provision that "[i]t shall not be an infringement of the rights of the owner of a variety to perform any act done privately and for noncommercial purposes."[137] There has been no litigation requiring the use of this provision, but, should such litigation arise, the noncommercial use exemption would provide an alleged infringer the ability to base their defense on the language of the statute. Additionally, a breeder who wishes to privately use a plant registered by another can rely on this exemption to learn about the technical mechanics of the novel variety and to increase his own understanding, thereby likely increasing the public's knowledge as well.

For software patents, a common strategy of patent-holders alleging patent infringement is to threaten a lawsuit based on dubious grounds.[138] If a party is on the receiving end of a cease and desist letter, reliance on a narrow experimental use exemption does not seem to provide any actual protection. Courts may simply look at the alleged infringer's general business model and find some tenuous connection between a noncommercial use and a commercial strategy. After all, even a nonprofit-producing business may make internal experiments for the purpose of hopefully leading to profit-producing endeavors.

A statutory solution allowing the private, noncommercial use of any purchased software would provide other software developers and programmers the freedom to learn from existing technology without worrying about the threat of a lawsuit. So much of modern software exists because of incremental improvements and "hacking" previous iterations. The industry is built on this concept and a realization of this is necessary for any intellectual property laws focusing on fostering software innovation.

2. Applying the Plant Breeder's Exemption to Software

The plant breeder's exemption in the Plant Variety Act, though similar to the noncommercial use exemption, covers a slightly different scope. Specifically, the plant breeder's exemption ensures that "[t]he use and reproduction of a protected variety for plant

Research Tools, 76 WASH. L. REV. 1 (2001) (proposing alternatives to the narrow formulation of the experimental use doctrine); Arti Kaur Rai, *Regulating Scientific Research: Intellectual Property Rights and the Norms of Science*, 94 NW. U. L. REV. 77 (1999) (describing the debate in patent treatment of basic scientific research).

[137] 7 U.S.C. § 2541(e).

[138] *See* Scott Joslove, *Patent Trolls Threaten Small Businesses*, HILL (Dec. 5, 2013, 2:00 PM), http://thehill.com/blogs/congress-blog/economy-budget/192096-patent-trolls-threaten-small-businesses [https://perma.cc/D5KA-WYMC].

breeding or other bona fide research shall not constitute an infringement."[139]

Though the noncommercial use exemption arguably protects certain forms of reproduction, the plant breeder's exemption directly protects an aspect of plant variety innovation that Congress deemed vital for future growth. And, as the plant breeder's exemption is distinct from the noncommercial use exemption, Congress is allowing the use of a breeding program using protected varieties to developing new varieties for commercial gain. Congress seemingly designed this exemption to act in conjunction with the noncommercial use exemption, specifically affording researchers the opportunity to develop new innovations cumulatively—an implementation of Isaac Newton's famous quote "[i]f I have seen further it is by standing on the sholders [*sic*] of Giants."[140]

Similarly to plants, innovation in software is built on prior innovations and cooperation.[141] When new software is developed, it rarely comes from nowhere. Rather, incremental steps are necessary for the industry to flourish.

It may, therefore, be beneficial for software patents to include an analog to the plant breeder's exemption. Congress could design such an exemption to allow startups and tech giants the free use of any non-original, or copied, software for experimental purposes. Allowing software development companies this opportunity would allow competitors to use each other's products and build upon them in new and unique ways.

But, this argument fails to consider the nature of the software industry. Though both plants and software are built on a spirit of collaboration, there is a key difference between them. Plant breeding takes a long time and by the time a competitor reaps the benefits of a breeding program the original filer would have had ample time to take advantage of his product. Software copying, on the other hand, hardly takes any time at all and a competitor could copy a novel program almost as soon as the novel program is created. The industry is a quick-moving one where momentous innovations seemingly occur every day. Regardless, the pace at which software is evolving should not hinder the idea of plant breeder's exemption analog and such an idea deserves scrutiny and

[139] 7 U.S.C. § 2544.

[140] Letter from Isaac Newton to Robert Hooke (Feb. 5, 1676), *in* 1 THE CORRESPONDENCE OF ISAAC NEWTON: 1661–1675, at 416 (H.W. Turnbull ed., 1959).

[141] Jonathan Zittrain, *Don't Let Software Patents Stop Us Standing on the Shoulders of Giants*, GUARDIAN (Apr. 18, 2012, 2:30 PM), https://www.theguardian.com/commentisfree/2012/apr/18/software-patents-shoulders-of-giants [https://perma.cc/TE4Y-EH9W].

consideration.

3. Applying the Saved-Seed Exemption to Software

The saved-seed exemption in the Plant Variety Act offers perhaps the most interesting ideas about how once-sold software should be treated. As it relates to plants, the saved-seed exemption allows "a person to save seed produced by the person from seed obtained, or descended from seed obtained, by authority of the owner of the variety for seeding purposes."[142] The sales of such seeds are allowed except for "reproductive purposes."[143]

The software equivalent of the saved-seed exemption would be something akin to saving programs or copies of programs. In the copyright statutes, it is not an infringement to copy a computer program if either (a) the new copy is created as an essential step with the successful utilization of a computer program, or (b) if the copies are made for archival purposes only.[144] While this provision is not directly related to patent law, it does affect computer programs and Congress should, in the event of unified software legislation, incorporate it into any intellectual property statute.

The copyright infringement statute allows copying in certain circumstances but appears to only protect copying in limited situations. A broader saved-seed analog, allowing a software developer the freedom not only to copy but to build upon and make improvements to existing forms of software, would benefit the field at large and result in new collaborative efforts.

CONCLUSION

Software patent law is becoming muddled. The long-used principles that have controlled American patent law for centuries are well suited for inventions that exist in a physical capacity but have begun to crack under the relatively recent onslaught of software innovations. Congress already faced a similar issue with plant breeding, as the industry developed and as breeders began to create novel varieties. In response to those issues almost one-hundred years ago, Congress laid groundwork for how plants should be treated—by providing a statutory solution—and blockbuster patent litigation disputes failed to materialize en masse.

Congress should consider a similar statutory solution for software as well. The above analysis provided areas where we can

[142] 7 U.S.C. § 2543.

[143] *Id.*

[144] *Id.* § 117(a).

learn from plant patent treatment and how Congress could apply principles that worked for plants to software. The courts' development of software law comes from people who rarely have any background or experience in any kind of technology and Congress is in a much better position to confer with experts to lay the groundwork for such legislation.