DEBUGGING SOFTWARE PATENTS AFTER ALICE

Jonathan Stroud* & Derek M. Kim**

I. INTRODUCTION ........................................................................................................ 178

II. SOFTWARE PATENT HISTORY ........................................................................... 182
   A. Prior to the Supreme Court’s Alice Decision ................................................. 182
      1. The Supreme Court’s Alice Decision ......................................................... 187
      2. Federal Circuit Decisions Following Alice .............................................. 191

III. CURRENT STATE OF THE SOFTWARE BUSINESS .................................. 199
   A. Problems in the Software Industry .............................................................. 199
   B. Software Patent Thickets .......................................................................... 204

IV. A ROADMAP FOR THE USPTO ................................................................. 208
    A. Patent Claim Value and Solutions to Low-Quality Patent Claims ............. 208
    B. Second-Pair-of-Eyes .............................................................................. 210
    C. Setting Fees and a Call for Consistent Guidance ........................................ 212
    D. Request for Continued Examination Practice .......................................... 214

V. CONCLUSION ....................................................................................................... 215

APPENDIX ............................................................................................................. 217

B. SOFTWARE PATENTS CHALLENGED UNDER 35 U.S.C. § 101 .................. 218
C. SECTION 101 REJECTION RATE POST-ALICE BY ART UNIT .............. 219
D. PERCENTAGES OF ISSUED PATENTS BY NUMBER OF RCEs ............... 220

* Adjunct Professor, American University, Washington College of Law; Chief Patent Counsel, Unified Patents Inc.; J.D., American University, Washington College of Law; M.A., University of Southern California; B.S., Tulane University. Thanks to Derek for his diligent, thoughtful, and insightful work, and to Catherine for help building something.

** J.D. Candidate, 2018, American University, Washington College of Law. I would like to thank Professor Jonas Anderson for his helpful comments on this Article. I also greatly appreciate and thank Jonathan Stroud for everything as co-author of this Article.
I. INTRODUCTION

When the Framers of the Constitution drafted the Intellectual Property Clause, they sought to promote science and technological advancement by prescribing limited monopolies in recognition of the hard work of inventors, in exchange for public disclosure and dissemination of their inventions for the public good.\(^1\) The discovery of algorithmic programming of electronic devices—by Lord Byron’s daughter Ada Lovelace, no less\(^2\)—and the subsequent developments in computing has long complicated matters. These modern marvels have rewritten the rules of business, but their abstract natures have to date ill-fitted the preexisting forms and implementations of intellectual property protection.

Software is patentable in the United States but has been the subject of much debate.\(^3\) In part because software is difficult to describe in traditional patent claims (due in part to lack of any requirement to provide enabled computer code when applying for patent protection), the patent office has struggled with the contours of software patentability, while courts have struggled with interpreting software claims and applying them to actual technologies in infringement actions.\(^4\)

This difficulty in defining disclosure leads to broad, ambiguous claims that do not give clear notice of either the claimed invention or the patentee’s

---

1. U.S. CONST. art. I, § 8, cl. 8 (authorizing Congress “[t]o promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries”).


Such claims often appear in otherwise legitimate patents and confuse and frustrate straightforward licensing and product development. The resulting “software patent thicket” leads to portfolios with hundreds, if not thousands, of claims protecting incremental software. To be clear, this is not good for practicing patent owners, either—according to economists (as cited by the United States Government Accountability Office (GAO)), “the less clear the claim boundaries are, the more likely that others will infringe the patent or will continue to infringe when confronted by the patent owner.” Indeed, when entities acquire swaths of these ambiguous patents, they can broadly restrict other players from entry into the marketplace and can use these patents to extract exorbitant fees from innovators far beyond any compensation needed to encourage innovation. Then, these entities can restrict entry through patent infringement litigation.

Recently, there has been a rise in meritless infringement lawsuits abusing our patent system. These litigants—so-called patent trolls or non-

5. See Shane D. Anderson, Software, Abstractness, and Soft Physicality Requirements, 29 HARV. J.L. & TECH. 567, 577–78 (2016) (“Software patents generally have broad claims that are fuzzy and ambiguous, and thus do not give clear notice of either the claimed invention or the patentee’s rights.”); see also James Bessen & Robert M. Hunt, An Empirical Look at Software Patents, 16 J. ECON. & MGMT. STRATEGY 157, 170 (2007) (showing that software patents contain 33% more claims than other patents on average).

6. See Anderson, supra note 5, at 577 (citing Carl Shapiro, Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard Setting, 1 INNOVATION POL’Y & ECON. 119, 119 (Adam B. Jaffee et al. eds., 2001)) (stating that a patent thicket is “an overlapping set of patent rights requiring that those seeking to commercialize new technology obtain licenses from multiple patentees”).


8. See James Bessen, A Generation of Software Patents, 18 B.U. J. SCI. & TECH. L. 241, 248 (2012) (“When firms acquire large numbers of patents, they can restrict entry into an industry and they can use these patents to extract rents from other firms beyond the rents needed to encourage innovation.”); see also Anderson, supra note 5, at 577 (“Thickets cause undue hold-up at firms, increase the cost of entry for new players, increase the cost of R&D, and suppress innovation.”); Julie E. Cohen & Mark A. Lemley, Patent Scope and Innovation in the Software Industry, 89 CALIF. L. REV. 1, 4–5 (2001) (“[W]ith software patents now being issued in large numbers, the patent system plays a newly prominent role in shaping the development of the software industry . . . [but] it is also possible that the patent system may constrain innovation if it draws protection too broadly.”).

9. See Bessen, supra note 8, at 248 (“Prospective innovators consider the risk of subsequent litigation when deciding whether to invest in research and development.”).

10. See GAO-16-490, supra note 7, at 1 (finding that new patent infringement lawsuits increased from about 2,000 in 2007 to more than 5,000 in 2015); Anderson, supra note 5, at
practicing entities (NPEs)—often instigate a barrage of litigation against myriad defendants for the sake of less money than the cost of defending against spurious claims. Others essentially extort high licensing fees for bad patents on fundamental concepts that somehow passed muster during the examination process at the United States Patent and Trademark Office (USPTO). “Ultimately, the victim innovator must pour resources into licensing or litigation against bad patents, rather than into socially beneficial innovation.” The GAO reports that district court filings of patent infringement lawsuits have increased from about 2,000 in 2007 to more than 5,000 in 2015, and most of these suits involved software-related patents and computer and communications technologies. At the same time, the USPTO has gone from issuing 100,000 patents a year in 1996 (and less than 50,000 a few years before that) to issuing more than 320,000 patents twenty years later. With at least three times the available assets, it is fair to expect that a growing number of software patent suits on the federal court dockets will continue to exacerbate the high overall cost of nuisance litigation. Likewise, patent utility applications are at an all-time high, with more than 600,000 new applications arriving in 2016.

The innovative software industry grows and evolves rapidly. Some point out that the slow-moving patent examination process does not interact well with the fast-moving software industry. Many software patents do not go to software companies but rather to LLCs and companies outside of the software industry. Many scholars point out that the software patent thicket
problem “arises from the flood of patents that are granted by the [USPTO] each year,” and this is evidenced by the more than 300,000 utility patents granted in 2016 alone.\textsuperscript{19} Intuitively, having more patents results in a higher probability of overlaps—particularly when many innovators remain willfully ignorant of what is patented.\textsuperscript{20}

Despite arguments that software patents create impediments to the software industry rather than spur innovation in any meaningful way,\textsuperscript{21} software patents can and should still be a valuable tool to promote public information and innovation; to facilitate American businesses through limited monopolies; and to provide a plethora of other economic benefits. But harnessing the value of software patents requires weeding out thousands of low-quality claims. The solution? Significant structural and regulatory reforms in the examination process at the USPTO, which currently allows hundreds of thousands of patents to issue each year, resulting in a state of affairs that leads to non-practicing entities like Intellectual Ventures amassing portfolios of tens of thousands of patents.\textsuperscript{22}


\textsuperscript{21} See id. at 27 (arguing that competition is a better spur to innovation than the monopoly granted through patent law).

\textsuperscript{22} Intellectual Ventures’ website boasts that “[w]e’ve acquired 95,000 patents and patent applications over our company’s lifespan. Today, we have approximately 30,000 in active monetization programs that span 50 technology areas . . . and rising.” Our Patent Portfolio, INTELLECTUAL VENTURES, http://www.intellectualventures.com/inventions-patents/patent-portfolio/ (last visited Oct. 11, 2017) (emphases added). For example, while difficult to verify, assignment records show that for Intellectual Ventures Management, one of Intellectual Ventures’ many subsidiaries, more than 152 U.S. patent assets have been listed alone. See, e.g., Assignment Database, USPTO, https://assignment.uspto.gov/patent/index.html#/patent/search (last visited Oct. 27, 2017). This only includes assigned assets that have been voluntarily recorded. As Professor Jorge Contreras notes, the only penalty parties generally pay from keeping their transfers and sales of assets and patent licenses secret (until occasionally forced by courts to record) is the old intervening bona fide purchaser problem those who took property in law school should remember well. Unlike motorcycles, cars, automobiles, homes, land, and hunting licenses, there is no government-maintained recordation associated with real legal benefit, and many NPEs use this fact to their advantage, masking ownership until necessary. See Jorge L. Contreras, Patent Pledges, 47 ARIZ. ST. L.J. 543, 601–02 (2015); Jorge L. Contreras, Assertions of Standards-Essential Patents by Non-Practicing Entities, in PATENT ASSERTION ENTITIES & COMPETITION POLICY 50 (D. Daniel Sokol ed., 2017). Interestingly, there was a proposed rule to that effect in 2014 that has been discarded, but that would have required disclosure of the ultimate parent corporate entity on pain of abandonment, among other things. See generally Changes to Require Identification of
This Article proposes means to deal with the problem at its source by eliminating “low-quality” patents from issue through structural and regulatory reforms in the USPTO’s examination process. Part II explores the Supreme Court and Federal Circuit decisions before and after *Alice*. This part provides a detailed background on the development of software patenting through court decisions and how these decisions have shaped the software industry. Part III discusses the impact of software patents on the software industry and software companies ranging from startups to large entities. Part IV emphasizes the importance of having software patents and proposes four solutions to combat the software patent thicket and the burgeoning issue of low-quality patent claims issuing from the USPTO.

II. SOFTWARE PATENT HISTORY

A. Prior to the Supreme Court’s Alice Decision

For decades, courts and the USPTO have been struggling to determine whether and how to protect software. Unfazed, the industry and innovation advanced at breakneck speed. For instance, “[i]n the late 1960s, the USPTO took a hard anti-software stance, issuing examination guidelines in 1968 that held computer programs generally unpatentable.” In parallel, the Supreme Court, in *Gottschalk v. Benson*, analyzed a general-purpose digital computer that converted signals from binary-coded decimal into pure binary numbers. The Court first explained that “[p]henomena of nature . . . mental processes, and abstract intellectual concepts are not patentable, as they are the basic tools of scientific and technological work.” The Court then ruled that the claims in question were “so abstract and sweeping as to cover both known and unknown uses” of the mathematical algorithm in question because the mathematical formula involved had “no substantial practical application except in connection with a digital computer.” The Court’s primary concern was that granting such a patent on basic software would “wholly pre-empt” the mathematical formula and allow for its monopolization. From the beginning, the Court expressly stated that an

---

23. See Anderson, supra note 5, at 569.
25. *Id.* at 67.
26. *Id.* at 68, 71.
27. *Id.* at 71–72.
intangible invention would not satisfy statutory muster and that the invention must have some physical effect to be patentable.\textsuperscript{28}

Five years later, the Court in \textit{Parker v. Flook} highlighted the concern set forth in \textit{Benson}.\textsuperscript{29} In \textit{Flook}, driven by concerns that a “competent draftsman” should not be able to transform a patent-ineligible abstract idea into a patent-eligible application by appending conventional or obvious “post-solution activity,” the Court held that a claim reciting an abstract idea must also include an “inventive concept” to be patent-eligible.\textsuperscript{30} Under this new “inventive concept” threshold, a “process is unpatentable under Section 101, not because it contains a mathematical algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention.”\textsuperscript{31}

The combination of \textit{Benson} and \textit{Flook} virtually eliminated patent protection for software—\textit{Benson} labeled algorithms, and thus by extension all software, as patent-ineligible abstract ideas, and \textit{Flook} by indicating that, as long as an invention’s sole point-of-novelty lay in the software, no additional limitation could be “enough” to confer patent eligibility.\textsuperscript{32}

The turning point for software patentability was \textit{Diamond v. Diehr}, in which the Court reversed course, effectively lowering the threshold of patent eligibility for software inventions.\textsuperscript{33} In \textit{Diehr}, the Court upheld the

\begin{itemize}
\item \textsuperscript{28} Id. at 70.
\item \textsuperscript{29} Parker v. Flook, 437 U.S. 584, 585 (1978).
\item \textsuperscript{30} Id. at 590, 594.
\item \textsuperscript{31} Id. at 594.
\item \textsuperscript{32} Ognjen Zivojnovic, \textit{Patentable Subject Matter after Alice—Distinguishing Narrow Software Patents from Overly Broad Business Method Patents}, 30 BERKELEY TECH. L.J. 807, 813 (2015).
\item \textsuperscript{33} The case below, authored by the well-respected Court of Claims and Patent Appeals Judge Giles Sutherland Rich, was a watershed for software patents. In it, he wrote: A claim drawn to a process or method does not depend for its validity under 35 U.S.C. § 101 on whether a computer is involved. If the claim is drawn to subject matter, which is otherwise statutory, it does not become nonstatutory merely because a computer is involved in its execution. Thus, the fact that an invention is drawn to a computer program or involves a computer is an observation which does nothing to aid in the determination of compliance with [section] 101 . . . . Therefore, any rejection which is based solely on the determination that a computer or computer program is involved is insupportable because it is overly broad and must be reversed as being without basis in the law.
\end{itemize}
patentability under Section 101 of a computer-controlled process for curing synthetic rubber.\footnote{Diehr, 450 U.S. at 176–77 (Specifically, the invention involved a process for curing synthetic rubber by monitoring the real-time conditions inside the mold. This process was “possible using well-known time, temperature, and cure relationships to calculate by means of the Arrhenius equation when to open the press and remove the cured product.”).} Although unapplied mathematical algorithms, formulas, and equations are considered unpatentable abstract ideas, the Court in Diehr made clear that the presence of such mathematical subject matter in a patent claim does not necessarily deprive the claim of potential patentability under Section 101.\footnote{Id. at 187.} Contrary to the previous decisions, the Court concluded that Diehr did not seek to patent a mathematical formula, but rather sought to protect a process of curing synthetic rubber.\footnote{Id. at 192.} This process “admittedly employ[ed] a well-known mathematical equation, but [it did] not seek to pre-empt the use of that equation.”\footnote{Id. at 187.} One did not need a computer to cure natural or synthetic rubber, but if the computer use incorporated in the process significantly lessened the possibility of “overcuring” or “undercuring,” the process as a whole did not thereby become unpatentable subject matter.\footnote{Id.}

Here, the Court redefined the contours of patent eligibility for software by paring back the ineligibility barrier created in Benson and Flook. Thereafter, “software related patents began to trickle into [and thus out of] the [USPTO] as skillful patent prosecutors began to couch their software claims into patentable processes.”\footnote{See Jack George Abid, Software Patents on Both Sides of the Atlantic, 23 J. MARSHALL J. COMPUTER & INFO. L. 815, 825 (2005).}

In the thirty years following the Diehr decision, “[u]ncertainty regarding software’s patent eligibility returned in the wake of the Supreme Court’s 2010 Bilski v. Kappos and 2012 Mayo Collaborative Services v. Prometheus Laboratories, Inc. decisions.”\footnote{See Zivojnovic, supra note 32, at 816.} These cases, while not categorically excluding software innovation from the patent system, effectively “raised the ‘enough’ threshold for patent eligibility, so that § 101 again became a substantive hurdle for software.”\footnote{See id. (citing Bilski v. Kappos, 561 U.S. 593 (2010); Mayo Collaborative Servs. v. Prometheus Labs., Inc., 566 U.S. 66 (2012)).}
The Bilski decision stemmed from the Federal Circuit’s now-infamous “machine-or-transformation” test.\textsuperscript{42} The key issues for the Federal Circuit were whether Bilski was “seeking to claim a fundamental principle (such as an abstract idea) or a mental process,” and if so, whether Bilski’s claim “would pre-empt substantially all uses of that fundamental principle if allowed.”\textsuperscript{43} The Federal Circuit held that “[a] claimed process is surely patent-eligible under § 101 if: (1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing.”\textsuperscript{44} Bilski’s claimed invention involved a method of hedging risk in the field of commodities trading.\textsuperscript{45} With the understanding of “transformation” in hand, the Federal Circuit returned to its analysis of Bilski’s claims and concluded that they did not satisfy the “transformation” prong of the machine-or-transformation test because Bilski’s claimed process for hedging risk in commodities trading involved the “[p]urported transformations or manipulations simply of public or private legal obligations or relationships, business risks, or other such abstractions . . . .”\textsuperscript{46} The court further explained that such abstractions cannot satisfy the transformation test because “they are not physical objects or substances” like the rubber cured in Diehr, and “they are not representative of physical objects or substances.”\textsuperscript{47} Here, the court did not resolve the threshold disclosure required for software patents to remain patent-eligible.\textsuperscript{48} Rather, the court vaguely asserted that transforming data representing a physical object and displaying such data was sufficient to meet the transformation prong of the test.\textsuperscript{49}

The Supreme Court’s subsequent ruling in Bilski left the patent eligibility of software uncertain. While the Court affirmed the Federal Circuit’s decision, it reversed the Federal Circuit’s analysis, concluding that even though it was “an important and useful clue,” “[t]he machine-or-transformation test is not the sole test for deciding whether an invention is a patent-eligible ‘process’” under Section 101.\textsuperscript{50} The Court refused to adopt categorical rules that might have wide-ranging and unforeseen impacts.

\begin{itemize}
  \item \textsuperscript{42} In Re Bilski, 545 F.3d 943 (Fed. Cir. 2008) (en banc), aff’d but criticized by sub nom. Bilski, 561 U.S. 593 (2010).
  \item \textsuperscript{43} In Re Bilski, 545 F.3d at 952–53.
  \item \textsuperscript{44} Id. at 954 (citing Gottschalk v. Benson, 409 U.S. 63, 70 (1972)).
  \item \textsuperscript{45} Id. at 949.
  \item \textsuperscript{46} Id. at 963.
  \item \textsuperscript{47} Id. at 945.
  \item \textsuperscript{48} See generally In Re Bilski, 545 F.3d 943 (Fed. Cir. 2008) (en banc), aff’d but criticized by sub nom. Bilski v. Kappos, 561 U.S. 593 (2010).
  \item \textsuperscript{49} In re Bilski, 545 F.3d at 963.
  \item \textsuperscript{50} Bilski, 561 U.S. at 604.
\end{itemize}
Instead, the Court resolved the case “narrowly on the basis of the Court’s [prior] decisions in Benson, Flook, and Diehr . . . .” 51 In view of Benson, Flook, and Diehr, the Court concluded that the basic concept of hedging is an unpatentable abstract idea and that “[a]llowing petitioners to patent risk hedging would pre-empt use of this approach in all fields, and would effectively grant a monopoly over an abstract idea.” 52 However, the Court did “not define further what constitutes a patentable ‘process’” and refrained from going beyond the definition provided in Section 100(b). 53

Relatedly, the Supreme Court in Mayo was examining the same question as Bilski in the context of drug dosage, but it did not add much substantively to patent eligibility of software inventions. 54 In Mayo, the invention at issue involved a method for calibrating the proper dosage of a drug for treating autoimmune diseases such as Crohn’s disease and ulcerative colitis. 55 The key exception there was “law of nature” rather than “abstract idea.” The Court again directed its attention to Diehr and Flook to determine whether an invention drawn to laws of nature must have additional steps to transform the process into an inventive application of the formula. 56 The Court explained that the additional steps in Diehr “apparently added to the formula something that in terms of patent law’s objectives had significance—they transformed the process into an inventive application of the formula.” 57 In determining whether the claims transformed unpatentable natural correlations into patentable applications, the Court inquired about whether “any additional steps consist[ed] of well-understood, routine, conventional activity already engaged in by the scientific community . . . .” 58 The Court found that the claim, simply telling doctors to (1) measure the current level of metabolite; (2) use particular laws of nature to calculate the current toxicity/inefficacy limits; and (3) reconsider the drug dosage based off of those same laws of nature, added nothing to the laws of nature. 59

---

51. Id. at 609.
52. Id. at 611–12.
53. Id. at 612.
55. Id. at 73.
56. Id. at 72–73.
57. Id. at 82 (citing Parker v. Flook, 437 U.S. 584, 594 (1978)) (contrasting claims with the claims in Flook, which were process claims that supplemented the recitation of a mathematical formula for updating alarm limits, and stating they were “well known” such that there “was no ‘inventive concept’ in the claimed application of the formula”).
59. Id. at 82.
the Court held that “those [additional] steps, when viewed as a whole, add nothing significant beyond the sum of their parts taken separately.”

“After Bilski and Mayo, the Federal Circuit issued a series of seemingly inconsistent decisions regarding the patent eligibility of claims directed to computer-implemented software or algorithms.”

Lacking a coherent framework or affirmative guidance from the Supreme Court, the Federal Circuit continued to flounder, working diligently to apply vague precedent panel by panel while navigating the swamp of verbiage regarding Section 101. “The Federal Circuit also had to contend with increasingly hostile public outcry against patent protection for these types of inventions.” At the time, some argued that “the outcome of any particular case depended more on which judges heard the case, rather than the actual merits and the claim language.”

In an attempt to clarify the patent protection for software and computer-related inventions, the Federal Circuit in 2013 published a bitterly divided en banc opinion in CLS Bank International v. Alice Corp. Pty. Ltd. with six different opinions, including one set of “additional views” by then-Chief Judge Rader. Unfortunately, the unusually fractured opinion failed to provide guidance to the bar or the lower courts on what constituted an abstract idea or what constituted “enough” to be an inventive concept to overcome an abstract idea.

1. The Supreme Court’s Alice Decision

In 2007, CLS Bank International (CLS Bank) sued Alice Corporation Pty. Ltd. (Alice) seeking declaratory judgment of non-infringement, invalidity, and unenforceability of Alice’s patents. Alice was the assignee of several patents at issue that “enable[d] the management of risk relating to

60. Id. at 80.

61. See Zivojnovic, supra note 32, at 818.


63. Id.

64. See Zivojnovic, supra note 32, at 818–19.


66. See id. at 1290 (stating that the appropriate question was whether the computer-based limitations added “enough” beyond the abstract idea itself without further explanation of what was “enough”).

67. Id. at 1274.
specified, yet unknown, future events." Specifically, the claims were designed "to facilitate the exchange of financial obligations between two parties by using a computer system as a third-party intermediary." All of the claims [were] implemented using a computer . . . .

As noted, a divided three-judge panel of the Federal Circuit held that the claims at issue were directed to an abstract idea, and on rehearing en banc, the court narrowly reversed the three-panel decision and affirmed the judgment of the district court that the claims were patent-ineligible. Seven of the ten judges participated in the opinion, and "the only thing a majority of the court agreed on was that the invention in question was not eligible for patenting; there was no agreement as to why."

The Supreme Court granted certiorari to review the fractured en banc decision. The claims at issue involved "a computerized scheme for mitigating 'settlement risk' . . . ."

During oral argument, Mark A. Perry, counsel for the respondent, noted that the CBM method Congress created at the USPTO provided a workable

69. Id.
70. Id. at 2353.
71. Id.
72. Osenga, supra note 62, at 1842; see, e.g., Zivojnovic, supra note 32, at 818 ("One faction of the court argued that a computer must 'play a significant part in' or 'be integral to' the claimed invention, 'facilitating the process in a way that a person making calculations or computations could not' for the limitation to be 'enough' to confer patent eligibility to an otherwise patent-ineligible abstract idea. The other faction argued that unless a claim is so lacking in additional limitation that it is 'manifestly evident that [it] is directed to a patent-ineligible abstract idea,' a claim incorporating an abstract idea is patent-eligible."); Bancorp Servs. LLC v. Sun Life Assurance Co. of Can., 687 F.3d 1266, 1278 (Fed. Cir. 2012).
73. Alice, 134 S. Ct. at 2352; see Zivojnovic, supra note 32, at 820 (providing an explanation of the invention in Alice, stating that "two individuals may agree in the morning to exchange currencies, but, to avoid allowing either party to gain a benefit by exactly timing its acceptance based on fluctuations in the exchange rate, the two individuals might agree that the exchange rate at the end of the day is controlling. When forming such a contract, a mechanism is necessary to ensure that each party is still able to pay its side of the bargain at the end of the day. The claimed invention solves this issue by having a trusted third-party keep track, via a so-called shadow credit and/or debit record, of a contracting party's financial transactions between the time a contract is initially formed (e.g., in the morning, when the two individuals agree to exchange currencies) and when it matures (e.g., at the end of the day, when the controlling exchange rate is set) . . . . Once the contract matures, the third-party instructs an exchange institution . . . to perform non-blocked financial transactions") (citing CLS Bank Int'l v. Alice Corp. Pty. Ltd., 768 F. Supp. 2d 221, 223, 228–29 (D.D.C. 2011), aff'd, 717 F.3d 1269 (Fed. Cir. 2013) (en banc)).
test that had been blessed by Congress and, it should be noted, is somewhat consistent with European practice. He noted:

Congress, in the CBM method, said business methods that are subject to special scrutiny—that is, dubious patents—include methods and corresponding apparatuses, which is what we have here, that pertain to data processing in the financial services industry and do not offer a technological solution. That describes Alice’s patents to a letter, Your Honor.

The oral arguments and briefing at the Supreme Court debated and raised some version of this technological test, among others. But while the Supreme Court agreed with Mr. Perry on the outcome, they neither adopted nor relied on any version of the “technological solution” test.

The Supreme Court unanimously held that all of the claims at issue were patent-ineligible, holding that “the claims at issue are drawn to the abstract idea of intermediated settlement, and that merely requiring generic computer implementation fails to transform that abstract idea into a patent-eligible invention.” The Court employed a version of their two-step analysis set forth in the Mayo framework, which first determines whether the claims at issue are directed to one of the patent-ineligible concepts. Second, the Court “consider[s] the elements of each claim both individually and ‘as an
ordered combination’ to determine whether the additional elements ‘transform the nature of the claim’ into a patent-eligible application.\textsuperscript{79}

In the first step, the Court considered the concept of intermediated settlement in conjunction with the concept of hedging in \textit{Bilski}.\textsuperscript{80} In \textit{Bilski}, the Court explained that “hedging is a fundamental economic practice long prevalent in our system of commerce and taught in any introductory finance class.”\textsuperscript{81} Therefore, “‘the concept of hedging’ as recited by the claims in suit was a patent ineligible ‘abstract idea’ . . . .”\textsuperscript{82} Similarly, the Court in \textit{Alice} held that the claims were drawn to abstract idea because “the concept of intermediated settlement is ‘a fundamental economic practice long prevalent in our system of commerce.”\textsuperscript{83} Under the second step, the Court held that “transformation into a patent-eligible application requires more than simply stat[ing] the [abstract idea] while adding the words ‘apply it.’”\textsuperscript{84} The Supreme Court then concluded that the claims at issue, including all of the method, system, and storage medium claims, do no more than “simply instruct the practitioner to implement the abstract idea of intermediated settlement on a generic computer.”\textsuperscript{85}

Troublingly, the Court was vague as to what qualified as an “abstract idea” and what was “significantly enough” to transform an abstract idea into patent-eligible subject matter.\textsuperscript{86} After determining that this \textit{particular} type of intermediate settlement at issue was an abstract idea, the Court did not further “delimit the precise contours of the ‘abstract ideas’ category . . . .”\textsuperscript{87} The Court also rejected petitioner’s argument that an idea must exist independent of human action in order to qualify for the abstract idea exception.\textsuperscript{88} With regards to the second prong of the test, whether the claims contain an inventive concept to transform the abstract idea into a patent-eligible application, the Court simply noted that “well-understood, routine, conventional activities previously known to the industry” were not sufficient to confer patent eligibility.\textsuperscript{89} The Court further determined that, with the

\begin{itemize}
  \item \textsuperscript{79} \textit{Id.} (citing Mayo Collaborative Servs. v. Prometheus Labs., Inc., 566 U.S. 66, 72–73 (2012)).
  \item \textsuperscript{80} \textit{Id.}
  \item \textsuperscript{81} \textit{Id.} at 2356 (citing Bilski v. Kappos, 561 U.S. 593, 611 (2010)).
  \item \textsuperscript{82} \textit{Id.} (citing \textit{Bilski}, 561 U.S. at 611).
  \item \textsuperscript{83} \textit{Id.}
  \item \textsuperscript{84} \textit{Id.} at 2357 (citing \textit{Mayo}, 566 U.S. at 72) (internal quotations omitted).
  \item \textsuperscript{85} \textit{Id.} at 2359.
  \item \textsuperscript{86} \textit{See id.} at 2357.
  \item \textsuperscript{87} \textit{Id.}
  \item \textsuperscript{88} \textit{Id.} at 2356–57.
  \item \textsuperscript{89} \textit{Id.} at 2359 (quoting \textit{Mayo}, 566 U.S. at 73) (internal quotations omitted).
\end{itemize}
ubiquity of computers, “wholly generic computer implementation” and a
generic computer performing generic-computer functions are insufficient to
protect against drafting efforts to monopolize an abstract idea, and thus
“wholly generic computer implementations” do not transform an abstract
idea into a patent-eligible subject matter.  

While many scholars will agree that “[t]he current collection of
unpatentable subject matter—abstract ideas, laws of nature, and natural
phenomena—is difficult to define and thus exclude from patenting[,] . . . [p]art of that difficulty stems from the lack of a consistent
theoretical underpinning for the doctrine.”  Compounding such existing
difficulty, the Supreme Court’s perceived lack of guidance has led to further
confusion among many practitioners, the lower courts, and the USPTO,
and it has also led to a surprising number of patents cancelled by the lower
courts.  Although the Alice decision seemingly ended pure business method
patents, it left much room for interpretation with regards to pure software
patents, and it remains to be seen how the Federal Circuit and the USPTO
will implement the Supreme Court’s ambiguous guidance. 

2. Federal Circuit Decisions Following Alice

After the Alice decision, many practitioners and the USPTO have been
relying on the Federal Circuit to clarify the Supreme Court’s notably
ambiguous opinion.  Interpreting Alice and other prior Supreme Court
cases, the Federal Circuit has been struggling to find a common ground in
applying the Alice two-prong test. To strike the balance between preventing
monopolization and promoting innovation, the Supreme Court has “urged
the courts and agency tribunals to ‘tread carefully’ going forward in
construing inventions and innovations to be non-patent-eligible abstract

---

90. Id. at 2358–59 (citing Mayo, 566 U.S. at 73).
91. See J. Jonas Anderson, Applying Patent-Eligible Subject Matter Restrictions, 17
92. See Scott Alter, One Year After Alice: Was it the Right Medicine?, LAW360 (June
18, 2015), https://www.law360.com/articles/654471/one-year-after-alice-was-it-the-right-
medicine (discussing the trend in lower courts of invalidating patent eligibility for not meeting
the requirements of Section 101).
93. Id.
94. See id. (discussing the “ambiguous language in that case,” cautioning it would lead
to “unpredictable applications”).
95. See Manatt Phelps & Phillips LLP, Once in a [Planet] Blue Moon—Software
Patents as Alice Turns Two, LEXOLOGY (Oct. 25, 2016), http://www.lexology.com/
library/detail.aspx?g =e4ab9515a9e640e8a73dd30881c6b7f7.
ideas ‘lest [the exclusionary principles] swallow all of patent law.’”

Nonetheless, from January to September 2016, the Federal Circuit invalidated fifty-three software patents out of fifty-eight total decisions on subject matter ground. This unsettling invalidation rate has worried some in the software industry and poses a threat to the protection of valuable software patents in our patent system.

These cases raise a serious question: Do we want a system where one arm grants 300,000 patents a year and two other arms—the federal courts and the Patent Trial and Appeal Board (PTAB)—cancel hundreds of claims?

As noted, a string of software patent invalidations followed the Alice decision. On July 11, 2014, the Federal Circuit held that Digitech’s device profile and method claims for creating a device profile within a digital image processing system were drawn to abstract idea and added no inventive concept and were therefore patent-ineligible.

The invention was directed to “the generation and use of an ‘improved device profile’ that describes spatial and color properties of a device within a digital image processing system.” In short, Digitech’s device profile “enabled a more accurate translation of the image’s pixel data into the independent color space and


99. One is reminded of the Dr. Seuss story The Sneetches. Dr. Seuss, The Sneetches and Other Stories (1953), http://www.csun.edu/~sm60012/GRCS-Files/Final%20Projects/The%20Sneetches.htm. In it, two groups, the Star-Belly Sneetches and the Plain-Belly Sneetches vie for cultural superiority until one day, Sylvester McMonkey McBean came and offered to put stars on the Plain-Belly Sneetches’ bellies. He did and then offered to remove the stars from the original Star-Belly Sneetches’ bellies. In the end, it became hopelessly confusing as to who was from what group, as they kept paying until they had paid their last cent. In the end, the only person who profited was, of course, Mr. McBean. A better patent system would prevent such churn, issuing only quality patents and avoiding the need for review and reversal. In such a system, the only parties that benefit are the lawyers—the McBeans of the world. The rest is a transaction cost and thus corporate waste.


101. Id. at 1347 (explaining that according to the ’415 patent, “all imaging devices impose some level of distortion on an image’s color and spatial properties. This distortion occurs because different devices (i.e., digital cameras, monitors, TVs, printers, etc.) allow for slightly different ranges of colors and spatial information to be displayed or reproduced”).
across the source and output devices.” 102 The district court found that “the ‘device profile’ claims [were] directed to a collection of numerical data that lacks a physical component or physical manifestation” and concluded that “a ‘device profile’ is nothing more than information,” and thus patent-ineligible. 103 The district court further concluded that “the asserted method claims for generating a device profile encompass the abstract idea of organizing data through mathematical correlations.” 104

On appeal, the Federal Circuit upheld the district court’s decision. In determining whether device profile claims were patent-eligible, the Federal Circuit first asserted that “[f]or all categories except process claims, the eligible subject matter must exist in some physical or tangible form.” 105 The device profile was merely “comprised of two sets of data”—one for color information and the other for spatial information. 106 The court determined that the claim did not recite anything “tangible or physical” to be eligible under “machine,” “manufacture,” or “composition of matter” under Section 101 subject matter. 107 With regards to the method claims, Digitech argued that the claims were “patent eligible because they describe a process for generating a device profile that is specifically tied to a digital image processing system and is integral to the transformation of a digital image.” 108 However, the court rejected Digitech’s argument and held that the claims were directed to “an abstract idea because it describes a process of organizing information through mathematical correlations and is not tied to a specific structure or machine.” 109 The court reasoned that the claims were merely “a process of taking two data sets and combin[ing] them into a single data set.” 110 Therefore, the court concluded that “[w]ithout additional limitations, a process that employs mathematical algorithms to manipulate existing information to generate additional information is not patent eligible.” 111

In Planet Bingo, Planet Bingo alleged that VKGS infringed its ’045 and ’646 patents. 112 The ’045 and ’646 patents claimed “managing a bingo game

---

102. Id. at 1348.
103. Id.
104. Id.
105. Id.
106. Id. at 1349.
107. Id. at 1348.
108. Id. at 1350.
109. Id.
110. Id. at 1351.
111. Id.
112. Planet Bingo, LLC v. VKGS LLC, 576 F. App’x 1005, 1006 (Fed. Cir. 2014).
while allowing a player to repeatedly play the same sets of numbers in multiple sessions.”\textsuperscript{113} On August 26, 2014, the Federal Circuit held that the claims at issue were “drawn to patent- ineligible subject matter.”\textsuperscript{114} In doing so, the court reasoned that “managing a game of Bingo” is analogous “to the kind of ‘organizing human activity’ at issue in \textit{Alice}.”\textsuperscript{115} Furthermore, the court concluded that the patents did not contain an “‘inventive concept’ sufficient to ‘transform’ the claimed subject matter into a patent-eligible application.”\textsuperscript{116} Here, Planet Bingo argued that “the patents recite ‘significantly more’ than an abstract idea because the invention includes ‘complex computer code with three distinct subparts.’”\textsuperscript{117} The court rejected that argument and held that the claims “recite a program that is used for the generic functions of storing, retrieving, and verifying a chosen set of bingo numbers . . . .”\textsuperscript{118}

On November 14, 2014, the Federal Circuit held that Ultramercial’s ’545 patent was patent- ineligible.\textsuperscript{119} The ’545 patent involved “a method for distributing copyrighted media products over the Internet where the consumer receives a copyrighted media product at no cost in exchange for viewing an advertisement, and the advertiser pays for the copyrighted content.”\textsuperscript{120} After reviewing the ordered combination of steps recited in the claims, the court determined that “the concept embodied by the majority of the limitations describe[d] only the abstract idea of showing an advertisement before delivering free content.”\textsuperscript{121} In sum, the court simply stated that the “method of using advertising as an exchange or currency” was directed to an abstract idea.\textsuperscript{122} In the second step in the analysis, the court determined that “the limitations of the ’545 claims do not transform the abstract idea . . . into patent-eligible subject matter because the claims simply instruct the practitioner to implement the abstract idea with routine, conventional activity.”\textsuperscript{123} Specifically, the court reasoned that “[a]dding routine additional steps such as updating an activity log, requiring a request

\begin{itemize}
  \item \textsuperscript{113} \textit{Id.} at 1007.
  \item \textsuperscript{114} \textit{Id.}
  \item \textsuperscript{115} \textit{Id.} at 1008 (quoting \textit{Alice Corp. Pty. Ltd. v. CLS Bank Int’l}, 134 S. Ct. 2347, 2356 (2014)).
  \item \textsuperscript{116} \textit{Id.} at 1009.
  \item \textsuperscript{117} \textit{See id.} at 1008.
  \item \textsuperscript{118} \textit{Id.} at 1009.
  \item \textsuperscript{119} \textit{Ultramercial, Inc. v. Hulu, LLC}, 772 F.3d 709, 722 (Fed. Cir. 2014).
  \item \textsuperscript{120} \textit{Id.} at 712.
  \item \textsuperscript{121} \textit{Id.} at 715.
  \item \textsuperscript{122} \textit{Id.}
  \item \textsuperscript{123} \textit{Id.}
\end{itemize}
from the consumer to view the ad, restrictions on public access, and use of
the Internet does not transform an otherwise abstract idea into patent-eligible
subject matter.”124 “Instead, the claimed sequence of steps comprises only
‘conventional steps, specified at a high level of generality,’ which is
insufficient to supply an ‘inventive concept.’”125 Thereafter, the court once
again rejected patent-eligibility of a software under Section 101.126 Shortly
after the Alice decision, the Federal Circuit invalidated all other software
claims under Section 101 until DDR Holdings, LLC v. Hotels.com, L.P.127

Despite the Federal Circuit’s alarming number of invalidations against
software patents, on December 5, 2014, for the first time since Alice, the
Federal Circuit held that DDR Holdings’ computer-implemented patent
claims overcame the Section 101 hurdle.128 DDR Holdings was an assignee
of a patent directed to “systems and methods of generating a composite web
page that combines certain visual elements of a ‘host’ website with content
of a third-party merchant.”129 The specification explained the drawbacks of
the prior art systems that the prior art systems “allowed third-party
merchants to ‘lure the host website’s visitor traffic away’ from the host
website because visitors would be taken to the third-party merchant’s
website when they clicked on the merchant’s advertisement on the host
site.”130

124. Id. at 716.
125. Id. (citing Alice Corp. Pty. Ltd. v. CLS Bank Int’l, 134 S. Ct. 2347, 2357 (2014)
(quoting Mayo Collaborative Servs. v. Prometheus Labs., Inc., 566 U.S. 66, 72–73, 79
(2012))).
126. Id. at 715–17.
127. See, e.g., In re TLI Commc’ns LLC Patent Litig., 823 F.3d 607, 615 (Fed. Cir.
2016) (concluding that a system and method claim for “classifying and storing digital images
in an organized manner” was drawn to abstract idea and the recitation of tangible components,
such as a “telephone unit” and a “server” failed to transform the abstract idea into a patent-
eligible invention); buySAFE, Inc. v. Google, Inc., 765 F.3d 1350, 1355 (Fed. Cir. 2014)
(concluding that the claims recited no more than using a computer to send and receive
information over a network in order to implement the abstract idea of creating a “transaction
F.3d 1336, 1344–45 (Fed. Cir. 2013) (concluding that claims merely recited “generalized
software components arranged to implement an abstract concept [of] generating [insurance-
policy-related] tasks based on rules to be completed upon the occurrence of an event” on
a computer); Bancorp Servs., LLC v. Sun Life Assurance Co. of Can., 687 F.3d 1266, 1278
(Fed. Cir. 2012) (concluding that claims recited no more than the use of a computer “employed
only for its most basic function, the performance of repetitive calculations,” to implement the
abstract idea of managing a stable-value protected life insurance policy).
129. Id. at 1248 (citing U.S. Patent No. 6,629,135, at 12:46–50).
The Federal Circuit applied the *Alice* two-prong test and found that the patented invention was not directed to abstract idea under the first step. The court reasoned that these claims stand apart from previously invalidated patents because “they do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the internet.” “Instead, the claimed solution [was] necessarily rooted in computer technology in order to overcome a problem specifically arising in the realm of computer networks.” The majority rejected the dissent’s patent-ineligible abstract idea argument in view of the “store within a store” concept, which suggested a warehouse store that contains a kiosk for selling a third-party’s goods may “lure away” customers’ attention from other kiosks. The majority reasoned that such nature does not apply to the “nature of an Internet ‘location’ or the near-instantaneous transport between these locations made possible by standard Internet communication protocols, which introduces a problem that does not arise in the ‘brick and mortar’ context.”

Though this case opened the door to software patent validations, only a few software patents were held patent-eligible under Section 101 in the following years. More importantly, this case did not clearly define what constitutes “abstract idea” and did not provide any more instruction than what was given in *Alice*.

The following three Federal Circuit cases are more instructive but still adopted no bright-line rule. On May 12, 2016, the Federal Circuit in *Enfish, LLC v. Microsoft Corp.* held that a “self-referential table,” a specific type of data structure designed to improve the way a computer stores and retrieves data in memory, was not directed to abstract idea. Here, “the claims [were] not simply directed to any form of storing tabular data, but instead [were] specifically directed to a self-referential table for a computer database.” The court distinguished that this self-referential table “achieves other benefits over conventional databases, such as increased flexibility, faster search times, and smaller memory requirements.”

131. *Id.* at 1259.
132. *Id.* at 1257.
133. *Id.*
134. *Id.* at 1258 (citing U.S. Patent No. 6,629,135, at 2:26–30).
135. *Id.*
136. See infra APPENDIX A (showing that only 274 patents were invalidated under 35 U.S.C. § 101 between June 19, 2014 and December 22, 2016).
137. *See DDR Holdings*, 773 F.3d at 1255–59.
139. *Id.* at 1337 (emphasis omitted).
140. *Id.*
the court held that the claims at issue were “directed to an improvement in the functioning of a computer,” and thus, the court was “not persuaded that the invention’s ability to run on a general-purpose computer dooms the claims.”\textsuperscript{141} Thereafter, the court concluded that the claims were directed to a specific implementation of a solution to a problem in the software arts; therefore, it was not drawn to an abstract idea.

In \textit{Bascom Global Internet Services, Inc. v. AT&T Mobility LLC}, the claimed invention was directed to a customizable Internet content filtering scheme.\textsuperscript{142} The court agreed with the lower court that “filtering content is an abstract idea because it is a longstanding, well-known method of organizing human behavior . . . .”\textsuperscript{143} However, under the second step, the court found that the claims “do not preempt the use of the abstract idea of filtering content on the Internet or on generic computer components performing conventional activities” because the claims “carve out a specific location for the filtering system” by establishing a remote ISP server and “require the filtering system to give users the ability to customize filtering for their individual network accounts.”\textsuperscript{144} The court concluded that this filtering scheme was “significantly more than the abstract idea itself” and that it was thus patent-eligible subject matter.\textsuperscript{145}

One of the most recent Federal Circuit cases to uphold the validity of a software patent was \textit{McRO, Inc. v. Bandai Namco Games America Inc.}\textsuperscript{146} The patent claims at issue involved the automation of a preexisting 3-D animation method where the software manipulated the facial expressions of a 3-D character with the implementation of a specific algorithm.\textsuperscript{147} The court

\textsuperscript{141} Id. at 1338. \textit{But cf. In re TLI Commc’ns LLC Patent Litig.}, 823 F.3d 607, 615 (Fed. Cir. 2016) (the claims fail to add or transform abstract idea into patent-eligible invention because it simply receives data, extracts classification information from the received data, and stores the digital images); Mortg. Grader, Inc. v. First Choice Loan Servs. Inc., 811 F.3d 1314, 1324 (Fed. Cir. 2016) (claims attaching generic computer components to perform “anonymous loan shopping” are not patent-eligible); Versata Dev. Grp. v. SAP Am., Inc., 793 F.3d 1306, 1334 (Fed. Cir. 2015) (computer performed “purely conventional” steps to carry out claims directed to the “abstract idea of determining a price using organization and product group hierarchies”).

\textsuperscript{142} Enfish, 822 F.3d at 1339.

\textsuperscript{143} Bascom Glob. Internet Servs., Inc. v. AT&T Mobility LLC, 827 F.3d 1341, 1345 (Fed. Cir. 2016).

\textsuperscript{144} Id. at 1348.

\textsuperscript{145} Id. at 1352.

\textsuperscript{146} Id. at 1349, 1352.

\textsuperscript{147} McRO, Inc. v. Bandai Namco Games Am. Inc., 837 F.3d 1299, 1316 (Fed. Cir. 2016).

\textsuperscript{148} Id. at 1307.
noted that “the concern underlying the exceptions to § 101 is not tangibility, but preemption.”

Giving more weight to the preemption concern, the court held that the claim was not drawn to abstract idea because, “[b]y incorporating the specific features of the rules as claim limitations, [the claim] is limited to a specific process for automatically animating characters using particular information and techniques and does not preempt approaches that use rules of a different structure or different techniques.”

The McRO decision has been interpreted by many scholars in the patent community as a signal that the so-called pendulum is swinging back to a more favorable position for patentees. However, while the McRO decision has been viewed as a positive development towards software patent validity, the Federal Circuit continues to affirm more invalidity decisions than it reverses. In fact, there have been 126 decisions by the Federal Circuit from October 2016 to April 2017, and the court has affirmed ineligibility in eighty-four (or 67%) of the decisions. The good news is that the percentage of invalidated software patents has dropped from 61% in 2015 to 54% in 2016, but it is still too early to predict that this will be the trend in coming years.

Substantively, as can be seen from the recent Federal Circuit cases mentioned above, the court has employed no bright-line rule to test abstract idea nor has it clearly explained what constitutes “inventive concept” that adds or improves “significantly more” than mere abstract idea. For example, the Digitech court invalidated the method of creating device profile because it merely used mathematical formulation to transform color and spatial information of an image into numerical data. However, the McRO court validated the method of automating 3-D facial expressions of an
animated character with specific mathematical formulation from a predetermined data.\textsuperscript{157} Even though both inventions were similar in nature, the former was drawn to abstract idea and the latter was not. In McRO, the court put more emphasis on preemption than tangibility and held that McRO’s approach did not preempt others from arriving at the same invention.\textsuperscript{158} While the court could have held that Digitech’s method of creating device profile did not preempt others from arriving at the same invention, the Federal Circuit refused to validate Digitech’s patent claims.\textsuperscript{159}

Thus, the aforementioned cases leave scholars and practitioners asking several questions. Where is the line to distinguish something that is preemptive from something that is not? Where is the line between tangible and intangible software invention? What constitutes an abstract idea, and what is the threshold for the patentee to overcome the “inventive concept” to be patent-eligible? This continuing trend of lack of guidance from the Federal Circuit has been confusing to many practitioners and USPTO examiners and has resulted in an unprecedented number of rejections under Section 101.\textsuperscript{160} Thus, the central question plaguing software patents—what additional limitation is “enough” to transform a patent-ineligible abstract idea into a patent-eligible application—remains unanswered.\textsuperscript{161}

III. CURRENT STATE OF THE SOFTWARE BUSINESS

A. Problems in the Software Industry

The patent system has played a prominent role in shaping the development of the software industry.\textsuperscript{162} Particularly for small startup software companies, a startup firm with limited capital often relies on patent protection to solicit investors to raise capital or as a tether to valuation when the company is being purchased.\textsuperscript{163} Some scholars point out that “[i]f the startup has developed software that spurs the development of a new technology, or serves as a base platform for the development of new

\begin{itemize}
  \item \textsuperscript{157} See McRO, 837 F.3d at 1316.
  \item \textsuperscript{158} See id. at 1315.
  \item \textsuperscript{159} Digitech, 758 F.3d at 1351.
  \item \textsuperscript{160} See infra APPENDIX A.
  \item \textsuperscript{161} See Zivojnovic, supra note 32, at 807–08 (quoting Mayo Collaborative Servs. v. Prometheus Labs., Inc., 566 U.S. 66, 77 (2012)).
  \item \textsuperscript{162} Cohen & Lemley, supra note 8, at 4–5.
  \item \textsuperscript{163} See GERALD B. HALT, JR. ET AL., INTELLECTUAL PROPERTY IN CONSUMER ELECTRONICS, SOFTWARE AND TECHNOLOGY STARTUPS 92 (2014).
\end{itemize}
technology, the implications of being the owner of that software could be immensely profitable if the [intellectual property] rights are properly protected.\footnote{164}

Contrarily, some scholars argue that “[s]oftware simply does not fit the patent system.”\footnote{165} Shane Anderson points out that “the development and lifecycle of most software is short and technologies are quickly supplanted, whereas patent prosecution is extremely slow, taking years.”\footnote{166} He further argues that “the twenty-year term of most patents is massively disproportionate to the lifespan of most software” and concludes that such a system “works as a barrier to innovation by locking up ideas from those who could contribute to follow-on innovation.”\footnote{167} Given the high rate of failure or purchase of startup businesses, small companies seeking patent protection are, more often than not, selling their assets on the secondary market or during an acquisition or bankruptcy proceedings.\footnote{168} Given the twenty-year term, this results in a relatively liquid secondary and tertiary market for patent assets being passed between non-practicing entities or aggregated by larger technology firms,\footnote{169} which themselves are prone to failure, change, and merger.\footnote{170}

James Bessen, in his extensive study of software patents, found that “relatively few software firms chose to acquire patents” and that “software patents were much more likely than other patents to be involved in litigation.”\footnote{171} He hypothesizes that some software firms go without patents due to a lack of social benefit, a high probability of litigation, and high

\footnote{164. Id.}
\footnote{165. Anderson, supra note 5, at 575.}
\footnote{166. Id.}
\footnote{167. Id.}
\footnote{169. See id. (discussing the value of the brokered market and the $165 million of brokered patent sales in 2016); see, e.g., ICAP PATENT BROKERAGE, http://icappatentbrokerage.com/ (last visited Apr. 28, 2017) (illustrating the common brokerage of patents).}

171. Bessen, supra note 8, at 242–43 (citing JAMES BESSEN & MICHAEL J. MEURER, PATENT FAILURE: HOW JUDGES, BUREAUCRATS, AND LAWYERS PUT INNOVATORS AT RISK 18 (2008)).
costs.\textsuperscript{172} He also points out that “[p]ossibly, the inexperience of patent examiners, the unfamiliarity of software firms with the patent process, and the difficulty of legal interpretation in court cases for this new subject matter” may explain the relatively low rate of patent turnout by practicing software companies.\textsuperscript{173}

Some scholars argue that patents provide “a real benefit to startups because they signal venture capitalists about the quality of the startup’s technology.”\textsuperscript{174} If this argument holds true, a patent is expected to provide “a signal of technology quality or of the quality of firm management” and should provide an important benefit to startup firms by facilitating investment.\textsuperscript{175} However, Bessen’s study finds that “most software patents still go to non-software firms.”\textsuperscript{176} By “non-software firms” he means patent trolls or, mostly, companies that not only produce software but also produce other hardware components. Bessen’s study reports the share of patents for publicly listed software firms and reveals that patents for startup companies decreased from 16.8% to 13.8% between 1996 and 2006, despite the overall number of software patents issued.\textsuperscript{177} This decline is notable because it “occurred at a time when it was well understood that software, including business methods . . . [was] patentable.”\textsuperscript{178} “It likely shows that these innovators understood that patents and software often do not work well together or that patents are unnecessary for software innovation.”\textsuperscript{179}

Moreover, the increase in the share of software patents granted to software firms is largely accounted for by the activity of a small number of large software firms.”\textsuperscript{180} Bessen’s article provides the lists of the top ten software patent recipients in 1996 and 2006.\textsuperscript{181} The lists reveal that large, well-known software firms such as Microsoft, Oracle, Adobe Systems, and SAP have increased their patenting by an order of magnitude, and “this

\begin{thebibliography}{99}
\item \textsuperscript{172} Id. at 243.
\item \textsuperscript{173} Id.
\item \textsuperscript{174} Id. at 257. To be sure, issued assets provide collateral and can form the basis of a lien, sale, or valuation. That tangibility—the ability to form the basis of loans, collateral, or other transfer—can be valuable to companies with no other tangible assets other than employee know-how—can be valuable to companies with no other tangible assets other than employee know-how. But with the startup lifecycle rarely approaching more than one to five years, a twenty-year term is too high a price to pay in exchange for a legal placeholder.
\item \textsuperscript{175} Id.
\item \textsuperscript{176} Id. at 256.
\item \textsuperscript{177} Id. at 255.
\item \textsuperscript{178} Anderson, \textit{supra} note 5, at 576.
\item \textsuperscript{179} Id.
\item \textsuperscript{180} Bessen, \textit{supra} note 8, at 256.
\item \textsuperscript{181} Id. at 257.
\end{thebibliography}
accounts for most of the increase in patents going to the software industry..."182 Furthermore, he notes that these few large firms account for about 76% and 81% of software patents granted to the software industry in 1996 and 2006, respectively.183 Indeed, the most frequent grantee of 2016 was IBM, with more than 8,000 patents issued in 2016 alone, followed by Samsung, Canon, Qualcomm, Google, Intel, and LG.184 It also stands to reason that at least one of those entities will either fail, merge, sell, split, or drastically change direction in the next twenty years, meaning that thousands of those patents, as liquid assets, could find their way to the open market.185

While it is relatively cheap to prosecute a patent—anywhere from $5,000 to $20,000 is a standard quoted rate for legal fees186—many startup firms grow wary of obtaining a software patent out of fear that they may be subject to litigation.187 Clearly, this risk of litigation is easier borne by a multinational firm than by a small startup, where a one-time $20,000 spend can break a company. As mentioned earlier, the GAO-13-490 reports that there has been a significant increase in the number of patent infringement lawsuits.188 The GAO also found that most patent suits involve software-related patents or computer and communications technologies.189 Looking at this trend, it is difficult for startup firms to ignore possible litigation costs arising from their self-executed patent. In the GAO-13-465 report, one stakeholder revealed that “the cost of defending one patent infringement lawsuit, which excludes any damages awarded, was from $650,000 to $5 million in 2011, depending on how much was at risk.”190 "As for damages

182. *Id.* at 256–57.
183. *Id.*
185. The authors note that the current landscape was created by a “largescale increase in patenting of software” starting roughly twenty years ago. What will the market and litigation look like ten years from now, with roughly three times as many patents in circulation? See James Bessen & Robert M. Hunt, *The Software Patent Experiment*, FED. RES. BANK OF PHILA. BUS. REV. 1, 9 (2004).
187. See Bessen, supra note 8, at 243 (discussing how high litigation rates may create disincentives for investing).
188. See GAO-16-490, supra note 7, at 14.
189. *Id.* at 14, 21.
awarded, a 2012 study that looked at all district court patent decisions that proceeded through trial from 1995 to 2011 found that the median damage award was over $5 million dollars and that damage awards in [nonpracticing entity] cases were higher than in other types of suits. Though larger companies are likely facing higher litigation costs, expensive litigation costs probably have a more overwhelming impact on startups since they are not as well-funded and are likely relying on fewer inventions to carry out their businesses. That forces small startups to turn to contingency counsel for assertions, who themselves are self-interested in maximizing licensing settlements and money judgments to justify a return on their time.

The goal of the patent system is to promote science and invention by granting an exclusive right to inventors. If the patent system fails to protect these startup companies hampered by many outside factors besides the quality of their invention itself, then the patent system is flawed and fails to incentivize the very companies it was created for. For issued patents, the cat is already out of the bag so to speak—little can be done about granted rights. The best way to fix the system moving forward, however, is to weed out low-quality patents prior to issue. This would force patent seekers to engage in targeted, quality prosecution of only those patents meriting allowance and prevent spurious patents from issuing and wreaking havoc in the marketplace, which undermines the remaining patents of merit and leads to calls for further reform in the post-issuance regime. Given the complexity of modern software, many patent attorneys try to broaden the scope of claims—as is their job—and examiners have difficulty allowing only the proper and exact claims, particularly given constraints on their time. Allowing overly broad claims by the tens of thousands of applications results in software patent thickets; hence, it drives up the chance of patent infringement lawsuits, patent aggregation entities licensing portfolios of tens of thousands of patents, and licensing requests for non-inventive patent thickets.

191. Id.
192. See id. (discussing the significant operational impacts of NPE litigation on small companies).
B. Software Patent Thickets

A patent thicket is “an overlapping set of patent rights requiring that those seeking to commercialize new technology obtain licenses from multiple patentees.”195 Shane Anderson argues that “[t]hickets cause undue hold-up at firms, increase the cost of entry for new players, increase the cost of Research and Development, and suppress innovation.”196 Anderson’s article focuses its attention on Research and Development (R&D) spending.197 R&D spending is an important indicator of the worth of a company’s software patent because it shows the willingness of a company to invest in research and development of its software product. R&D is commensurate with the Intellectual Property Clause in a sense because R&D promotes science and technology. However, some scholars assert that thickets are natural and not particularly concerning. By this they mean that no correlation exists between software patent spending and R&D expenditures.198 Rather, they argue that “software R&D spending tends to be relatively stable over time as a percentage of sales” and that “company size seems to be more important in explaining variations in R&D spending within the industry.”199 Anderson, however, points to the uniqueness of the software industry to explain that an increase in patent thickets reduces R&D spending.200 He asserts that “[s]oftware firms are sensitive to delays, and the first-mover market advantage gained by a firm is extremely important . . . .”201 Consequently, this advantage “is often a sufficient incentive by itself to promote R&D.”202

Bessen’s study lends support to a causal link that patent thickets cause less exclusive rights for software inventors, and this assumption leads to more patent infringement and less R&D spending by software developers.203 Even though this causal link may seem plausible, many other factors, such as cross-licensing, company size, timing, and profits may explain R&D

195. Anderson, supra note 5, at 577 (quoting Shapiro, supra note 6, at 119).
196. Id.
197. See id. at 576–77.
199. Id. at 1003.
200. See Anderson, supra note 5, at 577.
201. Id.
202. Id.
203. See Bessen, supra note 8, at 248.
spending behavior. Nonetheless, scholars agree that patent thickets exist and are troublesome for the software industry.

What causes patent thickets? For one, software patents, by their very nature, generally have broad, ambiguous claims and thus do not give clear notice of either the claimed invention or the patentee’s rights. The USPTO does not require practicable source code or prototypes of invention, and the enablement requirements have been relaxed for decades; thus, specifications themselves can be relatively generic, particularly regarding hardware or the algorithms required to practice the applied-for claims. As the patent owners can be their own lexicographers, and there are few agreed upon terms for certain programming actions, it can be difficult to perform semantic or Boolean searching on existing prior art. Fittingly, the existing patent thickets make it even harder to find one tree in the forest, so complex but known concepts under different names often slip through the examination corps. It is simply unrealistic to think that any examiner can search through hundreds of thousands of issued software patents, understand methods and abstract concepts, and apply them efficiently.

Broad and ambiguous claims increase costs imposed by patent thickets because they increase the chance of a software developer paying the cost of either licensing or patent infringement litigation that may not even cover the software created by the developer. This self-fulfilling cycle again makes it more difficult for examiners to identify known or explored concepts when faced with a record number of applications and grants. Such broad and ambiguous claims monopolize what is not included in the actual invention by preempting other inventors from obtaining patents.


205. Anderson, supra note 5, at 577–78.


207. See Nautilus, Inc. v. Biosig Instruments, Inc., 134 S. Ct. 2120, 2129 (2014) (citing Hormone Research Found., Inc. v. Genentech, Inc., 904 F.2d 1558, 1563 (Fed. Cir. 1990) (“It is a well-established axiom in patent law that a patentee is free to be his or her own lexicographer . . . .”)).

208. Anderson, supra note 5, at 578.
For example, Amazon’s “one-click” patent spurred much debate among scholars. To help capture customers abandoning their shopping carts due to whatever reason, “Amazon developed an online purchasing method where shoppers could complete a buying transaction with a single mouse click.” On September 28, 1999, the USPTO issued the patent to Amazon that claimed “[an] online purchasing system where customers enter their credit card number and billing address information on their initial visit to the Amazon.com website.” The claims further specified that “this personal information is stored [in] Amazon’s database system and when a customer returns to the website at a later point in time to make purchases, the customer will not have to reenter their personal information to make an online purchase.” Through such system, returning customers are able to purchase “with a single click of the mouse.” Around the same time, the Barnes & Noble website provided an online purchasing system called “Express Lane,” which used a similar method of online purchase as Amazon’s “one-click” patent. The Federal Circuit essentially granted injunctive relief for Amazon, and “Barnes & Noble was required to resort to the traditional shopping cart method of online transactions.” Consequently, being the only entity holding a patent for completing online transactions in a single click of the mouse, Amazon has economically benefited by excluding others from using its patented method.

Many scholars point out that “the thicket problem arises from the flood of patents that are granted by the USPTO . . . .” As mentioned in the example above, the USPTO’s imprudent issuance of the “one-click” patent contributes to the “thicket problem” and bestows more than enough monopoly power to a single entity. A study conducted prior to the Alice decision found that the number of software patents granted per year was:

210. Id. at 96.
211. Id.
212. Id.
213. Id.
214. See id.
215. Id. at 96–97.
216. Id. at 97.
218. See HALT ET AL., supra note 163, at 96–97 (discussing the monopoly that “one-click” has given Amazon).
Increasing.\textsuperscript{219} Professor Mark Lemley also points out that the USPTO has been issuing “a large number of dubious patents over the past 20 years, particularly in the software and electronic commerce space.”\textsuperscript{220} He further notes that the USPTO has been “issuing many more patents than its counterparts in Europe and Japan,” granting roughly one or more patents to three-fourths of applicants.\textsuperscript{221}

Compounding the problem, low-quality patents are difficult to overturn or invalidate because courts require defendants to provide “clear and convincing evidence” to invalidate issued patents, presuming that the USPTO has already done a good job of screening out bad patents.\textsuperscript{222} Intuitively, having more patents results in a higher probability of overlaps in claims. Thus, a natural solution is for “the USPTO to grant fewer software patents and diminish the growing thicket.”\textsuperscript{223} However, Lemley disagrees that the problem lies in the USPTO issuing many bad patents.\textsuperscript{224} Rather, Lemley argues that the problem arises from the USPTO issuing “a small but worrisome number of economically significant bad patents and those patents enjoy a strong but undeserved presumption of validity.”\textsuperscript{225}

Low-quality patents have significant social costs and must be eliminated to maintain a healthy patent system. This was at the core of the \textit{Alice} decision. Looking at the hard-to-grasp and growing number of issued patents and the growing number of frivolous patent infringement suits in software and business method patents, the problem has grown worse, if not better, in \textit{Alice}’s wake. However, the Supreme Court did not iterate a clear test for the lower courts,\textsuperscript{226} nor has the Federal Circuit been able to reach a clear and consistent ruling on Section 101 after \textit{Alice}.\textsuperscript{227} While the USPTO has

\begin{itemize}
\item \textsuperscript{219} See Bessen, supra note 8, at 252.
\item \textsuperscript{220} Mark A. Lemley, \textit{Fixing the Patent Office}, 13 \textit{INNOVATION POL’Y & ECON.} 83, 83 (2013).
\item \textsuperscript{221} \textit{Id.}
\item \textsuperscript{222} \textit{Id.} at 84.
\item \textsuperscript{223} Anderson, supra note 5, at 578.
\item \textsuperscript{224} Lemley, supra note 220, at 84.
\item \textsuperscript{225} \textit{Id.} (citing Mark Lemley et al., \textit{What to Do About Bad Patents}, \textit{REGULATION}, Winter 2005–2006, at 10, 12).
\item \textsuperscript{226} See Alice Corp. Pty. Ltd. v. CLS Bank Int’l, 134 S. Ct. 2347, 2355–57 (2014) (neglecting to define an abstract idea).
\item \textsuperscript{227} See, e.g., Chamberlain Grp. Inc. v. Linear LLC, 114 F. Supp. 3d 614 (N.D. Ill. 2015) (holding an Alarm System Patent had other additional commands and inputs that made it patent-eligible).
\end{itemize}
programs in place meant to address low-quality patents, patents continue to issue in record numbers, suggesting that at least some patent owners are still receiving allowances. Furthermore, the USPTO’s significant increase in the rejection rate for software patents under Section 101 evidences applicant confusion, though it has not slowed the overall rate of patent grants. So, where do we find the solution to fix this software patent thicket problem? As Lemley and other scholars point out, the core of this solution lies in the USPTO’s examination process.

IV. A ROADMAP FOR THE USPTO

A. Patent Claim Value and Solutions to Low-Quality Patent Claims

A fundamental principle of traditional patent law theory is that a patentee’s right to exclude others benefits the patentee and is needed to make the patent system work. The quid pro quo of the information disclosures required by patent law and the exclusive rights given to patentees are critical to creating significant informational value to the patent owner and to the public. Ideally, patents with fulsome disclosures that teach clear boundaries of inventions to the public are vital to warn potential infringers of their limits. Patent claims are “the touchstone of patent protection” and set forth the patentee’s proprietary boundaries. Thus, the patent examiner’s job, shaping those boundaries by limiting insignificant claims during the examination process, is crucial to solving software patent thicket problems and to potentially reducing infringement suits brought by non-practicing entities (NPE).

Anderson argues that “[w]ithout the requisite institutional and legal tools, it is virtually inevitable that large swaths of low quality software patents will find their way out the USPTO’s door.” More specifically, Lemley lays out possible reconstruction of the USPTO’s examination

228. See Patent Quality, USPTO, https://www.uspto.gov/patent/patent-quality (last visited Oct. 27, 2017) (explaining the three areas the USPTO has focused on to ensure patent quality).
230. See infra APPENDIX C (showing the rejection rate after the Alice decision).
231. Asay, supra note 4, at 259 (discussing the value of patents).
232. Id. at 275.
234. Anderson, supra note 5, at 577.
process to fix the “bad patents” problem.\textsuperscript{235} And, in 2016, scholars Melissa F. Wasserman and Michael D. Frakes demonstrated that the first year of examination—and subsequent examining behavior—can lead to substantive outcomes.\textsuperscript{236}

Adopting some of their views, we propose the following three solutions. First, as Lemley suggests, one solution to bad patent allowances is to adopt a mandatory second-pair-of-eyes process similar to past efforts.\textsuperscript{237} This means subjecting patent applications to mandatory assessment by more than one examiner before approval.\textsuperscript{238} This will capture more mistakes, ensure quality control, and will work to prevent tunnel vision or rogue examiners. Second, the USPTO needs more examiners with proper training and increased incentives for examining—and in many cases, not allowing—software patent applications. They should not only be equipped with engineering or science knowledge, but also with legal knowledge to correctly apply legal standards to patent applications. For example, instead of bluntly rejecting software patent applications under Section 101, examiners need to conduct a careful examination to differentiate accurate claims from those ambiguous and broad ones following the Federal Circuit decisions—and pay equal attention to disclosure and Section 112 problems. Third, the request for continued examination (RCE) must be tailored to prevent meritless patent claims. Patent applicants are currently allowed to file an unlimited number of RCEs, which are requests by an applicant to reopen examination of the patent application after the prosecution of the application has closed.\textsuperscript{239} As a result of applicants possibly filing several RCEs, worn-out examiners may end up allowing meritless claims to determined applicants due to repetitive and never-ending work. The proposal for limiting the number of RCEs under the former Dudas Administration had merit and should be revisited.\textsuperscript{240}

\begin{thebibliography}{99}
\bibitem{Lemley} See generally Lemley, supra note 220 (examining solutions to fix the USPTO).
\bibitem{FrakesWasserman} Michael D. Frakes & Melissa F. Wasserman, Patent Office Cohorts, 65 Duke L.J. 1601, 1603 (2016) (stating “the year an examiner was hired may help explain her granting proclivities”).
\bibitem{Lemley} Lemley, supra note 220, at 91.
\end{thebibliography}
would likewise behoove the administration to limit or scale the use of RCEs through their fee-setting authority or other “soft” means to influence or “nudge” behavior. The following sections will discuss these proposals in further detail.

B. Second-Pair-of-Eyes

In 1998, when the Federal Circuit held business methods patentable, the USPTO was inundated with business method patent applications. In response, the USPTO initiated a specific “quality control” measure, the “second-pair-of-eyes” review (SPER), under which applications were subjected to “mandatory assessment by more than one examiner before being allowed.” A study found that SPER had a dramatic effect: the lowest grant rate among high-volume classes. This grant rate reflected “better rigor during examinations rather than application volume.” But SPER has its drawbacks; in theory, it would increase the cost of prosecution and would delay the process. These drawbacks would be exacerbated in the software context, where the grant rate is already at its lowest (though the application rate is still extremely high, with many companies engaging in the “spray-and-pray” approach), and such a burden on the cost of prosecution would then turn into higher application fees. This would essentially burden startups with higher prosecution costs upfront.

Nonetheless, SPER is a worthwhile practice to be revisited at the USPTO because it would significantly reduce the patent thicket in software. The Supreme Court’s *Alice* decision and the following Federal Circuit decisions mainly stemmed from software patent thickets. The meritless claims granted by the USPTO spurred an increasing number of infringement suits with which non-practicing entities, or patent trolls, emerged to make


242. See Lemley, supra note 220, at 91.

243. *Id.*

244. *Id.*

245. *Id.* at 91–92.

benefits off of these broad and ambiguous claims. Although SPER may shift the burden on the USPTO in terms of cost and time consumption, it is likely to weed out unpatentable claims and, at the same time, catch some good ones within the net of rejected applications. Consider this: if appealed all the way to the Federal Circuit, a denial of an application can be seen by up to seventeen pairs of eyes before finally being rejected after an appeal. In most instances, current allowances are seen by one. In terms of basic balance, it seems reasonable to ask for a more searching review of claims before they are allowed. Anecdotally, Mr. Stroud experienced some informal SPER practices as an examiner, with good results. A second opinion is almost never a bad thing, and the costs at that stage are marginal, as opposed to the expensive, time-consuming, and ultimately inefficient second look performed by the courts or the PTAB.

One concern is that SPER is useless if examiners are not properly trained to review claims in patent applications, or at worst would increase cost and delay without affecting quality. These concerns can be mitigated with proper training at the USPTO. Modern technology advances at breakneck speed. To keep up with this rapid change, examiners must be knowledgeable about the trends and detailed workings of new inventions and must have the support of others. The current requirement to be an examiner at the USPTO is a bachelor’s degree in engineering or another specified science background. Comparatively, many private companies employ people with a Ph.D. or master’s degree to research and develop their products and work with patent attorneys to apply for patents. Even though a higher educational background is not the sole indicator that a person is sufficiently knowledgeable in the technical area, the USPTO must provide relevant and proper training for examiners and seek to hire and incentivize high-quality, technically trained examiners to close the gap, so that examiners can understand the specifications and claims recited in the applications.


C. Setting Fees and a Call for Consistent Guidance

When the Class 705, a technology area that reviews business method applications, adopted SPER in 2001 to initiate quality control, it created an enormous backlog of patent applications as examiners would reject applications, and applicants would attempt to continue examination through other mechanisms. As discussed in the first proposal, SPER may lead to a backlog, so there must be another mechanism to prevent the examination process from slowing down while still enforcing effective quality control measures.

A simple and intuitive solution to relieve the enormous backlog is to hire more examiners and to reasonably limit RCE practice, as outlined below. Training may take new hires some time to get up to speed on the application examination, but it would surely add more workforce to speed up the examination process. Some argue that the USPTO itself is overworked and underfunded. According to GAO’s interview with patent examiners, examiners said that time pressures on examiners are a central challenge for patent quality. Based on GAO’s survey of patent examiners, GAO estimated that 70% of the examiners say they do not have enough time to complete a thorough examination given a typical workload. If more examiners are available to do the work, then each examiner can take on a fewer number of applications for a more thorough examination. This could prevent the examination process from slowing down and could serve as one possible solution to the backlog problem while maintaining patent quality.

But, as some note, an increase in the number of employees simply means more expenses for the USPTO, and such increases in expenses can mean shifting its burden on the applicants and patentees to collect more fees. Lemley argues against spending more money to pay for a more intensive examination, finding that “[m]ost of that money will be wasted on applications that are of no consequence to anyone.” We disagree. That is the prerogative of companies seeking meritless or questionable claims; rigorous examination should not retard allowance of truly innovative patent claims. If the fees can be collected and spent constructively, then they will serve as a good weapon to improve patent quality. Furthermore, they will

---

249. See Lemley, supra note 220, at 88.
251. GAO-16-490, supra note 7, at 25.
252. Id. at 25–26.
253. See Lemley, supra note 220, at 88–89.
254. Id. at 86.
limit RCEs and issued patents. Making fee-setting authority permanent would clearly contribute to the USPTO’s ability to properly address the issue.

Under the America Invents Act, the USPTO is authorized to set fees.\textsuperscript{255} For example, the USPTO can “charge substantial maintenance fees for existing patents even though maintaining a patent once granted costs the [USPTO] nothing.”\textsuperscript{256} One counter is that the USPTO raising maintenance fees may cause patentees to abandon their patents altogether, regardless of whether it is a good patent or bad patent.\textsuperscript{257} But for patents with merit, a higher cost should be offset by licensing and product development of operating companies. Given the lifecycle of innovation versus patents, few startups will be confronted with paying maintenance fees, and conversely, patent aggregators or NPEs simply holding or asserting assets would be forced to make harder decisions over which patents to keep (i.e., those with merit) and which to abandon (i.e., meritless patents).

In theory, higher application fees could deter startups from applying for patents. We agree. But the USPTO could and should further raise requests for reexamination fees to disincentivize endless examination. Furthermore, the USPTO should raise reissue and reexamination fees because a patentee who wants to redefine the boundary of his already issued patents will likely do so for strategic business reasons and will likely pay a higher price for it. The USPTO could also raise fees in the other areas to recoup its costs accordingly to allow for a more thorough examination, reexamination, or review, while not deterring inventors and patentees from filing or maintaining their patents. Even if the USPTO raises fees in other areas and burdens small entities, if the resulting patent system guarantees clear boundaries of the claims, and thus brings down the chance of getting sued or involved in litigation, then investors will likely mark that money down for what it is worth. The only party to truly lose in that situation would be the well-paid patent bar. The question is whether the USPTO can spend the money constructively to improve patent quality.

Some voice concerns that “the USPTO simply does not staff examiners with the knowledge required to parse software patents.”\textsuperscript{258} While the authors

\textsuperscript{255} Id. at 85.
\textsuperscript{256} Id.
\textsuperscript{257} Id. at 86.
\textsuperscript{258} Anderson, supra note 5, at 577; cf. Kevin M. Baird, Business Method Patents: Chaos at the USPTO or Business as Usual?, 2001 U. ILL. J.L. TECH. & POL’Y 347, 355 (2001) (stating that the USPTO had increased difficulty with business method patents as there were few examiners with the appropriate background).
have found that the USPTO employs thousands of highly skilled and trained examiners and has done its best with an unwieldy number of applications and examiners, investing in proper training for examiners can help address the above issue. Another way to approach this issue is to find better ways to measure patent quality. It is currently difficult to judge what is a good or bad patent given market concern, legal flux, and other considerations. Nonetheless, the USPTO does not currently have a consistent definition for patent quality articulated in agency documents and guidance, which would be in line with best practices for organizational performance. Thus, the USPTO is unable to fully measure progress toward meeting its patent quality goals. If examiners are equipped with such guidelines containing clear definitions and can apply those definitions on application claims, there will be a more coherent and consistent examination process.

D. Request for Continued Examination Practice

One of the biggest contributors to pendency times and eventual grants of questionable claims is the so-called request for continued examination (RCE) practice. An RCE is “a request by an applicant to reopen examination of the patent application after the prosecution of the application has been closed.” To obtain an RCE, an applicant is only required to file a new submission and pay the additional fees. Surprisingly, there is no limit on the number of times an applicant may file an RCE. Because USPTO rules and procedures do not restrict the number of RCEs an applicant may file, issuing the patent is the only means by which the USPTO can end the examination process. This creates an environment where issued patents do not fully meet patentability standards because such unlimited attempts to secure a patent can wear down examiners, making them more likely to eventually grant the patent. Thus, the commonplace adage: “keep a continuation alive.”

259. See GAO-13-465, supra note 190, at 41–43 (discussing issues with patent quality without any clear definition for patent quality).
260. Sean Tu, Understanding the Backlog Problems Associated with Requests for Continued Examination Practice, 13 DUKE L. & TECH. REV. 216, 218–19 (2014–2015); see also GAO-16-490, supra note 7, at 32 (stating that unlimited RCE requests may wear down examiners and eventually make them more likely to grant a patent).
261. Sachs, Busting Myth, supra note 239 (citing GAO-16-490, supra note 7, at 31).
262. GAO-16-490, supra note 7, at 8.
263. Id. at 8.
264. Id. at 27–28.
265. GAO-16-490, supra note 7, at 32.
Some contend that “the problem of ‘unlimited’ RCEs is a myth.” While admitting that the patent statute does not limit the number of RCEs, Robert Sachs argues that “the reality of patent examination and prosecution yields a negligible number of high count RCE patents,” and that “they have no impact on the patent litigation landscape.” His study found that the amount of litigation resulting from RCE patents was not significantly more than the amount of litigation resulting from his control group of random patent samples. From his findings, he posits that because “more RCEs typically means more office actions, which in turn implies more amendments . . . as the applicant attempts to narrow the claims,” high-count RCE patents themselves do not contribute to patent litigation.

Sachs’s argument is unconvincing as he bases his study on RCE patents that should not have been granted in the first place. The entire sum of patent litigation that results from RCE patents could be avoided if they were not issued in the first place. Clearly, it cannot and should not be said that all RCE patents are low-quality. However, many of them may contain claims that add no value to our patent system or claims that respond directly to later-developed market competitors rather than the original invention. If the USPTO implements rules and procedures to limit the number of RCEs, or to steeply increase the cost of each subsequent request, the amount of patent litigation can decrease accordingly, reducing the number of RCE patents that issue. Therefore, limiting the number of RCE attempts will clear dockets and reinvest examiners’ energy to other patent applications, and thus use internal resources more efficiently.

V. CONCLUSION

Software patent applications are among the most complex, abstract patents that the USPTO has to examine. The above proposals do not directly confront the ambiguous two-prong test in the Alice decision—nor should that be the USPTO’s job—but they may serve as tools for improving patent claim quality. Higher quality valid issued patent claims will help the courts to pay deference to the USPTO’s decisions; avoid post-grant review; and later bodies and courts will be more likely to uphold the validity and eligibility of patent claims. Furthermore, fewer low-quality patent claims would decrease patent litigation lawsuits because litigants would be deterred.

266. See Sachs, Busting Myth, supra note 239.
267. Id.
268. See infra APPENDIX D (displaying a graph of issued patents).
269. Sachs, Busting Myth, supra note 239.
from fighting over validity and patentability. These suggestions will not end the patent troll problem. These entities will continue to collect patents to sue and license others for patent infringement. But the problem does not lie in the fact that patent trolls sue or threaten to sue others to collect licensing fees; rather, it is the ready availability of both issued and purchased broad and ambiguous software patent claims that can be asserted for broader exclusivity than the actual invention covers. If the USPTO would issue higher quality software patent claims that the PTAB and courts could uphold as largely valid, patentable, and eligible, more companies would likely turn to licensing those higher quality patent claims rather than taking their chances in courts—conversely, making the jobs of assertors with high-quality patents much easier by settling rights and expectations before incurring further transaction costs.

There are many approaches to protecting software. Nonetheless, the consensus among many scholars and practitioners is that courts need to clarify and give meaningful guidance for the USPTO in issuing Section 101 rulings. The above proposals will not solve our patent problems, but they may improve the quality of future-issued software patents and, in essence, stabilize our patent system by setting a higher standard for issuing patent claims. Establishing a more stable system would help startup companies attain more meaningful patent portfolios to conduct their businesses and seek out investors. To begin to deal with this cavalcade of problems, the first step is to weed out low-quality patent claims.
APPENDIX


<table>
<thead>
<tr>
<th>Percentage of Patent Ineligibility</th>
<th>Total</th>
<th>Total Ineligible Under § 101</th>
<th>% Ineligible</th>
</tr>
</thead>
<tbody>
<tr>
<td>§ 101 Decisions</td>
<td>405</td>
<td>274</td>
<td>67.7%</td>
</tr>
<tr>
<td>Federal Circuit</td>
<td>73</td>
<td>65</td>
<td>89.0%</td>
</tr>
<tr>
<td>District Courts</td>
<td>332</td>
<td>209</td>
<td>63.0%</td>
</tr>
</tbody>
</table>

B. Software Patents Challenged Under 35 U.S.C. § 101

### C. Section 101 Rejection Rate Post-Alice by Art Unit 272

<table>
<thead>
<tr>
<th>Tech Type</th>
<th>Art Unit Tech</th>
<th>Before Alice</th>
<th>6/14 Preliminary</th>
<th>12/14 Interim</th>
<th>7/15 Update</th>
<th>5/16 Examiner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biotech</td>
<td>Chemical Engineering</td>
<td>3.1%</td>
<td>4.5%</td>
<td>3.9%</td>
<td>4.2%</td>
<td>4.3%</td>
</tr>
<tr>
<td></td>
<td>Immunology</td>
<td>12.1%</td>
<td>19.3%</td>
<td>17.3%</td>
<td>13.3%</td>
<td>12.5%</td>
</tr>
<tr>
<td></td>
<td>Microbiology</td>
<td>12.7%</td>
<td>21.2%</td>
<td>19.5%</td>
<td>14.8%</td>
<td>12.7%</td>
</tr>
<tr>
<td>Computers</td>
<td>Molecular Biology</td>
<td>15.3%</td>
<td>20.7%</td>
<td>20.5%</td>
<td>18.8%</td>
<td>17.4%</td>
</tr>
<tr>
<td></td>
<td>Communications &amp; Networking</td>
<td>8.7%</td>
<td>8.8%</td>
<td>11.5%</td>
<td>14.4%</td>
<td>12.6%</td>
</tr>
<tr>
<td></td>
<td>Computer Engineering</td>
<td>5.5%</td>
<td>7.2%</td>
<td>5.3%</td>
<td>4.7%</td>
<td>4.4%</td>
</tr>
<tr>
<td></td>
<td>Computer Science &amp; Applications</td>
<td>16.6%</td>
<td>16.9%</td>
<td>16.3%</td>
<td>14.8%</td>
<td>13.2%</td>
</tr>
<tr>
<td></td>
<td>Cryptography</td>
<td>15.5%</td>
<td>15.8%</td>
<td>21.1%</td>
<td>21.9%</td>
<td>15.5%</td>
</tr>
<tr>
<td>Ecommerce</td>
<td>Business Crypto</td>
<td>42.1%</td>
<td>60.4%</td>
<td>64.5%</td>
<td>74.4%</td>
<td>74.9%</td>
</tr>
<tr>
<td></td>
<td>Business Processing</td>
<td>36.1%</td>
<td>83.7%</td>
<td>94.0%</td>
<td>89.2%</td>
<td>94.0%</td>
</tr>
<tr>
<td></td>
<td>Cost/Price Reservations</td>
<td>33.5%</td>
<td>68.9%</td>
<td>84.0%</td>
<td>85.0%</td>
<td>85.0%</td>
</tr>
<tr>
<td></td>
<td>E-Shopping</td>
<td>37.4%</td>
<td>83.4%</td>
<td>93.8%</td>
<td>82.5%</td>
<td>82.5%</td>
</tr>
<tr>
<td></td>
<td>Finance &amp; Banking</td>
<td>31.0%</td>
<td>74.2%</td>
<td>73.3%</td>
<td>91.0%</td>
<td>93.0%</td>
</tr>
<tr>
<td></td>
<td>Health Care</td>
<td>33.1%</td>
<td>78.1%</td>
<td>98.8%</td>
<td>85.6%</td>
<td>93.6%</td>
</tr>
<tr>
<td></td>
<td>Incentive Programs</td>
<td>39.4%</td>
<td>83.9%</td>
<td>91.4%</td>
<td>88.3%</td>
<td>85.7%</td>
</tr>
<tr>
<td></td>
<td>Operations Research</td>
<td>32.3%</td>
<td>83.3%</td>
<td>93.2%</td>
<td>95.3%</td>
<td>91.3%</td>
</tr>
<tr>
<td></td>
<td>PTO, Inventy, Accounting</td>
<td>29.9%</td>
<td>69.5%</td>
<td>67.4%</td>
<td>68.3%</td>
<td>74.9%</td>
</tr>
<tr>
<td>Other</td>
<td>Agriculture</td>
<td>21.1%</td>
<td>16.8%</td>
<td>15.5%</td>
<td>13.6%</td>
<td>14.3%</td>
</tr>
<tr>
<td></td>
<td>Amusement &amp; Education</td>
<td>13.8%</td>
<td>29.9%</td>
<td>36.1%</td>
<td>31.4%</td>
<td>25.9%</td>
</tr>
<tr>
<td></td>
<td>Chemical Engineering</td>
<td>1.5%</td>
<td>1.5%</td>
<td>1.5%</td>
<td>2.2%</td>
<td>3.9%</td>
</tr>
<tr>
<td></td>
<td>Civil Engineering</td>
<td>2.1%</td>
<td>2.7%</td>
<td>2.6%</td>
<td>2.6%</td>
<td>2.1%</td>
</tr>
<tr>
<td></td>
<td>Communications &amp; Networking</td>
<td>0.0%</td>
<td>5.6%</td>
<td>7.1%</td>
<td>6.9%</td>
<td>7.5%</td>
</tr>
<tr>
<td></td>
<td>Electrical Components, Devices &amp; Sys...</td>
<td>2.6%</td>
<td>4.1%</td>
<td>4.6%</td>
<td>4.2%</td>
<td>5.1%</td>
</tr>
<tr>
<td></td>
<td>Energy &amp; Power</td>
<td>1.0%</td>
<td>2.2%</td>
<td>3.2%</td>
<td>2.3%</td>
<td>2.2%</td>
</tr>
<tr>
<td></td>
<td>Manufacturing</td>
<td>1.0%</td>
<td>1.1%</td>
<td>1.1%</td>
<td>0.7%</td>
<td>0.6%</td>
</tr>
<tr>
<td></td>
<td>Medicine</td>
<td>4.1%</td>
<td>5.8%</td>
<td>7.1%</td>
<td>6.3%</td>
<td>6.0%</td>
</tr>
<tr>
<td></td>
<td>Physics</td>
<td>1.3%</td>
<td>1.3%</td>
<td>1.8%</td>
<td>1.3%</td>
<td>1.7%</td>
</tr>
<tr>
<td></td>
<td>Semiconductors</td>
<td>0.3%</td>
<td>1.2%</td>
<td>1.7%</td>
<td>0.4%</td>
<td>0.2%</td>
</tr>
<tr>
<td></td>
<td>Signal Processing</td>
<td>9.8%</td>
<td>9.5%</td>
<td>11.2%</td>
<td>10.2%</td>
<td>11.0%</td>
</tr>
<tr>
<td></td>
<td>Transportation</td>
<td>7.5%</td>
<td>9.6%</td>
<td>7.9%</td>
<td>8.0%</td>
<td>7.9%</td>
</tr>
<tr>
<td>Grand Total</td>
<td></td>
<td>12.1%</td>
<td>10.1%</td>
<td>11.5%</td>
<td>10.5%</td>
<td>10.6%</td>
</tr>
</tbody>
</table>

272. Note: Rejection rate at the USPTO by technology type and art unit before and after the Alice decision. Source: [http://www.bilskiblog.com](http://www.bilskiblog.com).
D. Percentages of Issued Patents by Number of RCEs

Figure 1: Percentages of Issued Patents by Number of RCEs