BUFFALO LAW REVIEW

VOLUME 60 JANUARY 2012 NUMBER 1

Taxing Facebook Code: Debugging the Tax Code and Software

XUAN-THAO NGUYEN†
JEFFREY A. MAINE††

INTRODUCTION

In the movie *The Social Network*, the character Eduardo Saverin writes the pivotal code for the software behind Facemash, the precursor of Facebook.[1] Left on the window of Saverin's Harvard dorm room,[2] this code became the center of the Facebook saga, in both the movie and real life. Saverin, as well as Cameron and Tyler Winklevoss, who later sued Mark Zuckerberg for stealing their code,[3] have all attempted to lay claim to the ownership of Facebook.

What is Facebook and what does code have to do with it? Facebook is a social network site where people take

---

†Professor of Law, SMU Dedman School of Law; former IP Associate, Fried Frank Harris Shriver & Jacobson (NYC); Pryor Cashman Sherman & Flynn (NYC). This Article was made possible with a summer grant from SMU Dedman School of Law. Professor Nguyen thanks her co-author Professor Jeffrey A. Maine for the joy of working together on the intersection of intellectual property and taxation. Special thanks to Erik Darwin Hille and Khai-Leif Nguyen-Hille for their love, patience, and support.

††Maine Law Foundation Professor of Law, University of Maine School of Law; former Tax Associate, Holland & Night (Tampa). Professor Maine owes special thanks to Professor Xuan-Thao Nguyen for many collaborations on the taxation of intellectual property. He also wishes to thank Julie Welch and Tudor Goldsmith for their invaluable research assistance.

1. *See* THE SOCIAL NETWORK (Columbia Pictures 2010), *available at* http://www.youtube.com/watch?v=BzZRr4KV59I&feature=watch_response ("The Algorithm Scene").

2. *Id.*

3. *See generally* Facebook, Inc. v. Pac. Nw. Software, Inc., 640 F.3d 1034 (9th Cir. 2011).

control of the Internet.[4] Unlike many earlier Internet applications, Facebook discourages anonymity; instead, Facebook blurs the line between life online and offline[5] by encouraging its users to let the world see them as their friends see them, by posting pictures and personal information, and by creating a network of "friends" viewable to the public.[6]

"Friending"[7] needs code—and lots of it.[8] Structurally, Facebook is all code, as thousands of Facebook engineers work endless hours designing, constructing, writing, integrating, debugging, and testing code to implement their

---

4. *See* Lev Grossman, *Person of the Year 2010: Mark Zuckerberg*, TIME, Dec. 27, 2010, at 49 ("[Mark Zuckerberg] started Facebook as a way for people on college campuses to communicate with and keep track of one another—and occasionally poke each other and leer at each other's pictures—but in a broader sense he was firing the first shot in his generation's takeover of the Internet.").

5. *Id.* at 52. As Grossman explained:

> [Facebook] grew because it gave people something they wanted. All that stuff that the Internet enabled you to leave behind, all the trappings of ordinary bourgeois existence—your job, your family, your background? On Facebook, you take it with you. It's who you are.
>
> Zuckerberg has retrofitted the Internet's idealistic 1960s-era infrastructure with a more pragmatic millennial sensibility. Anonymity may allow people to reveal their true selves, but maybe our true selves aren't our best selves. Facebook makes cyberspace more like the real world: dull but civilized. The masked-ball period of the Internet is ending. Where people led double lives, real and virtual, now they lead single ones again.
>
> The fact that people yearned not to be liberated from their daily lives but to be more deeply embedded in them is an extraordinary insight, as basic and era-defining in its way as Jobs' realization that people prefer a graphical desktop to a command line or pretty computers to boring beige ones.

Id.

6. *See id.* at 49, 52.

7. Facebook's powerful friending idea has attracted much private funding from hungry investors. *See* Editorial, *Friending Private Capital*, WALL ST. J., Jan. 5, 2011, at A14.

8. *See* Qiang Wu, *Keeping the Site Reliable While Moving Fast*, FACEBOOK (Sept. 6, 2011, 12:26 PM), http://www.facebook.com/note.php?note_id=10150277682538920 ("Moving fast is core to Facebook's culture and is key to keeping us innovative. Shipping code every day means our software infrastructure must be extremely dynamic.").

ideas for all aspects related to social networking.[9] These engineers are continually coming up with new code because Facebook's development cycle is, by its very nature, innovation-based, seeking to respond promptly to market demand.[10] Each piece of code is a building block of software.

Few analysts and policymakers disagree over the social benefits provided by Facebook and similar companies that rely on software to build and maintain their business models. Disagreement, however, often arises over the proper means of stimulating software development for the benefit of society. Instead of relying solely upon direct research subsidies, such as grants and subsidized loans, the government relies heavily on private spending on software innovation for the public good.[11] Private investment in software research is stimulated chiefly by a system of intellectual property laws for software innovation, as well as by indirect government subsidies in the form of software tax benefits.[12] The government hopes that the benefit to society from the added software investment will exceed the social costs of that investment.[13]

The most important policy issues surrounding software concern the intersection of intellectual property and tax policy. To that end, this Article sets out to analyze both intellectual property laws and tax systems as applied to software. The Article begins, in Part I, by highlighting the important role of software in many firms today.

Part II analyzes software within intellectual property's established doctrinal framework, a difficult task due to the fact that software can encompass some combination of the traits of copyrights, trade dress, patents, and trade secrets.

---

9. *See How Facebook Ships Code*, FRAMETHINK (Jan. 17, 2011), http://framethink.wordpress.com/2011/01/17/how-facebook-ships-code/ (stating that fifty percent of Facebook's employees are engineers).

10. *See generally id.* (describing Facebook's process for updating code and noting that code changes are implemented on a weekly basis).

11. *See* Linda R. Cohen & Roger G. Noll, *Is U.S. Science Policy at Risk?: Trends in Federal Support for R&D*, BROOKINGS INST. (Winter 2001), http://www.brookings.edu/articles/2001/winter_technology_cohen.aspx (noting the decrease in federal spending, but increase in private spending, on research and development).

12. *See id*; *see also infra* Parts II, III.

13. *See infra* Part IV.B.

Part III of the Article examines both the federal and state tax systems governing software. As this Part shows, fitting software within current tax schemes presents unique challenges, as software contains both tangible and intangible elements, is subject to varying intellectual property protections, and is rapidly emerging into different products due to technological innovation. Part III also highlights certain tax preferences for software innovation.

Part IV critiques the current tax approaches to software. It argues that there are a number of incongruous tax distinctions for software that are theoretically and analytically unsatisfactory. Specifically, these tax distinctions run counter to the sound tax principles of fairness and efficiency, suggesting legislative changes may be warranted. Accepting that software innovation provides important spillover effects and external benefits, this Part also points out certain flaws in the design of several tax preferences for software.

I. FACEBOOK, GROUPON, CLOUD COMPUTING—SOFTWARE BY DIFFERENT NAMES

When we think of "software," we generally think of things like Microsoft Word, McAfee Antivirus, or Adobe Photoshop. But the world of "software" is not confined to just the programs we can buy in a store and install on our personal computers; software is everywhere, sometimes hiding in plain sight.

For example, when someone mentions Facebook, we generally do not think about software—we think of "friends."[14] But the people who run Facebook are thinking about software twenty-four/seven. Consider this: Facebook owns 146 patents and patent applications, with some of the patents invented in-house, but most acquired through purchases of other companies.[15] Most, if not all, of these

---

14. *See* United States v. Jeffries, No. 3:10-CR-100, 2010 WL 4923335, at *5 n.3 (E.D. Tenn. Oct. 22, 2010) (explaining the "friend" system in Facebook).

15. A search in the United States Patent & Trademark Office ("USPTO") patent database resulted in fourteen patents assigned from inventors to Facebook. *USPTO Patent Full-Text and Image Database*, U.S. PATENT & TRADEMARK OFFICE, http://patft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnetahtml%2FPTO%2Fsearch-adv.htm&r=0&p=1&f=S&l=50&Query=an%2F%28facebook%29%0D%0A%0D%0A&d=PTXT (last visited

patents are software patents.[16] Because Facebook is the most visited website on the Internet,[17] it must zealously police the Internet to ensure that its look and feel, features, and accompanying services are not being copied.[18] Software is the backbone for these jealously-guarded assets, and Facebook did not hesitate to bring a trade dress

---

Nov. 23, 2011). A search in the USPTO's Assignment Database showed that Facebook is the assignee of 132 patents and patent applications from various assignors. *Patent Assignee Summary*, U.S. PATENT & TRADEMARK OFFICE, http://assignments.uspto.gov/assignments/q?db=pat&qt=asne&reel=&frame=&pat=&pub=&asnr=&asnri=&asne=Facebook&asnei=&asns (last visited Nov. 23, 2011).

16. Dr. Roy Schestowitz, *More Software Patents Stockpiling at Facebook*, TECHRIGHTS (Oct. 7, 2010, 3:51 PM), http://techrights.org/2010/10/07/sw-monopolist-clues/ (blogging about Facebook's acquisition of software patents); Fred Wilson, *More Patent Nonsense*, AVC: MUSINGS OF A VC IN NYC (Feb. 26, 2010), http://www.avc.com/a_vc/2010/02/more-patent-nonsense.html (critiquing Facebook's software patent for news feeds). For an explanation of Facebook's news feed software patent, see Gene Quinn, *Facebook Gets US Patent on Social Network News Feeds*, IPWATCHDOG (Feb. 26, 2010, 2:59 PM), http://ipwatchdog.com/2010/02/26/facebook-social-network-news-feed-patent/id=9317/ ("The truth is patents like this, and many other software patents no doubt, give the anti-software patent crowd all the ammunition they need to say software patents and computer related innovations should not be patentable.").

17. Bianca Bosker, *Facebook More Popular Than Any Other Website—By a Lot: Nielsen*, HUFFINGTON POST (Sept. 12, 2011, 5:22 PM), http://www.huffingtonpost.com/2011/09/12/facebook-most-popular-website-nielsen_n_958254.html ("American Internet users now devote more time to Facebook than any other website, spending a total of 53.5 billion minutes a month on the world's largest social networking site."); *see* Facebook, Inc. v. MaxBounty, Inc., 274 F.R.D. 279, 280 (N.D. Cal. 2011) (stating that Facebook has 500 million active users).

18. *See* Facebook, Inc. v. StudiVZ Ltd., No. C 08-3468 JF (HRL), 2009 WL 1190802 (N.D. Cal. May 4, 2009). A summary of the claims is as follows:

> Plaintiff Facebook, Inc. ("Facebook") alleges that Defendants StudiVZ ("StudiVZ") and related corporate entities variously bearing the name Holtzbrink ("the Holtzbrink entities," collectively "Defendants") are responsible for the creation of an illegal "knock-off" version of Facebook's popular social utility website. StudiVZ launched its original website in 2005. The site, which bears a disputed degree of visual and functional similarity to Facebook, was oriented towards university students. As the site became increasingly popular in Germany, StudiVZ launched several other services targeting other age groups. Believing that the StudiVZ websites infringed its proprietary trade dress, Facebook sent a demand letter to StudiVZ in 2006.

Id. at *1.

infringement action against a German company for copying the look and feel of its website.[19]

Like Facebook, Groupon is experiencing an astronomical rise in popularity and valuation.[20] Groupon, which offers online coupons to groups, is currently valued at around $15 to $20 billion and is waiting for its initial public offering this year.[21] Groupon has a very simple business model[22] and, like Facebook, could not function without software.[23] It owns very few issued patents[24] and has a few

---

19. *Id*.

20. Rita McGrath, *The Problem With Groupon's Business Model*, HBR BLOG NETWORK (July 13, 2011, 12:05 PM), http://blogs.hbr.org/hbr/mcgrath/2011/07/the-problem-with-groupons-busi.html.

21. *Id.*

22. *Id.*; Panos Mourdoukoutas, *Is Groupon's Business Model Sustainable?*, FORBES (Oct. 22, 2011, 2:04 PM), http://www.forbes.com/sites/panosmourdoukoutas/2011/10/22/is-groupons-business-model-sustainable/ ("While Groupon's model is simple, it isn't sustainable, for two reasons. First, as has been the case with other web-based companies like Netflix . . . and Open Table . . . , Groupon is selling other companies' products that have the upper hand in any deal negotiations. Second, they have plenty of competition from direct offerings from companies and from other web-based companies with a broad user base like Google . . . , Amazon.com . . . , Yahoo . . . , Expedia . . . , Priceline.com . . . and Travelzoo . . . ."); *see also* Merri A. Baldwin et al., *Should I Friend the Judge and Other Current Ethical Challenges for Lawyers*, *in* 2 TWELFTH ANNUAL INSTITUTE ON PRIVACY AND DATA SECURITY LAW 21, 36 (2011) ("Groupon is a social media site that offers users discounts on goods and services offered by advertisers. The advertiser pays Groupon a percentage of the fee earned by the advertiser from Groupon users who obtain the discounts.").

23. Sarah Lacy, *NetSuite Is Pushing into Enterprise Software, Lands Groupon as Marquee Customer*, TECHCRUNCH (May 10, 2011), http://techcrunch.com/2011/05/10/netsuite-is-pushing-into-enterprise-software-lands-groupon-as-marquee-customer/ (reporting that Groupon's fast growth depends on software).

24. A search conducted in the USPTO's patent database for patents assigned from Groupon's employee inventors to Groupon showed no patent. A search conducted in the USPTO's Assignment Database showed that Groupon received thirty-two patents and patent applications from various assignors. *Patent Assignee Summary*, UNITED STATES PATENT & TRADEMARK OFFICE, http://assignments.uspto.gov/assignments/q?db=pat&asne=GROUPON&page=1 (last visited Nov. 23, 2011). Groupon has been in a patent litigation against MobGob LLC over the 6,269,343 patent issued in 2001. *See* Victoria Slind-Flor, *Groupon, Yale, Ray Charles: Intellectual Property*, BLOOMBERG BUSINESSWEEK (Nov. 22, 2010, 11:07 AM), http://www.businessweek.com/news/2010-11-22/groupon-yale-ray-charles-intellectual-property.html.

pending patent applications.[25] Its patents are software business methods issued during the Internet bubble eruption.[26] Essentially, Groupon uses software to build and maintain its business model.

The cloud, or cloud computing, is another new technology,[27] in which companies manage their clients' web hosting, email, blogs, and applications in a digital network using multiple servers.[28] In essence, cloud computing is all about software.[29] Much like Facebook depends on software

---

25. A search conducted on July 18, 2011 in the United States Patent and Trademark Office's website for patents and pending applications in the name of Groupon as the Assignee from employee inventors revealed that there are no patents issued in Groupon's name. Presently, Groupon owns patents through acquisition only; it owns no self-developed patents. *See supra* note 24.

26. *See* Victoria Slind-Flor, *Groupon, IPCom, Bayer, ICANN: Intellectual Property*, BLOOMBERG (Dec. 3, 2010, 7:01 AM), http://www.bloomberg.com/news/2010-12-03/groupon-ipcom-nokia-bayer-icann-intellectual-property.html ("Groupon's only patents were bought from other companies, including one owned by MobShop, a site that closed in 2001.").

27. *See* Todd Machtmes, *Cloud Computing—A Lawyer's Primer*, *in* CLOUD COMPUTING 2011: CUT THROUGH THE FLUFF AND TACKLE THE CRITICAL STUFF 359, 375 (2011) ("The world is quite clearly moving toward a model of cloud computing for an increasing number and array of tasks that used to be performed by individuals and companies on systems owned and run by the user.").

28. *See* Peter Mell & Tim Grance, *The NIST Definition of Cloud Computing*, NAT'L INST. OF STANDARDS AND TECH. (Oct. 7, 2009), http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf. NIST defines cloud computing as:

> [A] model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimum management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.

Id.

29. *See* Baldwin et al., *supra* note 22, at 24. An example of a cloud-based service model is "Software as a Service," or SaaS, described as follows:

> The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible through a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. Examples

for its program's look, feel, and utility and Groupon depends on software for its business model, companies offering cloud computing are offering software as a service ("SaaS").[30]

II. SOFTWARE IN THE INTELLECTUAL PROPERTY PARADIGM

To protect their work, software developers have had to rely on pre-software law to classify software within intellectual property's established doctrinal framework.[31] Because the foundations for intellectual property law were established hundreds of years before the arrival of software, it has been a difficult task to fit software into a particular classification of intellectual property.[32] The process is further complicated by the fact that software can encompass some combination of the traits of copyrights, trade dress, patents, and trade secrets.[33]

---

include Google Gmail, Google Docs, Facebook, Microsoft Business Online Services, and Zoho.

Id.

30. *See id.*; *see also IT Management SaaS Service Agreement (US Version)*, DELL (July 8, 2010), http://www.dell.com/downloads/global/services/Dell_IT_Mgmt_SaaS_Agreement__US_%20_v1_3.pdf.

31. *See* JULIE E. COHEN ET AL., COPYRIGHT IN A GLOBAL INFORMATION ECONOMY 266 (2d ed. 2006) (detailing the scope of various intellectual property regimes extended to protect software, such as patent, copyright, and trade secret); *see also* Michael Meehan, *Increasing Certainty and Harnessing Private Information in the U.S. Patent System: A Proposal for Reform*, 2010 STAN. TECH. L. REV. 1, 5 (noting computer and software companies rely on the patent system for their software inventions).

32. *See* Gregory J. Maier, *Software Protection—Integrating Patent, Copyright and Trade Secret Law*, 69 J. PAT. & TRADEMARK OFF. SOC'Y 151, 151 (1987) ("It is the hybrid nature of software that causes its failure to fit neatly into any one existing category of intellectual property, resulting in seemingly endless confusion as to how it may best be protected."); Andrew Nieh, Note, *Software Wars: The Patent Menace*, 55 N.Y.L. SCH. L. REV. 295, 297-98 (2010) (observing the uncertainty of extending legal protection for software under the current intellectual property law regime).

33. *See* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2343-47 (1994) ("No one knows just where the boundary between these domains [copyright and patent] does or should lie.").

A. *Software as Copyrights*

In 1974, two years before the major revision of the copyright statute, Congress appointed the National Commission on New Technological Uses of Copyrighted Works ("CONTU") to tackle the issue of whether software is entitled to copyright protection.[34] Much of the discussion involved the distinction between the two forms software can take: source code and object code.[35]

---

34. Congress dictated the following mandate to CONTU:

> (b) The purpose of the Commission is to study and compile data on:
>
> (1) the reproduction and use of copyrighted works of authorship—
>
> (A) in conjunction with automatic systems capable of storing, processing, retrieving, and transferring information . . . .
>
> (c) The Commission shall make recommendations as to such changes in copyright law or procedures that may be necessary to assure for such purposes access to copyrighted works, and to provide recognition of the rights of copyright owners.

Act of Dec. 31, 1974, Pub. L. No. 93-573, § 201(b)-(c), 88 Stat. 1873, 1873-74.

35. *See* 17 U.S.C. § 101 (2006) ("A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."). The process of designing a computer software program using the source code and object code is described in *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992):

> In writing these directions, the programmer works from the general to the specific.
>
> The first step in this procedure is to identify a program's ultimate function or purpose. . . . Once this goal has been achieved, a programmer breaks down or "decomposes" the program's ultimate function into "simpler constituent problems or 'subtasks,'" which are also known as subroutines or modules. . . .
>
> . . . [A] programmer will then arrange the subroutines or modules into what are known as organizational or flow charts. . . .
>
> . . . .
>
> Once each necessary module has been identified, designed, and its relationship to the other modules has been laid out conceptually, the resulting program structure must be embodied in a written language that the computer can read. This process is called "coding," and requires two steps. First, the programmer must transpose the program's structural blue-print into a source code. This step has been described as "comparable to the novelist fleshing out the broad outline of his plot by crafting from words and sentences the paragraphs that convey the ideas." The source code may be written in any one of several computer languages, such as COBAL, FORTRAN, BASIC, EDL, etc.,

Source code is the direct product of computer programmers.[36] Using various computer languages, programmers write source code in a text editor, and the resultant file is readable to anyone familiar with the particular computer language in which the source code is written.[37] Object code is derived from source code, by way of a program called a compiler, which generates the object code from the source code.[38] Object code is a sequence of instructions in binary numbers—0s and 1s—that humans cannot comprehend, but which the computer processor can process.[39] Software is made up of this object code.[40]

In 1978, CONTU issued its Final Report, recommending that software be copyrightable in both object and source code forms.[41] In 1980, Congress amended the copyright

---

> depending upon the type of computer for which the program is intended. Once the source code has been completed, the second step is to translate or "compile" it into object code. Object code is the binary language comprised of zeros and ones through which the computer directly receives its instructions.
>
> After the coding is finished, the programmer will run the program on the computer in order to find and correct any logical and syntactical errors. This is known as "debugging" and, once done, the program is complete.

Id. at 697-98 (citations omitted) (internal quotation marks omitted).

36. *See Altai*, 982 F.2d at 698 (describing the software engineering process).

37. *Id.*

38. *Id.*

39. *Id.*

40. *See* Stephen Gold, *Overview of Open Source Software Issues*, *in* UNDERSTANDING THE INTELLECTUAL PROPERTY LICENSE 2010, at 385, 388 (2010) (explaining the industry practice that software companies distribute their software in object code form while keeping their source code confidential).

41. *See* Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 50 (D. Mass. 1990). The district court stated that "CONTU observed a need for copyright protection of creative expression embodied in computer programs" for the following reason:

> The cost of developing computer programs is far greater than the cost of their duplication. Consequently, computer programs . . . are likely to be disseminated only if . . . [t]he creator can spread its costs over multiple copies of the work with some form of protection against unauthorized duplication of the work. . . . The Commission is, therefore satisfied that some form of protection is necessary to encourage the creation and broad distribution of computer programs in a competitive market, . . .

statute to extend protection for software as literary work.[42] Subsequently, the Third Circuit has held that both source and object code could be copyrighted as literary work under the amended copyright provisions.[43]

As courts have acquired a better understanding of software, the law has evolved as to which aspects of software are entitled to protection under copyright law. Software has both literal and non-literal components.[44] The literal component consists of source code, object code and the visual displays; the non-literal component comprises "the structure, sequence, and organization of the program's code and visual displays."[45] Courts have narrowed the scope of the copyright protection for the non-literal components by applying the abstract-filtration-comparison test.[46] For

---

> [and] that the continued availability of copyright protection for computer programs is desirable.

Id. (internal quotation omitted).

42. *Id.* (noting that Congress amended section 101 to include the definition for "computer program" by "tracking verbatim CONTU's recommendation"); *see also* 17 U.S.C. § 101 (2006) ("A 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."). This definition qualified computer programs as literary works: "'Literary works' are works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied." *Id.*

43. Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983) ("[A] computer program, whether in object code or source code, is a 'literary work' and is protected from unauthorized copying, whether from its object or source code version."). Many commentators, however, are opposed to copyright protection for software. *See generally* Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 DUKE L.J. 663; Pamela Samuelson, *Creating a New Kind of Intellectual Property: Applying the Lessons of the Chip Law to Computer Programs*, 70 MINN. L. REV. 471 (1985).

44. *See, e.g.*, MDY Indus., LLC v. Blizzard Entm't, Inc., 629 F.3d 928, 952-54 (9th Cir. 2010) (dividing the World of Warcraft computer game software into literal and non-literal elements).

45. *See* XUAN-THAO NGUYEN ET AL., INTELLECTUAL PROPERTY, SOFTWARE AND INFORMATION LICENSING: LAW AND PRACTICE 505 (2006).

46. *See* William B. Bunker, *Method Approach: For Comprehensive Software Protection, Patent Is Preferable to Copyright*, KNOBBE MARTENS, http://www.kmob.com/art2.htm (last visited Nov. 23, 2011) ("More recently, however, courts have been narrowing copyright protection for computer software

example, in *Lotus Development Corp. v. Borland International Inc.*, the court denied copyright protection to most of the non-literal elements of Lotus's 1-2-3 spreadsheet.[47]

In the aftermath of this and other caselaw narrowing copyright protection for software, programmers have developed technological devices to prevent unauthorized access or use of their software.[48] Congress has taken steps to bolster the steps taken by developers; for example, Congress passed the Digital Millennium Copyright Act ("DMCA"), which prohibits users from circumventing the technological measures employed by copyright holders to control unauthorized access to copyrighted works.[49] The DMCA also prohibits users from manufacturing, importing, trafficking in, and marketing devices for circumvention purposes.[50]

The duration of copyright protection for software is very long.[51] The duration of a copyright for software legally owned by the individual developer who created it is the life of the author plus seventy years; for software developed as a work made for hire, the duration is either 120 years from creation or 95 years from publication.[52] Because the reality is that most software has a much shorter economic life,

---

by closely examining the program's individual elements rather than the overall characteristics or look and feel.").

47. 49 F.3d 807, 815 (1st Cir. 1995), *aff'd by an equally divided court,* 516 U.S. 233 (1996).

48. *See* NGUYEN ET AL., *supra* note 45, at 506 ("[S]ome software developers are deploying technological devices to prevent unauthorized use[, such as] devices that meter how many simultaneous copies are in use, digital rights management code, digital watermarks, product activation code, and copy protection."). Game developers, for example, rely on copyright law to prevent unauthorized access to the game's "dynamic non-literal elements" which are entitled to copyright protection "independently from the software program code, even though [these elements are] partially dependent on user input." *MDY*, 629 F.3d at 953-54 (citing Atari Games Corp. v. Oman, 888 F.2d 878, 884-85 (D.C. Cir. 1989)).

49. Digital Millennium Copyright Act, Pub. L. No. 105-304, 112 Stat. 2860 (1998) (codified as amended at 17 U.S.C. § 1201 (2006)).

50. 17 U.S.C. § 1201(b).

51. Duncan M. Davidson, *Common Law, Uncommon Software*, 47 U. PITT. L. REV. 1037, 1079 (1986) ("The long term of copyright is largely irrelevant to software, rather than an impediment to competition.").

52. *See* 17 U.S.C. § 302.

many commentators have advocated for a shorter duration of copyright protection for software.[53]

Contrary to popular understanding, the underlying purpose behind all types of copyright protection is not to reward developers or authors, but instead to advance the public welfare by encouraging new, creative, tangible ideas, along with their disclosure and dissemination to the public.[54] Although the creative labor expended by software developers does garner them a fair return through sales or licensure, the "ultimate aim" of copyright protection is still "to stimulate artistic creativity for the general public good."[55]

Nevertheless, copyright protection provides a powerful tool to the developers of proprietary software, who can use it to bring actions against others for whole-sale reproduction of the software, or for the distribution and sale of counterfeit copies.[56] Even in cases where developers encourage others to freely use or modify their source code—for example, in open source software—developers continue to rely on copyright law to dictate the terms of the source code's modification or use.[57]

---

53. *E.g.*, Leo J. Raskind, *The Uncertain Case for Special Legislation Protecting Computer Software*, 47 U. PITT. L. REV. 1131, 1153 (1986) ("Given the rapid rate of innovation in the production of software programs, a shorter term of protection seems warranted.").

54. *See* Mazer v. Stein, 347 U.S. 201, 219 (1954) ("[E]ncouragement of individual effort by personal gain is the best way to advance public welfare through talents of authors and inventors in 'Science and useful Arts.'").

55. Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417, 432 (1984).

56. *See* Microsoft Corp. v. Rechanik, 249 F. App'x 476 (7th Cir. 2007) (holding that distributor of counterfeit software violated copyright laws). In addition to civil action, software counterfeiters also face criminal proceedings. *See, e.g.*, United States v. Kononchuk, 485 F.3d 199 (3d Cir. 2007) (discussing software counterfeiter's criminal liability).

57. *See* Robert W. Gomulkiewicz, *How Copyleft Uses License Rights to Succeed in the Open Source Software Revolution and the Implications for Article 2B*, 36 HOUS. L. REV. 179, 180, 185-86 (1999). Gomulkiewicz observed that hackers or proponents of the Open Source movement used copyright law to license and control their software "because they want to control what is done with their code. Licensing allows hackers to perpetuate their particular software development and distribution model. Without licensing, the open source software development model would be nothing more than an honor system." *Id.* at 186; *see also* Robert A. Hillman & Maureen A. O'Rourke, *Rethinking Consideration in the Electronic Age*, 61 HASTINGS L.J. 311, 313-14 (2009) ("[T]he

However, some software developers find that copyright law poses as many problems as it solves. For example, software developers often create different versions of software,[58] and the process of obtaining copyright registration for each version can be burdensome and costly.[59] Furthermore, the Copyright Office requires submission of the software's source code for registration.[60] Although the majority of source code can be redacted for copyright registration purposes,[61] registration brings with it the risk of the source code's disclosure.[62] Because of these burdens and risks, many developers avoid copyright registration in favor of other types of intellectual property

---

General Public License (GPL) . . . authorizes copyholders to transfer, copy, or modify the software subject to a series of restrictions. The restrictions are designed to further an environment of openness by requiring copyholders to reveal the source code to transferees of any software products that are derived from the original source code (often referred to as the 'copyleft' provision) and to transfer such software under the same terms as the GPL ('same terms' provision), making the terms themselves 'viral' in nature.").

58. *In re* World Auxiliary Power Co., 303 F.3d 1120, 1131 (9th Cir. 2002). The Ninth Circuit recognized that most copyrightable works are not registered because a "copyright is created every time people set pen to paper, or fingers to keyboard, and affix their thoughts in a tangible medium, writers, artists, computer programmers, and web designers would have to have their hands tied down to keep them from creating unregistered copyrights all day every day." *Id.*

59. As developers create different versions of software, failure to submit the correct version of the source code for registration purposes may prevent the developers from claiming ownership of the source code for purposes of copyright infringement litigation. *See* Geoscan, Inc. v. Geotrace Techs., Inc., 226 F.3d 387, 393 (5th Cir. 2000) (noting that although the plaintiff submitted a deposit of the first ten lines and last ten lines of its software source code with the copyright registration application and fee, the plaintiff submitted the later versions of the source code, not the original version, and therefore failed to establish that it has ownership in its software for the purposes of a copyright infringement claim).

60. *See Copyright Registration for Computer Programs*, U.S. COPYRIGHT OFFICE (May 2011), http://www.copyright.gov/circs/circ61.pdf.

61. *See id.* (outlining the amount of software code that may be permissibly redacted); *see also* Fonar Corp. v. Domenick, 105 F.3d 99, 105 (2d Cir. 1997) (holding that the plaintiff need only submit the first and last twenty-five pages of source code to the Copyright Office to fulfill the registration requirement).

62. *See* Gen. Universal Sys., Inc. v. Lee, 379 F.3d 131, 138 n.4 (5th Cir. 2004) (noting that the Copyright Office misplaced the source code printouts after the software company submitted the source code for copyright registration).

law for protection of their interests (where such protection is not preempted).[63]

B. *Software as Trade Dress*

Facebook brought a trade dress infringement action against StudiVZ for copying the user interface or the overall appearance of its website.[64] Facebook is just one example of the many developers who have looked to trade dress law for legal protection after the courts rejected the concept of copyright protection for software's non-literal components. As another example, software owners and developers have brought trade dress infringement action against others for copying screen displays.[65]

As noted above, the non-literal components of software generally include the user interface—the overall structure and organization of a computer program, including its audiovisual displays, or screen look and feel.[66] The look and

---

63. For example, a trade secret claim is not preempted by the Copyright Act and therefore developers may bring both copyright infringement and misappropriation of trade secret claims. *See* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 717 (2d Cir. 1992); S.O.S., Inc. v. Payday, Inc., 886 F.2d 1081, 1090 n.13 (9th Cir. 1989) ("Since the California statute pleaded in this case does not involve a legal or equitable right equivalent to an exclusive right of a copyright owner under the Copyright Act, but only prohibits certain means of obtaining confidential information, its application here would not conflict with federal copyright law.").

64. Complaint, Facebook, Inc. v. StudiVZ Ltd., 2009 WL 1190802 (N.D. Cal. 2009) (No. 5:08-cv-03468), 2008 WL 2914576 (asserting various trade dress infringement claims); *see also* Kevin O'Brien, *Facebook and StudiVZ Battle over Germany*, N.Y. TIMES, (Aug. 7, 2008), http://www.nytimes.com/2008/08/07/technology/07iht-social.4.15091587.html (discussing Facebook's suit against the German site StudiVZ for mimicking its user interface).

65. *See* Complaint, Fidelity Info. Servs., Inc. v. Debtdomain GLMS Pte Ltd., 2011 WL 4526140 (S.D.N.Y. 2011) (No. 09 Civ. 7589), 2009 WL 2895016 (alleging that the defendant systematically copied Fidelity's SyndTrack Software and created defendant's Syndication Software to look like SyndTrack); *see also* Chad King, Note, *Abort, Retry, Fail: Protection for Software-Related Inventions in the Wake of* State Street Bank & Trust Co. v. Signature Financial Group, Inc., 85 CORNELL L. REV. 1118, 1161 (2000) ("By patenting the 'guts' of the program and obtaining trade dress protection for the look and feel of the interface, a software developer may be able to secure comprehensive protection for the entire application.").

66. *See* Lisa M. Byerly, Comment, *Look and Feel Protection of Web Site User Interfaces: Copyright or Trade Dress?*, 14 SANTA CLARA COMPUTER & HIGH TECH.

feel of a product, or the total appearance or packaging of a product, is entitled to trade dress protection under section 43(a) of the Lanham Act if the trade dress is (1) inherently distinctive, and (2) non-functional.[67]

1. *Inherently Distinctive.* The "look" and "feel" of a product or service is inherently distinctive if it functions as a source identifier. A source identifier identifies and distinguishes a product from other products on the market. For example, the décor, style, menu, and ambience of a Mexican fast-food restaurant,[68] the shape of a Coke bottle,[69] and the gold-foil dressing of a Godiva chocolate[70] all qualify

---

L.J. 221, 222-23 (1998) (describing the look and feel of user interfaces). *See generally id.* (discussing how various intellectual property regimes might protect the look and feel of user interfaces); Lauren Fisher Kellner, Comment, *Trade Dress Protection for Computer User Interface "Look and Feel*,*"* 61 U. CHI. L. REV. 1011 (1994) (discussing the use of trade dress law to protect the look and feel of user interfaces).

67. Lanham Act, ch. 540, § 43(a), 60 Stat. 427, 444 (1946) (codified at 15 U.S.C. § 1125 (2006)); *see* Mattel Inc. v. Walking Mountain Prods., 353 F.3d 792, 808 n.13 (9th Cir. 2003) ("Trade dress involves 'the total image of a product and may include features such as size, shape, color or color combination, texture, graphics, or even particular sales technique.'" (quoting Two Pesos, Inc. v. Taco Cabana, Inc., 505 U.S. 763, 764 n.1 (1992))).

68. *See Two Pesos*, 505 U.S. at 764 & n.1, 776 (extending trademark protection to trade dress of a fast food restaurant and holding that inherently distinctive trade dress functions as a source identifier and thus secondary meaning is not required); *see also* Wal-Mart Stores, Inc. v. Samara Bros., 529 U.S. 205, 214-15 (2000) ("In [*Two Pesos*], we held that the trade dress of a chain of Mexican restaurants, which the plaintiff described as 'a festive eating atmosphere having interior dining and patio areas decorated with artifacts, bright colors, paintings and murals,' could be protected under § 43(a) without a showing of secondary meaning." (internal citation omitted)).

69. *Wal-Mart*, 529 U.S. at 215. The Court recognized that under trade dress classification a classic Coca-Cola bottle may be viewed as both trade dress packaging and trade dress design because:

> [A] classic glass Coca-Cola bottle, for instance, may constitute packaging for those consumers who drink the Coke and then discard the bottle, but may constitute the product itself for those consumers who are bottle collectors, or part of the product itself for those consumers who buy Coke in the classic glass bottle, rather than a can, because they think it more stylish to drink from the former.

Id.

70. Grey v. Campbell Soup Co., 650 F. Supp. 1166, 1173-74 (C.D. Cal. 1986) (finding the defendants' use of the silver ballotin for Dogiva and Cativa biscuits infringed on Godiva's trade dress).

as source identifiers because each product identifies to potential consumers exactly whose product it is. Likewise, for a software user-interface design to qualify as a source identifier, it must identify and distinguish the software from the software developed, distributed, or sold by others.[71] Unless the user-interface design or audiovisual display of the software is unique and the developer has actually used the display to advertise and sell the software, it is very difficult to assert that the total appearance of the software is inherently distinctive and functions as a source identifier.[72]

If a trade dress is not inherently distinctive, the law requires the trade dress to acquire distinctiveness or secondary meaning.[73] For the software developer whose user interface is not inherently distinctive, this means proving that through years of use and advertisement, the software user interface has *become* the source of the product in the mind of the consumer.[74] This is not an easy task because such proof is factually intensive, and it is costly to obtain the evidence.[75] Courts generally require consumer survey evidence as part of the proof that the trade dress has acquired distinctiveness or secondary meaning.[76]

---

71. *See* Computer Access Tech. Corp. v. Catalyst Enters., Inc., No. C-00-4852, 2001 WL 34118030, at *11-12 (N.D. Cal. June 13, 2001).

72. *See generally id*. (finding that the user interface failed to function as a source identifier, despite the plaintiff having spent $4.5 million on advertising).

73. *See Two Pesos*, 505 U.S. at 776.

74. *See, e.g.*, SG Servs., Inc. v. God's Girls Inc., No. CV 06-989, 2007 WL 2315437, at *8-10 (C.D. Cal. May 9, 2007) (finding no secondary meaning had been acquired for the look and feel of the website or user interface). *See generally* Xuan-Thao N. Nguyen, *Should It Be a Free for All?: The Challenge of Extending Trade Dress Protection to the Look and Feel of Web Sites in the Evolving Internet*, 49 AM. U. L. REV. 1233, 1237-38 (2000) (advocating that the trade dress for a website that acquires secondary meaning receive protection under the Lanham Act).

75. Kevin D. Hughes & David E. Rosen, *Screen Grabbing*, L.A. LAW., June 2010, at 42, 44 (2010) (noting that a court found that a website user interface had not acquired secondary meaning, despite the website being immensely popular).

76. *See, e.g.*, *SG Servs.*, 2007 WL 2315437, at *9 (finding that the evidence submitted by the plaintiff failed to establish that its website has acquired secondary meaning and citing to a case where the plaintiffs provided consumer

2. *Non-Functional.* To qualify for trade dress protection, not only must the elements of a software interface be distinctive, they must also be non-functional.[77] This requirement has a utilitarian angle: the elements cannot be essential to the manufacture, development, usage, or purposes of the products, because giving one party exclusive rights in those elements would put competitors at a significant disadvantage unrelated to reputation.[78]

Elements affecting the cost or quality of the products are also considered functional because they, too, disadvantage competitors in a way unrelated to reputation.[79]

The non-functionality requirement also includes the "aesthetic functionality doctrine,"[80] which denies trade dress protection to products possessing competitive advantages relating to some aesthetic value unrelated to reputation.[81]

---

survey to support that the trade dress has become a source identifier in the mind of the consuming public).

77. *See* Rachel v. Banana Republic, Inc., 831 F.2d 1503, 1506 (9th Cir. 1987). The Ninth Circuit provided that:

> A product feature is functional if it is essential to the product's use or if it affects the cost and quality of the product. In determining functionality, a product's trade dress must be analyzed as a whole. The issue of functionality has been consistently treated as a question of fact. . . . [H]owever, we have placed the burden of proof on the plaintiff. . . . Functional features of a product are features which constitute the actual benefit that the consumer wishes to purchase, as distinguished from an assurance that a particular entity made, sponsored, or endorsed a product.

Id. (citations omitted) (internal quotation marks omitted).

78. *See* Peter Lee, *The Evolution of Intellectual Infrastructure*, 83 WASH. L. REV. 39, 85 (2008) (noting that widespread adoption of certain user interface features discourages extending protection to the features for fear of hindering competition).

79. Inwood Labs., Inc. v. Ives Labs., Inc., 456 U.S. 844, 850 n.10 (1982).

80. Qualitex Co. v. Jacobson Prods. Co., 514 U.S. 159, 168-70 (1995) (discussing aesthetic functionality and color trademarks).

81. Jay Franco & Sons, Inc. v. Franek, 615 F.3d 855, 860-61 (7th Cir. 2010). The Seventh Circuit applied the aesthetic functionality doctrine and rejected the trade dress of circle-shaped beach towels, explaining that "[f]ashion is a form of function. A design's aesthetic appeal can be as functional as its tangible characteristics. And many cases say that fashionable designs can be freely copied unless protected by patent law." *Id.* at 860 (citations omitted). The court continued: "A circle is the kind of basic design that a producer like Jay Franco

For example, under the aesthetic functionality doctrine, it would be an impermissible monopoly to provide trade dress protection for heart-shaped chocolate because the relationship between love and the giving of chocolates is unrelated to the reputation of any one company or product.[82]

Accordingly, the non-functionality requirement will also limit the elements of a software user interface eligible for trade dress protection[83] where there is an overlap between what is "functional" and what is "non-functional."[84]

C. *Software as Trade Secrets*

In some cases, even where software is patentable, the developers will instead rely on trade secret law for protection, to avoid having to disclose their invention as part of the patent application.[85] Software developers rely on trade secret law for the protection of source code,[86] as well as other aspects of software, including "feature lists, bug databases, specifications, flow charts, and protocol

---

adopts because alternatives are scarce and some consumers want the shape regardless of who manufactures it." *Id.* at 861.

82. *Cf. Qualitex*, 514 U.S. at 169-70 ("[W]e note that lower courts have permitted competitors to copy the green color of farm machinery (because customers wanted their farm equipment to match) and have barred the use of black as a trademark on outboard boat motors (because black has the special functional attributes of decreasing the apparent size of the motor and ensuring compatibility with many different boat colors)." (citations omitted)).

83. SG Servs., Inc. v. God's Girls Inc., No. CV 06-989, 2007 WL 2315437, at *9 (C.D. Cal. May 9, 2007) (accepting the plaintiff's assertion that its website colors and phraseology features were not functional under the Ninth Circuit test in *Rachel v. Banana Republic*).

84. *See* Nguyen, *supra* note 74, at 1265-69.

85. Lowndes Prods., Inc. v. Brower, 191 S.E.2d 761, 766 (S.C. 1972) ("A trade secret need not be a patentable invention. The fact that an invention is patentable does not compel the taking out of a patent, nor prevent the person entitled to it from keeping it secret." (citations omitted) (internal quotation marks omitted)).

86. *See* ClearOne Commc'ns, Inc. v. Bowers, 651 F.3d 1200, 1203, 1209-11 (10th Cir. 2011) (affirming the jury verdict against the corporation defendant for misappropriating the plaintiff's computer codes); Mid-Mich. Computer Sys., Inc. v. Marc Glassman, Inc., 416 F.3d 505, 506 (6th Cir. 2005) (affirming the damages award of $2 million against the defendant for unauthorized access and use of the plaintiff's source code).

information . . . ."[87] Because the value of a trade secret diminishes when members of the public learn the secret, the owner of a trade secret often expends considerable effort to safeguard the trade secret.[88]

Historically,[89] trade secrets have received legal protection as a form of property.[90] The development of

---

87. NGUYEN ET AL., *supra* note 45, at 506.

88. R.C. Olmstead, Inc., v. CU Interface, LLC, 606 F.3d 262, 276-77 (6th Cir. 2010) (affirming the district court's finding that the plaintiff did not take "affirmative steps . . . to protect its alleged trade secret—its software interface"). In *Minnesota Mining & Manufacturing Co. v. Kirkevold*, 87 F.R.D. 324 (D. Minn. 1980), the district court found that the acquisition of 3M trade secrets by the defendant company's employment of a former 3M employee posed a threat to the disclosure of the trade secrets because "the acquisition of such knowledge by Verbatim can only tend to diminish the competitive advantage which the confidential character of the information provides to 3M, as the secrecy of 3M's property would be unveiled." *Id.* at 326-27, 338. Companies employ various safeguards to maintain secrecy of their trade secrets and seek judicial intervention to prevent disclosure of trade secrets. *See* Consumer Prod. Safety Comm'n v. GTE Sylvania, Inc., 447 U.S. 102, 107, 112, 124 (1980) (affirming the injunction against the Consumer Product Safety Commission from disclosing accident reports that contain trade secret information to the public).

89. On the other side of the Atlantic, the first reported English cases on misappropriation of trade secrets appeared in 1817. *See* De Lage Landen Operational Servs., LLC v. Third Pillar Sys., LLC, No. 09-2439, 2011 WL 1627899, at *3 (E.D. Pa. Apr. 28, 2011) (stating that the first case regarding trade secret protection was *Newberry v. James*, (1817) 35 Eng. Rep. 1011 (Ch.)); Highland Tank & Mfg. Co. v. PS Int'l, Inc., 393 F. Supp. 2d 348, 355 (W.D. Pa. 2005) (citing *Newberry* as the earliest reported trade secret case); *see also* Miguel Deutch, *The Property Concept of Trade Secrets in Anglo-American Law: An Ongoing Debate*, 31 U. RICH. L. REV. 313, 313 n.3 (1997) (discussing the two earliest English cases on trade secret protection). In the United States, early trade secret misappropriation cases surfaced in 1837. *See* Vickery v. Welch, 36 Mass. 523 (1837); *see also* Sharon K. Sandeen, *The Evolution of Trade Secret Law and Why Courts Commit Error When They Do Not Follow the Uniform Trade Secrets Act*, 33 HAMLINE L. REV. 493, 498 (2010) (discussing the trade secret claims in *Vickery*). The concept of trade secret as property was first recognized by the Massachusetts Supreme Judicial Court in 1868. *See* William B. Barton, *A Study in the Law of Trade Secrets*, 13 U. CIN. L. REV. 507, 513-14 (1939) (citing Peabody v. Norfolk, 98 Mass. 452 (1868)).

90. Shyamkrishna Balganesh, *The Pragmatic Incrementalism of Common Law Intellectual Property*, 63 VAND. L. REV. 1543, 1555 (2010) (stating that courts originally viewed trade secrets as property). The difference between a trade secret and ordinary property is that a holder of a trade secret lacks the right to exclude others from using the trade secret; therefore, trade secrets are thought of as weak property. *See* Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470,

decisional law for the protection of trade secrets serves two important policy goals: maintaining standards of commercial ethics and encouraging invention.[91] To better serve these two important policies, the National Conference of Commissioners on Uniform State Laws drafted the model Uniform Trade Secrets Act ("UTSA") to resolve conflicting state laws on trade secrets, and most states have adopted the UTSA, either in whole or in part.[92] The UTSA provides a uniform definition for trade secret, explains how a trade secret is misappropriated, lists proper means of obtaining trade secrets, and affords remedies for violation of another's trade secret.[93]

Under trade secret law, both source code and object code may qualify for protection if the software can derive independent economic value and is not generally known or readily ascertainable by the public.[94] This means that a software developer can bring a trade secret misappropriation claim against others for using improper means to gain access to the code.[95] While dependent on the jurisdiction, "improper means" generally include "theft, bribery, misrepresentation, breach or inducement of a

---

489-90 (1974); Robert G. Bone, *A New Look at Trade Secret Law: Doctrine in Search of Justification*, 86 CALIF. L. REV. 241, 254 (1998).

91. *Kewanee Oil*, 416 U.S. at 481.

92. *See* John T. Cross, *UTSA Displacement of Other State Law Claims*, 33 HAMLINE L. REV. 445, 446 (2010) (observing that forty-six states have adopted the UTSA).

93. UNIF. TRADE SECRETS ACT § 1 & cmts. 2-3 (amended 1985); *see* Elizabeth A. Rowe, *Striking a Balance: When Should Trade-Secret Law Shield Disclosures to the Government?*, 96 IOWA L. REV. 791, 799-800 (2010) (explaining trade secret law and protection under UTSA).

94. Trandes Corp. v. Guy F. Atkinson Co., 996 F.2d 655, 663 (4th Cir. 1993). The Fourth Circuit held that the source code of "Tunnel System" "(1) is not generally known, (2) is not readily ascertainable by proper means, and (3) if acquired by competitors would improve their ability to compete with" the plaintiff; therefore, the source code was protected under trade secret law. *Id.* The court in *Trandes* also held that the object code of the "Tunnel System" qualified for trade secret protection. *Id*. at 664.

95. *Id*. at 667 (affirming the jury verdict on trade secret misappropriation of the software code); *see also* Mid-Mich. Computer Sys., Inc. v. Marc Glassman, Inc., 416 F.3d 505, 512 (6th Cir. 2005) (finding that the defendants misappropriated the plaintiff's source code "in its entirety" with "line-by-line code copying").

breach of a duty to maintain secrecy, or espionage through electronic or other means."[96]

Trade secret protection for software ceases when the source code of the software is no longer a secret.[97] As such, the legal life or protection for a trade secret is indeterminable, as it can last for days, months, or years, depending on whether it remains a secret. Sometimes, the software developer will disclose the source code, whereas other times, it will become readily ascertainable by others utilizing proper means.[98] "Proper means" generally includes discovering the trade secret "by independent invention," observing the trade secret "in public use or on public display," "obtaining the trade secret from published literature," or discovering the trade secrets by "reverse engineering."[99]

In reverse engineering, a customer purchases a copy of a software, then decompiles and disassembles the object code, working backwards to obtain the underlying source code.[100] Courts have routinely held that reverse engineering of software code is "fair use" under copyright law.[101] Upon a

---

96. *E.g.*, IDAHO CODE ANN. § 48-801(1) (West 2003); *see also* Vt. Microsystems, Inc. v. Autodesk, Inc., 88 F.3d 142, 145, 148-49 (2d Cir. 1996) (finding defendants misappropriated the plaintiff's trade secret software code through improper means by former employee programmers who had access to plaintiff's code).

97. *Cf.* JustMed, Inc. v. Byce, 600 F.3d 1118, 1128-30 (9th Cir. 2010) (holding the disclosure by the former employee of the source code was not sufficient to result in loss of trade secret).

98. *See* R.C. Olmstead, Inc., v. CU Interface, LLC, 606 F.3d 262, 276 (6th Cir. 2010) (finding that third-party companies have unfettered access to the plaintiff's software interface and therefore there was no trade secret protection for the interface).

99. UNIF. TRADE SECRETS ACT § 1 cmt. (amended 1985) (internal quotation marks omitted).

100. *See* Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 599-601, 608 (9th Cir. 2000) (discussing process of reverse engineering and holding that reverse engineering of a video game software was fair use).

101. *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1518 (9th Cir. 1992) (holding that intermediate copying of a videogame console for purpose of reverse engineering to obtain the software code was fair use); Atari Games Corp. v. Nintendo of Am. Inc., 975 F.2d 832, 843 (Fed. Cir. 1992) ("[R]everse engineering object code to discern the unprotectable ideas in a computer program is a fair use.").

third party's reverse engineering of the software code, the owner of the trade secret will no longer have legal protection for the software because its code has become known to the public.[102] To prevent reverse engineering, the owner of a software trade secret can include a provision in the software license agreement prohibiting the reverse engineering of the software,[103] which is often done in mass-market license agreements to end-users.[104]

D. *Software as Patents*

The patentability of certain software innovations has had a tortuous history, as courts have wrestled with the question of whether software is patentable subject matter.[105] Patentable subject matter is one of the statutory requirements for patentability.[106] The other requirements dictate that an invention must be useful, novel, nonobvious,

---

102. Richard F. Dole, Jr., *The Uniform Trade Secrets Act—Trends and Prospects*, 33 HAMLINE L. REV. 409, 414 (2010) ("Acquiring knowledge of a trade secret by proper means, including independent discovery and reverse engineering of a publicly-available product, is not actionable under trade secret law.").

103. *See* Mid-Mich. Computer Sys., Inc. v. Marc Glassman, Inc., 416 F.3d 505, 507, 509 (6th Cir. 2005) (finding trade secret misappropriation in a case where the defendants signed the Source Code Agreement and engaged in unauthorized reverse engineering of the software).

104. *See, e.g.*, Davidson & Assocs. v. Jung, 422 F.3d 630, 634 & nn.4-5 (8th Cir. 2005).

105. For a discussion on the landscape of judicial decisions on software patents, see generally Susan J. Marsnik & Robert E. Thomas, *Drawing a Line in the Patent Subject-Matter Sands: Does Europe Provide a Solution to the Software and Business Method Patent Problem?*, 34 B.C. INT'L & COMP. L. REV. 227 (2011).

106. Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 483 (1974) ("[N]o patent is available for a discovery, however useful, novel, and nonobvious, unless it falls within one of the express categories of patentable subject matter."); *see also* 35 U.S.C. §101 (2006) ("Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.").

and enabling in order to receive patent protection under federal law.[107]

With respect to the patentable subject matter requirement, the four categories of invention are process, machine, manufacture, and composition of matter.[108] Natural and physical phenomena and abstract ideas are not patentable subject matter.[109] Mathematical algorithms frequently found in computer software programs are abstract ideas and excluded from patentable subject matter, as seen in a trilogy of Supreme Court cases.

In *Gottschalk v. Benson*, the invention at issue was a "method for converting binary-coded decimal (BCD) numerals into pure binary numerals."[110] The Supreme Court concluded that the invention was not patentable subject matter because the software algorithm was an abstract idea.[111]

Then, in *Parker v. Flook*, the Court rejected the patentability of another invention, again based on the rationale that a mathematical algorithm or formula is an abstract idea.[112] The Court further reasoned that the invention at issue was not recognizable as a "process" because the technological environment (the involvement of the computer) was insignificant post-solution activity.[113]

---

107. 35 U.S.C. § 101 (2006) (usefulness requirement); § 102 (novelty requirement); § 103 (nonobvious requirement); § 112 (enablement requirement).

108. Bilski v. Kappos, 130 S. Ct. 3218, 3225 (2010) (identifying the four categories of invention).

109. Diamond v. Chakrabarty, 447 U.S. 303, 309 (1980).

110. 409 U.S. 63, 64 (1972).

111. *Id*. at 71. In *Benson*, the Court rejected the software patent application a "process" under section 101 of the Patent Act because the application was an unpatentable abstract idea that merely "convert[ed] BCD numerals to pure binary numerals." *Id.* at 71. The court stated that a contrary holding "would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself." *Id*. at 72.

112. 437 U.S. 584, 585-86 (1978). The invention was a method for continually monitoring and adjusting alarm rates with the use of an algorithm. *Id.*

113. *Id.* at 590. In *Flook*, the invention at issue was an algorithm that monitored the conditions during the catalytic conversion process in the petrochemical and oil-refining industries. *Id*. at 585-86. The Court explained that the invention was "unpatentable under § 101, not because it contains a

Essentially, this means that one cannot convert an abstract idea to a patentable invention by merely placing it into one of the categories of patentable subject matter.[114]

Finally, in *Diamond v. Diehr*, the Court held a procedure for "molding raw, uncured synthetic rubber into cured precision products"[115] *was* patentable subject matter because it was not "an attempt to patent a mathematical formula, but rather [was] . . . an industrial process for the molding of rubber products."[116] The Court concluded that "an *application* of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection."[117] *Diehr* is an important case because the Court accepted that the algorithm or software can be part of a process invention and did not reject the process invention simply because it contains an algorithm.

These cases reject, per se, the patentability of software algorithms, computer programs, and software. However, the progression of this trilogy of cases left the door open for *Bilski v. Kappos*, in which the Federal Circuit recognized patentable subject matter for a process or method[118] of doing business that was "tied to a machine or apparatus" or "transform[ed] a particular article into a different state."[119] However, the Supreme Court rejected this "machine or transformation" test as the sole test to determine patentable subject matter for business method inventions, although it

---

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." *Id.* at 594.

114. *Id.* at 590 ("The notion that post-solution activity, no matter how conventional or obvious in itself, can transform an unpatentable principle into a patentable process exalts form over substance.").

115. 450 U.S. 175, 177 (1981).

116. *Id.* at 192-93.

117. *Id.* at 187.

118. 130 S. Ct. 3218, 3226 (2010) ("Section 100(b) [of the patent statute] provides that '[t]he term "process" means process, art or method, and includes a new use of a known process, machine, manufacture, composition of matter, or material.'" (quoting 35 U.S.C. § 100(b) (2006)).

119. *Id.* at 3225-27 (internal quotation marks omitted) (recounting what the Federal Circuit had developed as its own test for patentable subject matter for software or business method patents).

did recognize the test as useful.[120] The Court then referred the lower courts back to the "guideposts" of the software trilogy of *Benson*, *Flook* and *Diehr*.[121]

Post *Bilski*, if a business method claim added a computer, perhaps to perform faster calculations or to obtain a solution more efficiently, the business method claim is deemed to contain abstract ideas, thereby rendering it unpatentable.[122] To qualify for patent protection, a computer that has been added to a business method must "play a significant part" in allowing the claimed method to be performed.[123]

---

120. *Id*. at 3226-27 ("The machine-or-transformation test is not the sole test for deciding whether an invention is a patent-eligible 'process.'"); *see also* Kevin Emerson Collins, Bilski *and the Ambiguity of "An Unpatentable Abstract Idea,"* 15 LEWIS & CLARK L. REV. 37, 37 (2011) (criticizing the Supreme Court for offering nothing new except conclusory reasoning for the abstract-ideas test).

121. *Bilski*, 130 S. Ct. at 3230-31. The Court discussed the trilogy of cases and concluded that the application at issue is not a patentable "process" because "the basic concept of hedging . . . is a fundamental economic practice long prevalent in our system of commerce and taught in any introductory finance class. . . . Allowing 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." *Id.* at 3231 (citation omitted) (internal quotation marks omitted).

122. Marsnik & Thomas, *supra* note 105, at 265 ("Following *Bilski v. Kappos*, U.S. patent process subject-matter law has thus returned to the 1980s, perhaps with a revitalized exclusion for abstract ideas. Any mixed process claim that does not solely claim a law of nature, physical transformation, or abstract idea meets the initial patent subject-matter threshold.").

123. SiRF Tech., Inc. v. Int'l Trade Comm'n, 601 F.3d 1319, 1333 (Fed. Cir. 2010) ("In order for the addition of a machine to impose a meaningful limit on the scope of a claim, it must play a significant part in permitting the claimed method to be performed, rather than function solely as an obvious mechanism for permitting a solution to be achieved more quickly, i.e., through the utilization of a computer for performing calculations."); *see* Blake Reese, *Judicially Re(de)fining Software Patent Eligibility II: A Survey of Post*-Bilski *Jurisprudence*, 27 SANTA CLARA COMPUTER & HIGH TECH. L.J. 673, 682 (2011) (discussing recent Federal Circuit cases on software business method patents); Marsnik & Thomas, *supra* note 105, at 265. Post *Bilski*, "USPTO must apply the amorphous standards of *Benson*, *Flook*, and *Diehr*. If the claim provides a physical transformation of matter, then the decisions agree that the claim is patentable." *Id.* The U.S. Patent Office sent a message to the examining corps with the following guidance:

> Examiners should continue to examine patent applications for compliance with section 101 using the existing guidance concerning the machine-or-transformation test as a tool . . . . If a claimed method

III. CURRENT TAX APPROACHES TO SOFTWARE

Both the federal government and state governments have struggled with the taxation of computer software. Fitting computer software within current tax regimes has presented unique challenges because it contains both tangible and intangible elements, is subject to varying intellectual property protections, and is rapidly emerging into different products due to technological innovations. In determining taxability, governments have had to answer tough questions, such as: Should "canned" (i.e., prepackaged or off-the-shelf) software be treated the same for tax purposes as "custom" software?[124] Should varying intangible intellectual property rights covering software (copyright, trade dress, trade secret, or patent) impact tax treatments?[125] Should the method by which software is delivered (e.g., purchased at a store or received electronically) make a tax difference?[126] Government approaches to these questions have not always been

---

> meets the machine-or-transformation test, the method is likely patent-eligible under section 101 unless there is a clear indication that the method is directed to an abstract idea. If a claimed method does not meet the machine-or-transformation test, the examiner should reject the claim under section 101 unless there is a clear indication that the method is not directed to an abstract idea. If a claim is rejected under section 101 on the basis that it is drawn to an abstract idea, the applicant then has the opportunity to explain why the claimed method is not drawn to an abstract idea.

Memorandum from Robert W. Bahr, Acting Associate Commissioner for Patent Examination Policy, U.S. Patent and Trademark Office, to the Patent Examining Corps (June 28, 2010), *available at* http://www.uspto.gov/patents/law/exam/bilski_guidance_28jun2010.pdf. The "existing guidance" the statement references is the agency's. *See* Memorandum from Andrew H. Hirshfeld, Acting Deputy Commissioner for Patent Examination Policy, U.S. Patent and Trademark Office, to TC Directors (Aug. 24, 2009), *available at* http://www.uspto.gov/patents/law/comments/2009-08-25_interim_101_instructions.pdf (containing "New Interim Patent Subject Matter Eligibility Instructions").

124. *See infra* notes 211-15 and accompanying text.

125. *See infra* notes 194-97 and accompanying text

126. *See infra* Part III.B.2 and accompanying text.

consistent, resulting in tax distinctions that run counter to fundamental notions of equity and efficiency.[127]

The federal and state governments have also struggled to determine the appropriate level of tax subsidy to provide for desirable software activities. Current tax preferences for software development recognize the important spillover effects or external benefits of the software industry, but many of these tax benefits have design flaws that minimize their effectiveness.

A. *Federal Tax Treatment of Software*

The federal government's tax treatment of software occurs largely within the context of the federal income tax. The United States federal government "derives the bulk of its revenue from an income tax."[128] A taxpayer's federal income tax liability for any given year is determined by applying the appropriate tax rates to the appropriate tax base.[129] The tax base for all taxpayers is taxable income, which is defined loosely as gross income minus deductions.[130] Once the applicable tax rates are applied to the taxpayer's taxable income, credits may be available to

---

127. *See infra* Part IV.A.

128. JOHN A. MILLER & JEFFREY A. MAINE, THE FUNDAMENTALS OF FEDERAL TAXATION: PROBLEMS AND MATERIALS 4 (2d ed. 2010). The United States does not have a federal sales tax. But "[t]he income tax has some of the features of a consumption tax. For example, earnings that are 'saved' in retirement accounts are not taxed currently. Instead those earnings are taxed years later when they are distributed from the accounts" and presumably consumed. *Id.*

129. *See* I.R.C. § 1 (2006). There are two rate structures: the ordinary income rate structure and the capital gains rate structure. The rate structure for ordinary income is progressive, meaning that as ordinary income increases, a taxpayer's tax liability also increases, but at a greater rate. *See* I.R.C. § 1(a)-(d). By contrast, the rate structure for capital gains (i.e., gains that arise from the sale of capital assets) is flat and significantly more favorable to the taxpayer. *See* I.R.C. § 1(h).

130. I.R.C. § 63(a). Gross income is a statutory term that does not have the same meaning as economic income. *See* I.R.C. § 61 (defining gross income broadly as "all income from whatever source derived"); I.R.C. §§ 101-140 (excluding particular kinds of receipts from gross income). Deductions are a "matter of legislative grace." INDOPCO, Inc. v. Comm'r, 503 U.S. 79, 84 (1992). The Code contains numerous provisions that authorize a deduction or that restrict or limit deductions. I.R.C. §§ 161, 261-280H.

reduce the amount of tax due.[131] The federal income tax treatment of computer software comes in the form of both deductions and credits, and differs depending on whether it deals with software development, acquisition, or transfer.[132]

1. *Software Development.* For federal income tax purposes, there is a 100% tax deduction available for software developments costs.[133] Specifically, all costs of developing computer software may be deducted in the year development costs are incurred,[134] provided such costs are incurred in connection with a trade or business.[135]

Computer software eligible for the 100% deduction is broadly defined as "any program or routine (that is, any sequence of machine-readable code) that is designed to cause a computer to perform a desired function or set of functions, and the documentation required to describe and maintain that program or routine."[136] Under this broad

131. *See* I.R.C. §§ 21-55AA (outlining various credits available).

132. *See infra* Part III.A.1-3.

133. I.R.C. § 174(a)(1) (permitting taxpayers to deduct immediately "research or experimental expenditures" that might otherwise have to be capitalized); Rev. Proc. 2000-50 § 5.01(2), 2000-2 C.B. 601, 603 (permitting taxpayers to treat the costs of developing computer software in a manner analogous to the treatment of research or experimental expenditures under Section 174).

134. I.R.C. § 174(a)(2)(A). If the taxpayer does not elect to deduct software development costs in the year they are incurred under Section 174(a), such costs may be amortized ratably over five years from the date of completion of the software's development under Section 174(b), or over three years from the date the software is placed in service under Section 167(f)(1). Rev. Proc. 2000-50 § 5.01(2), 2000-2 C.B. 601, 603.

135. I.R.C. § 174(a)(1). In *Saykally v. Comm'r*, 85 T.C.M. (CCH) 1401, 1403, 1410-11 (2003), *aff'd*, 247 F. App'x 914 (9th Cir. 2007), the Tax Court held that a software developer was not entitled to current deductions under Section 174 because he did not intend to market the developed technology himself, but rather, intended to market the technology through his wholly-owned corporation. The taxpayer did not have the objective intent to enter into a future business of his own with the developed technology. *Id.* at 1410. Rather, the taxpayer's purpose for engaging in the software development was to create the developed technology that could be licensed to the corporation for use in the corporation's existing business. *Id.*

136. Rev. Proc. 2000-50 § 2, 2000-2 C.B. 601, 601. Computer software includes:

> [A]ll forms and media in which the software is contained, whether written, magnetic, or otherwise. Computer programs of all classes, for example, operating systems, executive systems, monitors, compilers and translators, assembly routines, and utility programs as well as

definition, the current deduction applies regardless of intended use—whether for internal use or for commercial sale, lease, or license.[137] Thus, costs of developing routine accounting and management information systems and costs of developing billing systems or payroll systems are deductible even though they do not involve new or significantly improved programs and involve no uncertainty as to the software design or capability.[138]

The 100% tax deduction for software development costs represents a departure from normative tax principles in two significant ways. First, the current deduction applies even if the software yields benefits to the taxpayer beyond the year the development costs are incurred.[139] This is an exception

---

> application programs, are included. Computer software also includes any incidental and ancillary rights that are necessary to effect the acquisition of the title to, the ownership of, or the right to use the computer software, and that are used only in connection with that specific computer software.

Id.

137. The deduction applies even if the development costs are not experimental or investigative in a laboratory sense—general requirements that must be met for most research and developments costs to be deductible under Section 174. Treas. Reg. § 1.174-2(a)(1) (2011).

138. *Cf. id.* Before the Internal Revenue Service ("IRS") adopted its current position, it twice issued proposed regulations governing computer software development costs. In 1983, the IRS issued proposed regulations that would have permitted computer software developments costs to be deductible under Section 174 only if they were paid or incurred for developing "new or significantly improved computer software," a determination made with respect to "the computer program itself rather than the end use of the program." Prop. Treas. Reg. § 1.174-2(a)(3), 48 Fed. Reg. 2790, 2799 (Jan. 21, 1983). In 1989, the IRS issued revised proposed regulations under Section 174 softening the harsh "new or significantly improved" standard of the 1983 proposed regulations, and focusing instead on whether the product had met its basic design specifications related to function and performance level. Prop. Treas. Reg. § 1.174-2(a)(1), 54 Fed. Reg. 21224, 21225-26 (May 17, 1989). In 1993, these regulations were withdrawn, and the IRS announced that it would continue to apply its position in Revenue Procedure 69-21. 58 Fed. Reg. 15819, 15820 (Mar. 24, 1993).

139. *See* I.R.C. § 263(a)(1)(B) (providing that the capitalization under Section 163(a) does not apply to research or experimental expenditures deductible under Section 174); I.R.C. § 263A(c)(2) (providing that the uniform capitalization rules of Section 263A do not apply to any amount allowed as a deduction under Section 174). The regulations provide that capitalization is required for the costs of creating a "separate and distinct intangible asset," defined as "a property interest . . . of value . . . that is subject to protection under . . . law and the

to the widely accepted general principle that a taxpayer should not be able to immediately deduct an expenditure that produces benefits lasting beyond the current tax period.[140] Second, the deduction applies regardless of the intangible intellectual property rights covering the software program.[141] This is a departure from the government's tax approach for intellectual property development costs under which, for example, patent developments costs are generally deductible, but copyright creation costs are not.[142]

In addition to the 100% tax deduction, the federal government also offers a 20% tax credit for certain software development costs as an incentive for *increasing* research activities over time.[143] The credit is more limited than the tax deduction in several ways. First, the credit does not apply to total research spending; rather it only applies to software research spending above a base amount, which can normally be thought of as a firm's normal level of research spending.[144] The incremental nature of the credit was

---

possession and control of which is intrinsically capable of being sold, transferred or pledged . . . apart from a trade or business." Treas. Reg. § 1.263(a)-4(b)(1)(iii), (b)(3)(i) (2011). The regulations provide an exception, however, for software development, stating that software development costs are not treated as capitalized costs of creating a separate and distinct asset. *Id.* § 1.263(a)-4(b)(3)(iv).

140. *See* I.R.C. § 263(a)(1); Treas. Reg. § 1.263(a)-4(b)(1)(ii) to (iii). *But see* Treas. Reg. § 1.263(a)-4(b)(3)(iv), (b)(4).

141. *See* Treas. Reg. § 1.263(a)-4(b)(3)(iv), (b)(4) (making no reference to the intellectual property rights covering the software program); *see also* Rev. Proc. 2000-50 § 2, 2000-2 C.B. 601, 601 (same).

142. JEFFREY A. MAINE & XUAN-THAO NGUYEN, INTELLECTUAL PROPERTY TAXATION: TRANSACTION AND LITIGATION ISSUES 174, 184 (2003).

143. I.R.C. § 41(a); Economic Recovery Tax Act of 1981, Pub. L. No. 97-34, § 221(a), 95 Stat. 172, 241 (establishing original research credit at I.R.C. § 44F (1981)). The credit applies to "qualified research," defined as research (1) with respect to which expenditures may be treated as expenses under Section 174; (2) that is "undertaken for the purpose of discovering information" that is "technological in nature," and "the application of which is intended to be useful in the development of a new or improved business component of the taxpayer"; and (3) "substantially all of the activities of which constitute elements of a process of experimentation" that relates to a new or improved function, performance, reliability, or quality. I.R.C. § 41(d)(1).

144. The credit is 20% of qualified research spending in excess of a "base amount," which, pursuant to a complicated formula, is an estimate of the amount of gross receipts a taxpayer would normally expect to spend on qualified

designed to encourage developers to increase their research spending levels over time.[145] Second, the credit does not apply to the development of internal-use software,[146] unless the internal-use software is used in qualified research, is used in a plant process, or meets a "high threshold of innovation" test.[147] The federal government has issued

---

research. *See* I.R.C. § 41(a)(1). As an alternative to using the incremental 20% credit, a taxpayer may elect to use a 14% "alternative simplified credit" (14% of the amount by which qualified research expenses exceed 50% of the average research expenses for the three preceding years). I.R.C. § 41(c)(5) (2006 & West 2011).

145. *See* STAFF OF JOINT COMM. ON TAXATION, 97TH CONG., GENERAL EXPLANATION OF THE ECONOMIC RECOVERY TAX ACT OF 1981, at 120 (Comm. Print 1981) ("The new credit applies only to increases in qualified research expenditures, in order to encourage enlarged research efforts by companies which already may be engaged in some research activities.").

146. I.R.C. § 41(d)(4)(E) (2006). The statute provides that "[e]xcept to the extent provided in regulations, any research with respect to computer software which is developed by (or for the benefit of) the taxpayer primarily for internal use by the taxpayer" is excluded from the definition of qualified research. *Id.*; *see also* Tax Reform Act of 1986, Pub. L. No. 99-154, § 231(b), 100 Stat. 2085, 2173 (stating that adapting or duplicating existing business components cannot receive credit).

147. I.R.C. § 41(d)(4)(E) (noting that Treasury Regulations provide an exception for certain internal use software). Under the regulations, internal use software constitutes qualified research if the research satisfies a three-part, high threshold of innovation test. Specifically, under final regulations that were suspended in January 2001, software satisfies this test if:

> (A) The software is innovative in that the software is intended to result in a reduction in cost, improvement in speed, or other improvement, that is substantial and economically significant;
>
> (B) The software development involves significant economic risk in that the taxpayer commits substantial resources to the development and there is a substantial uncertainty, because of technical risk, that such resources would be recovered within a reasonable period; and
>
> (C) The software is not commercially available for use by the taxpayer . . . .

Treas. Reg. § 1.41-4(c)(6)(vi) (2001). According to a new set of regulations that were proposed in December 2001, software satisfies the first prong of the test only if "the software is intended to be unique or novel and is intended to differ in a significant and inventive way from prior software implementations or methods." Prop. Treas. Reg. § 1.41-4(c)(6)(vi)(A), 66 Fed. Reg. 66362, 66371 (Dec. 26, 2001). Until final regulations are issued, taxpayers may rely on the prior suspended regulations or the new proposed regulations. 66 Fed. Reg. at 66366-67.

conflicting definitions of internal-use software, and has yet to provide clear guidance to taxpayers.[148] Finally, the 20% tax credit is only temporary.[149]

It might be possible for software development costs to qualify for the 20% tax credit as well as the 100% tax deduction. In such a case, to the extent the credit is taken, deductions must be reduced.[150] For example, assume a company expends $100,000 during the year on software development (credit-eligible expenses), and that the base period amount (normal level of research spending) is $60,000. The company is allowed a tax credit equal to 20% of the $40,000 increase in research expenditures, or $8000. The company's research and development deduction is reduced by the $8000 credit, leaving a deduction of $92,000.

These tax preferences for software development—a 100% tax deduction and 20% tax credit—contain no restriction on the location of the resulting intellectual property.[151] Thus, even though software developed in the United States is eligible for federal tax incentives, it is not required to stay in the United States, and can be transferred to low-tax countries, such as Ireland, for subsequent software manufacturing and exploitation of

---

148. Under final regulations that were suspended in January 2001, "[s]oftware is developed primarily for the taxpayer's internal use if the software is to be used internally, for example, in general administrative functions of the taxpayer (such as payroll, bookkeeping, or personnel management) or in providing noncomputer services (such as accounting, consulting, or banking services)." Treas. Reg. § 1.41-4(c)(6)(iii). Under a new set of proposed regulations issued in December 2001, "software is presumed developed by (or for the benefit of) the taxpayer primarily for the taxpayer's internal use" unless it "is developed to be commercially sold, leased, licensed, or otherwise marketed, for separately stated consideration to unrelated third parties." Prop. Treas. Reg. § 1.41-4(c)(6), 66 Fed. Reg. 66362, 66371 (Dec. 26, 2001). Until final regulations are issued, taxpayers may rely on the prior suspended regulations or the new proposed regulations. 66 Fed. Reg. at 66366-67.

149. I.R.C. § 41(h)(1)(B) (West 2011). The credit has been continually renewed as a temporary provision: President Barack Obama proposed making the credit permanent in his Fiscal Year 2010 and 2011 budgets. *See* NAT'L ECON. COUNCIL ET AL., A STRATEGY FOR AMERICAN INNOVATION: SECURING OUR ECONOMIC GROWTH AND PROSPERITY 4 (2001).

150. I.R.C. § 280C(c)(3) (2006).

151. DELOITTE, 2010 GLOBAL SURVEY OF R&D TAX INCENTIVES 34-35 (2011).

software development results.[152] To prevent off-shoring of software manufacturing and production, an additional tax benefit is available to companies that manufacture or produce software in whole or in significant part within the United States.[153] The tax benefit is an exclusion of 9% of net income from U.S. production activities, which includes the production of software.[154] The tax benefit, added by Congress in 2004, is an incentive to retain software

---

152. Ireland is a popular tax jurisdiction for the migration of software offshore. It has a low 12.5% tax rate on active business income, an extensive treaty network, and a well-educated, relatively inexpensive workforce. *See* Joseph B. Darby III & Kelsey Lemaster, *Double Irish More Than Doubles the Tax Saving: Hybrid Structure Reduces Irish, U.S. and Worldwide Taxation*, PRACTICAL US/INTERNATIONAL TAX STRATEGIES, May 15, 2007, at 2, 12. Other popular tax jurisdictions are Switzerland and The Netherlands. *See* Joseph B. Darby III, *Everyone Goes to Zug: Tax Planners Benefit as Swiss Cantons Compete for Lowest Tax Rates*, PRACTICAL EUROPEAN TAX STRATEGIES, June 2006, at 2; Joseph B. Darby III, *Dutch Treat: How the Netherlands Has Prospered by Offering Attractive Tax Planning Opportunities*, PRACTICAL EUROPEAN TAX STRATEGIES, May 2006, at 2.

153. American Jobs Creation Act of 2004, Pub. L. No. 108-357, § 199(a), 118 Stat. 1418, 1421.

154. The tax benefit is actually a deduction "equal to nine percent of the lesser of (A) the qualified production activities income of the taxpayer for the taxable year, or (B) taxable income . . . for the taxable year." I.R.C. § 199(a) (2006). The amount of the deduction is limited annually to 50% of the W-2 wages paid by the taxpayer during the calendar year that ends in such taxable year, which creates an incentive to obtain services from employees rather than outsourcing work to independent contractors. I.R.C. § 199(b)(1).

The taxpayer's Section 199 deduction depends primarily on the amount of the taxpayer's "qualified production activities income" which is defined as "domestic production gross receipts" minus the cost of goods sold allocable to such receipts and other expenses, losses, or deductions that are properly allocable to such receipts. I.R.C. § 199(c)(1). A taxpayer's "domestic production gross receipts" are the taxpayer's gross receipts derived from one of several listed activities—for example, gross receipts "derived from any lease, rental, license, sale, exchange, or other disposition of qualifying production property which was manufactured [or] produced . . . by the taxpayer in whole or in significant part within the United States." I.R.C. § 199(c)(4)(A). "Qualifying production property" is defined as any "tangible personal property," any "*computer software*," and any sound recordings. I.R.C. § 199(c)(5) (emphasis added). The Section 199 deduction may be available to computer software updates, as well as to some online software and online games. T.D. 9262, 2006-1 C.B. 1040, 1041.

manufacturing and production jobs in the United States.[155] While the 9% exclusion has had the effect of reducing taxes for certain U.S. software companies, many U.S. software developers still find significant tax benefits in off-shoring software production and manufacturing.[156]

2. *Software Acquisitions: Tax Consequences to the Purchaser/Licensee.* Technology companies often acquire software relevant to their business needs to avoid bad software patent litigation.[157] In 2011, for example, a consortium of technology companies, which included Apple and Microsoft, paid $4.5 billion in cash for more than 6000 software patents from the telecommunications equipment maker Nortel Networks.[158] In the same year, Google purchased 1000 software patents from IBM for a broad range of applications.[159]

In contrast to federal tax benefits for software development, tax incentives for software acquisition come in

---

155. Senator Charles Grassley, Chairman of the Committee of Finance, stated that Section 199 "tips the scales of global competiveness more in favor of American businesses," does "more to help to maintain and create jobs in the United States than any law in decades," and "contains far-reaching measures to revise the manufacturing base in America." Press Release, Senator Chuck Grassley, Chairman of the Comm. on Fin., Grassley Praises President's Signing of Business Tax Relief, Key Reforms into Law (Oct. 22, 2004) (internal quotation marks omitted), *available at* http://finance.senate.gov/newsroom/chairman/release/?id=69713724-e275-42a9-982f-3328884909e4. "In 2005, the first year the [Section 199] deduction was available to taxpayers, . . . the total deduction claimed on corporation income tax returns was approximately $9.3 billion." Beth M. Benko, *Section 199: Deduction Related to Income Attributable to Domestic Production Activities*, 510-2d TAX MGMT. PORTFOLIOS (BNA), at A-6 (2006).

156. *See infra* notes 292-95 and accompanying text.

157. Evelyn M. Rusli, *In Battle for Patents, Google Buys a Batch from I.B.M.*, DEALBOOK (July 29, 2011, 1:51 PM), http://dealbook.nytimes.com/2011/07/29/in-battle-for-patents-google-buys-a-batch-from-i-b-m/ (statement from Google).

158. John Ribeiro, *Apple, Microsoft Consortium Beats Google for Nortel Patents*, PC WORLD (July 1, 2011, 2:00 AM), http://www.pcworld.com/businesscenter/article/234887/apple_microsoft_consortium_beats_google_for_nortel_patents.html.

159. Paul McDougall, *Google Acquires 1,000 IBM Patents*, INFORMATIONWEEK (July 29, 2011, 12:40 PM), http://www.informationweek.com/news/internet/google/231002937.

the form of tax deductions only (and not tax credits).[160] If software is acquired by non-exclusive license (under which the acquirer receives the limited right to use the software), and the software is used in the licensee's business or investment activity, the annual license fees are deducted when paid or incurred.[161]

However, the costs of *purchasing* software—including acquisitions by exclusive license, which are treated as purchases—generally must be capitalized.[162] Software purchasers can then recover their costs over time through tax depreciation deductions. As a general rule, the method of depreciation for the purchase of any type of property depends on whether the property is tangible or intangible. As described below, the government artificially characterizes the software as either tangible or intangible, depending on a number of factors. These factors include (1)

---

160. The investment tax credit, enacted in 1962 and repealed in 1986, provided a credit for investment in certain tangible property. I.R.C. §§ 38, 48 (Supp. 1964), *repealed by* Tax Reform Act of 1986, Pub. L. No. 99-514, 100 Stat. 2085, 2166. Neither the Code nor the treasury regulations addressed whether software qualified as tangible property and, thus, fell within the scope of the credit. *See* Treas. Reg. § 1.48-1(c) (2009) (providing definition of "tangible personal property"); *Id.* § 1.48-1(f) (providing that intangible property did not qualify for the credit). The IRS struggled with the issue. *See* Rev. Proc. 69-21, 1969-2 C.B. 303 (characterizing computer software as intangible property ineligible for the investment tax credit). *But see* Rev. Rul. 71-177, 1971-1 C.B. 5 (concluding that bundled software was eligible for the investment tax credit). And there was considerable diversity of opinion among the courts. *Compare* Ronnen v. Comm'r, 90 T.C. 74, 96-100 (1988) (holding software was intangible property ineligible for the investment tax credit), *with* Comshare v. United States, 27 F.3d 1142, 1142-43 (6th Cir. 1994) (holding software constituted tangible property eligible for tax credits), *and* Norwest Corp. v. Comm'r, 108 T.C. 358, 375-76 (1997) (holding software was tangible personal property qualifying for the investment tax credit). For an analysis of these decisions, see Eric W. Castillo, Note, *Federal Tax Treatment of Computer Software Under* Norwest v. Commissioner, 26 RUTGERS COMPUTER & TECH. L.J. 157 (1999). Although repealed in 1986, the investment tax credit could resurface as tool to stimulate the economy and the proper classification of software as tangible or intangible would likewise resurface.

161. I.R.C. §§ 162(a), 212 (2006); Treas. Reg. § 1.162-11 (2011). If, however, the software is being used to create an asset with a useful life in excess of one year, the license fee must be capitalized and depreciated as part of the asset. *See* Comm'r v. Idaho Power Co., 418 U.S. 1, 16-19 (1974).

162. Treas. Reg. § 1.263(a)-4(b)(1)(i), (c)(1)(xiv) (2011). An exception exists for off-the-shelf software, as discussed *infra* notes 179-82 and accompanying text.

whether the acquired software is bundled with hardware or non-bundled, (2) whether the software is off-the-shelf software or custom software, and (3) whether the software was acquired separately or along with business assets.[163]

The government has long taken the position that software that is bundled with hardware or other tangible property without a separately stated cost should be depreciated under the rules for depreciating *tangible* property.[164] Treating bundled software as tangible property provides significant tax benefits to software purchasers—namely the availability of accelerated depreciation deductions[165] and, in some cases, the availability of immediate expensing of software purchase costs.[166] These bonus depreciation rules, which are applicable only to tangible property, apply an arbitrary cost recovery system that generally allows purchasers to recover their costs before the property ceases to be useful in their business or income-producing activity.

---

163. *See infra* notes 164-82 and accompanying text.

164. Treas. Reg. § 1.167(a)-14(b)(2) (2011) ("The cost of acquiring an interest in computer software that is included, without being separately stated, in the cost of the hardware or other tangible property is treated as part of the cost of the hardware or other tangible property that is capitalized and depreciated under other applicable sections of the Internal Revenue Code."); *Id.* § 1.197-2(g)(7) ("[S]ection 197 does not apply to the cost of an interest in computer software to the extent such cost is included, without being separately stated, in the cost of the hardware or other tangible property and is consistently treated as part of the cost of the hardware or other tangible property."); *see also* Rev. Proc. 2000-50 § 6.01(1), 2000-2 C.B. 601, 601; Rev. Rul. 71-177, 1971-1 C.B. 5 (holding the cost of a computer included the cost of the software provided with it for purposes of tax depreciation allowed under Section 167 of the Code and the investment tax credit allowed under Section 38 of the Code).

165. I.R.C. § 168. Under accelerated depreciation rules, tangible property is depreciated using an accelerated depreciation method (e.g., 200 percent declining balance method as opposed to the straight-line method) over short, arbitrary recovery periods (e.g., three-, five-, or seven-year recovery periods as opposed to useful life). *Id.*

166. I.R.C. § 179. Section 179, which was enacted to encourage investment in productive property, allows one to elect to write off the cost of acquisition of "[S]ection 179 property" as an expense "not chargeable to capital account." I.R.C. § 179(a). Section 179 property is generally tangible property that is purchased for the "active conduct of a trade or business." I.R.C. § 179(d)(1). There are limits on the amount that can be expensed in any given year. I.R.C. § 179(b) (West 2011).

In contrast, non-bundled software is subject to the depreciation rules governing *intangible* property,[167] with certain exceptions that will be discussed below.[168] Taxpayers seeking to recover the costs of acquiring intangible property must use the non-accelerated straight-line method of depreciation and a lengthy fifteen-year recovery period.[169] Specifically included within the scope of the intangible rule is computer software,[170] regardless of intellectual property

167. I.R.C. § 197 (2006) ("Amortization of goodwill and certain other intangibles[.]").

168. Two important exceptions discussed here are I.R.C. § 197 (providing an immediate deduction for off-the-shelf software) and I.R.C. § 167(f)(1) (providing a three-year depreciation rule for separately acquired software).

169. I.R.C. § 197, *added by* Revenue Reconciliation Act of 1993, Pub. L. No. 103-66, 107 Stat. 416, 532-33. Section 197 provides a list of intangible assets that fall within the definition of "[S]ection 197 intangible" and are subject to fifteen-year amortization. I.R.C. § 197(d). The Section also specifically excludes certain intangible assets. I.R.C. § 197(e). Section 197 reflects Congress's attempt to simplify tax depreciation rules for intangible property. Under historic tax depreciation rules, the capitalized costs of acquiring intangibles could be recovered over the intangible's useful life provided the intangible had a limited useful life that could be determined with reasonable accuracy. *See* STAFF OF JOINT COMM. ON TAXATION, 103D CONG., TECHNICAL EXPLANATION OF THE TAX SIMPLIFICATION ACT OF 1993, at 147 (Comm. Print 1993) (explaining that Congress created Section 197 to eliminate confusion over the federal tax treatment of intangible assets).

170. I.R.C. § 197(d)(1)(C)(iii), (e)(3); Treas. Reg. § 1.197-2(b)(5) (2011) (including computer software as a Section 197 intangible). Computer software is defined broadly in the rule as "any program designed to cause a computer to perform a desired function." I.R.C. § 197(e)(3)(B). This "includes all forms and media in which the software is contained, whether written, magnetic, or otherwise. [Also included are] operating systems, executive systems, monitors, compilers and translators, assembly routines, and utility programs as well as application programs . . . ." Treas. Reg. § 1.197-2(c)(4)(iv). This broad definition is modified to exclude:

> [A]ny data or information base . . . unless the data base or item is in the public domain and is incidental to a computer program. . . . [A] copyrighted or proprietary data or information base is treated as in the public domain if its availability through the computer program does not contribute significantly to the cost of the program.

Id.; *see also* H.R. REP. NO. 103-213, at 680 (1993), *reprinted in* 1993 U.S.C.C.A.N. 1088, 1369. Moreover, "[c]omputer software also includes any incidental and ancillary rights that are necessary to effect the acquisition of title to, the ownership of, or the right to use the computer software, and that are used only in connection with that specific computer software." Treas. Reg. 1.197-2(c)(4)(iv).

protection (as a copyright, trade dress, trade secret, or patent).[171]

While the fifteen-year depreciation rule for non-bundled software seems harsh (as compared to the more favorable rules for bundled software), Congress carved out two important exceptions: off-the-shelf software[172] and software acquired outside the context of a business acquisition.[173] These types of software are depreciated over three years (as opposed to fifteen years) using the straight-line method.[174]

There are two apparent justifications for carving out a short three-year recovery period for readily available software and separately acquired software. First, computer software differs significantly from other forms of intangibles in that its value is ascertainable and it has a measurable useful life.[175] As such, a lengthy fifteen-year amortization period would bear no resemblance to the actual useful life of software; for example, Microsoft's word processing program "Word," which was introduced in 1983, saw "four new versions and three major upgrades" in the ten years subsequent to release.[176] Second, a lengthy fifteen-year depreciation period would exact a penalty on U.S. companies that extensively use computer software in their operations.[177] A short recovery period provides some parity with the many other countries that offer either a low recovery period of depreciation (e.g., periods ranging from two to five years on a straight-line basis), or an accelerated

---

171. I.R.C. §197(d)(1)(C); Treas. Reg. § 1.197-2(b)(5).

172. Section 197 does not apply to off-the-shelf software—software that is "readily available for purchase by the general public, is subject to a non-exclusive license, and has not been substantially modified." I.R.C. § 197(e)(3)(A).

173. Section 197 does not apply to any interest in computer software that is not acquired as part of a purchase of a trade or business. I.R.C. § 197(e)(3)(A)(ii), -(e)(4); Treas. Reg. § 1.197-2(c)(7).

174. I.R.C. § 167(f), *added by* Revenue Reconciliation Act of 1993, Pub. L. No. 103-66, 107 Stat. 416, 538; *see* H.R. REP. NO. 103-213, at 680.

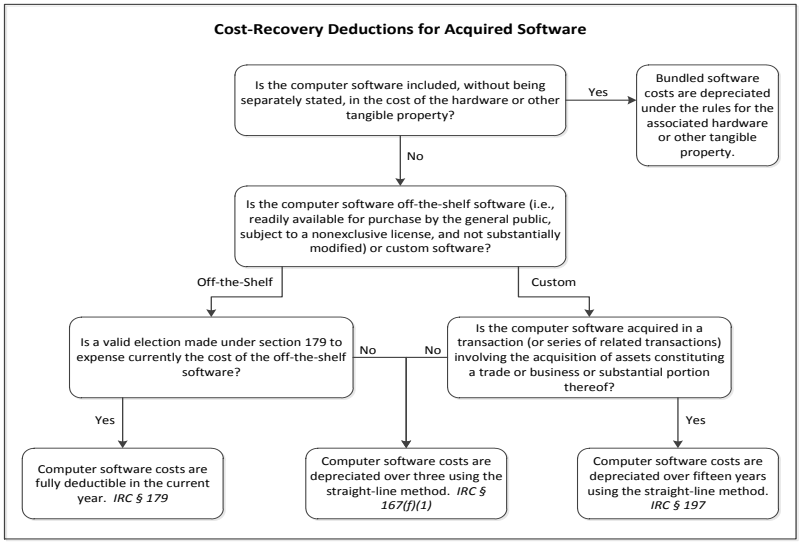
175. *Tax Treatment of Intangible Assets: Hearing on S. 1245, H.R. 3035, and H.R. 4210 Before the S. Comm. on Finance*, 102d Cong. 146-48 (1992) (statement of the Coalition for Fair Treatment of Intangibles).

176. *Id.* at 51.

177. *Id.* at 36-37 (statement of William P. Benac, Treasurer, Electronic Data Systems).

depreciation method (e.g., declining balance depreciation) for purchased software.[178]

In 2003, Congress enacted a temporary measure that allows purchasers to immediately deduct the cost of off-the-shelf software, rather than depreciating over three years.[179] It modified its classification of off-the-shelf software from intangible property to tangible property, thereby making it eligible for 100% expensing in year of purchase.[180] Although 100% expensing for off-the-shelf software is an approach consistent with that of many countries,[181] absent congressional action, it will expire in 2012.[182] The following chart illustrates current federal tax depreciation rules for capitalized software purchase costs:



178. *See* Jacek Warda, *Tax Treatment of Business Investments in Intellectual Assets: An International Comparison* 38 (Organisation for Economic Co-operation and Development, Working Paper No. 4, 2006), *available at* http://www.oecd.org/dataoecd/53/4/36764076.pdf.

179. I.R.C. § 179(d)(1) (West 2011), *added by* Jobs and Growth Tax Relief and Reconciliation Act of 2003, Pub. L. No. 108-27, 117 Stat. 752, 757.

180. *Id.* Section 179 property is generally tangible, depreciable, personal property that is acquired for use in the active conduct of a trade or business. *Id*.

181. Warda, *supra* note 178, at 38 (noting a 100% write off for off-the-shelf software is offered in a number of countries, such as Australia, Canada, Denmark, Finland, France, Greece, Mexico, and the United Kingdom).

182. Only off-the-shelf software purchased in a tax year beginning after 2002 and before 2013 qualifies for the 100% expensing. I.R.C. § 179(d)(1)(A)(ii).

3. *Software Sales and Licenses: Tax Consequences to the Seller/Licensor*. For federal income tax purposes, the transfer of computer software typically produces income to the transferor that is taxed at either ordinary income tax rates or a reduced capital gains tax rate.[183] As explained below, a number of factors determine which tax rate—ordinary income or capital gain—will be employed, including whether the software was bundled or non-bundled, whether the transfer was a non-exclusive license or exclusive license, and whether the software was inventory or used in taxpayer's trade or business. The tax treatment of a software transfer can also be impacted by the intellectual property covering the software (patent, copyright, trade secret, etc.).[184]

The transfer of software and hardware together in a bundled package is taxed as a transfer of the hardware.[185] Thus, gain on the sale of hardware that is inventory in the seller's hands is taxed as ordinary income.[186] Otherwise, the gain is most likely characterized as capital gain,[187] except to the extent of any depreciation deductions previously taken on the hardware.[188]

The taxation of transfers of non-bundled software requires unique considerations. The first consideration is whether the transfer is a license or sale for federal tax purposes. If computer software is transferred by *nonexclusive license*, the federal tax consequences are straightforward: the licensor must report payments received in full as ordinary income and may not recover any

183. Ordinary taxable income is subject to graduated tax rates, ranging from a low of 10% (for individuals) or 15% (for corporations) to a high of 35% (for both individuals and corporations). I.R.C. § 1(i)(1)-(2) (2006 & Supp. 2010); I.R.C. § 11 (2006). By contrast, most capital gains are taxed at a preferential rate of 15% (for individuals). I.R.C. § 1(h).

184. *See infra* notes 185-97 and accompanying text.

185. *See supra* notes 164-66 and accompanying text.

186. I.R.C. § 1221(a)(1) (excluding inventory from the definition of capital asset eligible for preferential capital gains tax treatment); I.R.C. § 1231(b)(1)(A)-(B) (excluding inventory from the definition of quasi-capital asset).

187. I.R.C. §§ 1222, 1231.

188. I.R.C. § 1245.

remaining basis in the software.[189] If, on the other hand, computer software is transferred by *sale* or *exclusive license* (which constitutes a sale for federal tax purposes), the transferor is permitted to recover any remaining basis in the software and may be entitled to preferential capital gains treatment.[190]

Whether a particular transfer is treated as a license (which results in ordinary income to the transferor) or a sale (which may or may not result in capital gains treatment) is not always an easy determination. As a general rule, the transfer of all substantial rights in computer software, including the right to produce and modify the software, constitutes a sale for federal income tax purposes.[191] A sale will be found, for example, when the transferor transfers "all rights, title to, and interest in and to the [software], including without limitation, all source and object code and manuals and all other related

---

189. *See* I.R.C. § 61(a)(6). A taxpayer may recover tax free any unrecovered cost of, or remaining basis in, software, only upon a "sale or other disposition" of software within the meaning of Section 1001. I.R.C. § 1001(a). A mere nonexclusive license of software is not a sale under general tax principles. As continuing owner of the software, however, the licensor may continue to depreciate the software.

190. Exclusive licenses are treated as sales for tax purposes because the transferor is transferring the exclusive right to make, use, and sell for the life of the software. *See, e.g.*, Rollman v. Comm'r, 244 F.2d 634, 639-41 (4th Cir. 1957); Watson v. United States, 222 F.2d 689, 692 (10th Cir. 1955). Gain on a sale is the excess of the amount realized over the software's unrecovered cost or basis in the software. I.R.C. § 1001(a). Such gain may be entitled to preferential capital gain treatment because a requirement for special rate treatment is a "sale or exchange" of property. I.R.C. § 1222(3) ("The term 'long-term capital gain' means gain from the sale or exchange of a capital asset held for more than 1 year . . . .").

191. Under general principles of tax law, a transfer of intellectual property will qualify as a sale, as opposed to a license, if the transferor assigns all substantial rights to the intellectual property (or the rights retained by the transferor are of no substantial value). *See* I.R.C. § 1235 (providing sale treatment results when "all substantial rights" in a patent are transferred); Rev. Rul. 64-56, 1964-1 C.B. 133, *amplified by* Rev. Rul. 71-564, 1971-2 C.B. 179, 179-80 (promulgating an objective substantial rights test for trade secrets and know how); Rev. Rul. 60-226, 1960-1 C.B. 26, 26 (ruling that a transfer of a copyright will be considered a sale, rather than a license, whenever the transferor transfers "the exclusive right to exploit the copyrighted work throughout the life of the copyright in a medium of publication"). For a case involving a software transfer, see *Levy v. Comm'r*, 64 T.C.M. (CCH) 534 (1992).

documentation and materials therefore and all enhancements now existing or hereafter made thereto."[192] A license will be found when the transferor transfers less than all the rights to the software (e.g., the object code, but not the source code, is transferred).[193]

Assuming a sale has occurred, determining the character of the gain or loss can be difficult in light of the fact that software may be protected as a patent, a copyright, a trade secret, or a combination of these. The tax consequences depend on whether the transfer was a sale of a capital asset or was the sale of an asset that is excludable from the definition of capital asset under the Code,[194] which, in turn, depends on the method of computer software protection. For individuals, self-developed software protected only as a copyright receives ordinary income treatment under general characterization principles.[195] For individuals, self-developed software sold as a patent may receive preferential capital gains treatment under a special characterization provision.[196] As previously discussed, the

192. *Levy*, 64 T.C.M. (CCH) at 535. In *Levy*, the issue did not involve whether the software transfer constituted a sale, but rather whether the transfer was a sale of a capital asset or was the sale of an asset which is excludable from the definition of capital asset under the Code. *Id.*

193. *Id.* at 536.

194. *See id.*

195. I.R.C. § 1222 (requiring for preferential capital gains treatment the sale or exchange of a capital asset held for more than one year); I.R.C. §§ 1221(a)(3), 1231(b)(1)(C) (excluding from the definition of capital asset self-created copyrights). An important question arises whether the capital asset exclusion for self-developed copyrights (Sections 1221(a)(3) and 1231(b)(1)(C)) applies to self-developed computer software that is protected as both a copyright and a patent. According to the regulations, the copyright exclusion does not apply if a patent, an invention, or a design "may be protected only under the patent law and not under the copyrightable law." Treas. Reg. § 1.1221-1(c)(1) (2011). Hence, computer software that is copyrightable but also protected as patent would be subject to the copyright exclusion (if the sale of the self-developed computer software did not qualify for capital gains treatment under the special character rule of Section 1235). *Id.* § 1.1235-1(b) ("If a transfer is not one described in . . . this section, section 1235 shall be disregarded in determining whether or not such transfer is a sale or exchange of a capital asset.").

196. I.R.C. § 1235 (providing all the requirements under general characterization provisions—sale or exchange, capital asset, and requisite holding period—for preferential capital gains treatment). Although Section 1235 refers to a patent, self-developed computer software that is patented or that is patentable should also qualify under the Section 1235 safe harbor. Treas. Reg.

method of software protection does not dictate the federal tax treatment of development or acquisition costs.[197] It is questionable whether the method of protection should dictate tax consequence of a sale, given its irrelevance on the development or acquisition side.

B. *State Tax Treatment of Computer Software*

States raise revenue through various taxes. A majority of the states impose an income tax,[198] which is generally based on the taxable income reported on the taxpayer's federal return.[199] Although federal taxable income is the starting point in determining the state income tax, states have their own income rate structures, which are typically lower than federal income tax rates.[200] In addition, states offer their own tax credits, which are often variations of tax credits available for federal income tax purposes. The majority of the states, for example, provide a research tax credit, similar to the federal research tax credit described above, for software development projects in their states.[201] And some states offer an investment tax credit, based on

---

§ 1.1235-2(a) ("The term *patent* means a patent granted under the provisions of title 35 of the United States Code . . . . It is not necessary that the patent or patent application for the invention be in existence if the requirements of section 1235 are otherwise met."). Therefore, the transfer by a holder of all substantial rights to a software patent should qualify for long-term capital gains treatment.

197. *See supra* Part III.A.1-2.

198. Nine states have no state income tax: Alaska, Florida, New Hampshire, Nevada, South Dakota, Tennessee, Texas, Washington, and Wyoming. *See States Without a State Income Tax*, INTERNAL REVENUE SERVICE, http://www.irs.gov/efile/article/0,,id=130684,00.html (last updated Mar. 31, 2011).

199. *See* JEROME R. HELLERSTEIN & WALTER HELLERSTEIN, STATE TAXATION ¶ 20.02 (3d ed. 1998 & 2011 Supp.).

200. *See State Individual Income Tax Rates, 2000-2011*, THE TAX FOUNDATION (Mar. 3, 2011), http://www.taxfoundation.org/taxdata/show/228.html.

201. *See State R&D Tax Benefits*, WARNER ROBINSON LLC, http://www.warner-robinson.com/rd-tax-credit/state-benefits (last visited Nov. 10, 2011) (noting that thirty-three states currently provide for a research and development tax credit). New York, for example, offers a research and development tax credit "equal to 10 percent of the federal R&D credit for the portion attributable to R&D conducted in . . . New York." *State Tax Credits*, WARNER ROBINSON LLC, http://www.warner-robinson.com/state-tax-credits (last visited Nov. 10, 2011).

the federal investment credit that was repealed by Congress in 1986, for purchases of tangible property (e.g., buildings, machinery, or equipment) used to develop computer software.[202] States generally apply the federal rules to determine what software costs qualify for state income tax credits.[203]

Most states also raise significant revenue by imposing a sales tax.[204] A sales tax is a consumption tax charged for the purchase[205] of many goods and some services, and is determined by applying a percentage rate to the sales price for the item.[206] To enhance compliance, states imposing a sales tax require sellers (with sufficient nexus) to collect the sales tax from customers, then remit the tax to the state.[207]

---

202. *See, e.g.*, N.Y. TAX LAW § 210(12) (McKinney 2005 & Supp. 2011). New York's investment tax credit equals 5% of the cost of qualified property up to $350,000,000, and 4% of the cost of qualified property in excess of $350,000,000. *Id.*; *State Tax Credits*, *supra* note 201.

203. *See State R&D Tax Benefits*, *supra* note 201. ("[S]tates [with R&D tax credits or incentives for R&D spending] generally follow the federal regulations and IRS guidance on what constitutes qualified research expenses."). As an example, in determining whether software developed primarily for the taxpayer's internal use qualifies for a state's research and development tax credit, federal treasury regulations generally must be often be consulted. As a further example, in determining whether purchased software qualifies for a state's investment tax credit, federal rules and precedent must often be consulted.

204. All states except Alaska, Delaware, Montana, New Hampshire, and Oregon, collect sales taxes. *See* HELLERSTEIN & HELLERSTEIN, *supra* note 199, ¶ 12.02, tbl.12.1.

205. As Rendleman and Neely have stated:

> Technically speaking, software is rarely 'purchased,' despite the common use of that term. Rather, software is licensed to the customer . . . . Nevertheless, courts have held these transactions to be sales. Furthermore, many state statutes broadly define 'sale' to include licenses . . . . Thus, licenses for use subject to the terms of the licensor generally are sales for purposes of sales and use tax.

Nancy S. Rendleman & Charles B. Neely, Jr., *States Seek to Broaden Sales and Use Taxation of Computer Software*, J. MULTISTATE TAXATION, Nov.-Dec. 1992, at 196, 197 n.3.

206. *See* HELLERSTEIN & HELLERSTEIN, *supra* note 199, ¶¶ 12.01, 12.02 & tbl.12.1 (providing for each state with a sales tax both the "rate of tax" and the "basis or measure of tax").

207. *Id*. ¶ 12.01 ("[S]ales taxes are collected from the purchaser by the seller . . . ."). Most states that impose a sales tax also impose a use tax, which is

1. *General Sales Tax Principles.* States that have a sales tax generally impose their sales tax on tangible personal property, and exempt intangible property and most services.[208] Applying this sales tax framework to computer software transactions has presented unique problems because computer software possesses characteristics of both tangible and intangible property, as well as services. Considerable controversy has focused on questions such as: Are software transactions considered sales of property or services? If viewed as sales of property, are software transactions considered sales of tangible property or intangible property? Does the characterization change if the software is delivered electronically? States have not been uniform in their resolution of these questions.

Early on, states did not specifically address the taxability of computer software, instead leaving it up to administrative bodies and courts to determine classification (i.e., whether software is included or excluded within the general definition of tangible personal property subject to sales taxation).[209] In recent years, however, states have used legislation or regulations to address the taxability of software.[210]

Many states now make a tax distinction between "canned" software and "custom" software.[211] In these states,

---

imposed directly on buyers who purchased goods without paying a sales tax (e.g., item purchased in another state without a sales tax). *See, e.g.*, Tex. Comptroller of Pub. Accounts, Hearing No. 36,957 (1999) (ruling that computer software purchased outside of Texas is subject to Texas use tax when used in Texas); Tax Policy Div., Tex. Comptroller of Pub. Accounts, Ltr. Rul. 200609749L (2006).

208. *See* HELLERSTEIN & HELLERSTEIN, *supra* note 199, ¶ 12.04. For an earlier treatment, see Clyde L. Ball, *What is a Sale for Sales Tax Purposes?*, 9 VAND. L. REV. 227, 228-29 (1956).

209. Court decisions have not always been consistent. *Compare* Commerce Union Bank v. Tidwell, 538 S.W.2d 405 (Tenn. 1976) (finding software to be intangible), *with* Comptroller of the Treasury v. Equitable Trust Co., 464 A.2d 248 (Md. 1983) (finding software to be tangible). For an analysis of early court decisions and state legislative response, see Rendleman & Neely, *supra* note 205, at 199-200.

210. *See, e.g.*, Rendleman & Neely, *supra* note 205 (noting state legislative responses to early court decisions). For specific examples of recent legislative or regulatory response, see *infra* notes 211-35.

211. States that make the distinction provide their own definitions of canned software and custom software. *See* ALA. ADMIN. CODE r. 810-6-1.37(3) (2008

canned software is treated as tangible property subject to sales tax.[212] Customized software, however, is treated as a service (or intangible information) that is exempt from sales tax.[213] Some states have rejected the distinction and tax both canned and custom software.[214]

The tax distinction between canned and custom software, which many states have adopted, raises interesting questions. For example, it is not uncommon for canned software to be modified in some respects to suit a business's particular needs, raising the question whether and to what extent modifications to canned software are

---

Supp.) ("'[C]anned computer software' [includes] software programs prepared, held, or existing for general or repeated use, including software programs developed in-house and subsequently held or offered for sale or lease. Canned computer software includes all software, except custom software programming, regardless of its function and regardless of whether it is transferred to the purchaser in physical form, via telephone lines, or by another alternative form of transmission."); *Id.* r. 810-6-1.37(5) ("'[C]ustom software programming' [includes] software programs created specifically for one user and prepared to the special order of that user. The term 'custom software programming' also includes programs that contain pre-existing routines, utilities, or other program components that are integrated in a unique way to the specifications of a specific purchaser."). See also N.Y. TAX LAW § 1115(a)(28) (McKinney 2008), for a definition of custom software that is exempt from sales tax as "[c]omputer software designed and developed by the author or creator to the specifications of a specific purchaser . . . but in no case including computer software which is pre-written." Some states rely on regulations for determining whether a program is prewritten or custom. *See, e.g.*, WIS. ADMIN. CODE TAX § 11.71(1)(k) (2011) (defining "prewritten computer software").

212. All states that have a sales tax now treat canned software as tangible property subject to sales tax. *See* HELLERSTEIN & HELLERSTEIN, *supra* note 199, ¶ 12.02, tbls.12.4 & 12.5. The chief difference among states is whether custom software is taxable or not, and whether the method by which software is delivered is relevant.

213. *See id.*

214. *See id.* (listing Arkansas, Hawaii, Mississippi, Nebraska, New Mexico, South Carolina, South Dakota, Tennessee, and Texas as states that do not exempt custom software from taxability). For examples of specific state legislation, see TENN. CODE ANN. § 67-6-231(a) (West 2009) ("The retail sale, lease, licensing or use of computer software in this state, including prewritten and custom computer software, shall be subject to the tax levied by this chapter . . . ."); TEX. TAX CODE ANN. § 151.0031 (West 2008) ("[Software that is subject to sales tax includes] a series of instructions that are coded for acceptance or use by a computer system and that are designed to permit the computer system to process data and provide results and information.").

subject to sales tax. In addition, it is not uncommon for custom software programs to be resold to different users, raising the question of whether resales of custom programs are subject to sales tax. Some states have attempted to draw lines in answering these questions by manually classifying software that is, in reality, neither canned nor custom.[215]

Although states impose their sales tax on sales of all tangible personal property, they typically impose their sales tax on only specified services.[216] A number of states have begun to include computer, data processing, and information services in the "taxable services" category.[217]

2. *Electronically Delivered Software.* Technological innovation has transformed how goods and services are purchased. Canned software, as well as music, books, newspapers, magazines, and videos, are now routinely downloaded from the Internet whereas they once were purchased only at brick and mortar establishments. States vary as to whether the method by which canned software is delivered impacts the taxability of canned software. Some states tax canned software regardless of the form in which it is delivered (i.e., it does not matter if the canned software was purchased at a retail store or downloaded from the

---

215. In Florida, for example, if a vendor modifies prepackaged software, it is considered custom software and is exempt from tax. FLA. ADMIN. CODE ANN. r. 12A-1.032 (1998) ("[W]here the vendor, at the customer's request, modifies or alters a pre-packaged program to the customer's specification and charges the customer for a single transaction, the charge is for a customized software package and is exempt as a service transaction."). In Alabama, custom software exempt from sales tax is defined to include "those services represented by separately stated charges for modifications to a canned computer software program when such modifications are prepared to the special order of the customer," but modifications to canned software "to meet the customer's needs" is considered custom software "only to the extent of the modification." ALA. ADMIN. CODE r. 810-6-1.37(5).

216. *See* HELLERSTEIN & HELLERSTEIN, *supra* note 199, ¶ 12.05.

217. *See id.* ¶ 15.11 (discussing Texas's taxes on "information services" and "data processing services" (citing TEX. TAX CODE ANN. § 1.151.0101(a)(10), (12) (West 2008))). As an additional example, see also N.Y. TAX LAW § 1105(c)(9)(i) (McKinney 2008), which imposes a tax on the "furnishing . . . of an . . . information service . . . delivered by means of telephony or telegraphy or telephone or telegraph service" and which would "otherwise be subject to taxation . . . if it were furnished by printed, mimeographed or multigraphed matter."

Internet).[218] Some states, however, exempt canned software if it is delivered electronically (i.e., canned software is taxable if bought off the shelf at a retail store but nontaxable if downloaded from the Internet).[219]

The use of the Internet to sell software raises other legal issues.[220] Under current law, an Internet software

---

218. *See, e.g.*, TENN. CODE ANN. § 67-6-231(a) ("[The sales tax applies] regardless of whether the software is delivered electronically, delivered by use of tangible storage media, loaded or programmed into a computer, created on the premises of the consumer or otherwise provided."); ILL. ADMIN. CODE tit. 86, § 130.1935(a) (2000) ("Canned software is considered to be tangible personal property regardless of the form in which it is transferred or transmitted, including tape, disc, card, electronic means or other media."); La. Dep't of Revenue, Rev. Rul. 10-001 (Mar. 23, 2010) ("Tangible personal property includes, but is not limited to, all electronically delivered products, including computer software and applications, stored media, and entertainment media or products, to equipment located in Louisiana. Taxable transactions include, but are not limited to, remotely accessed software, information materials, and entertainment media or products, whether as a one-time use or through ongoing subscription, and whether capable of only being viewed, or being downloaded . . . ."); *see also* ME. REV. STAT. ANN. tit. 36, § 1752(17) (2010) ("[Taxable] 'tangible personal property' includes any computer software that is not a custom computer software program."); N.Y. TAX LAW § 1101(b)(6) (McKinney 2008) (providing that taxable tangible personal property includes "pre-written computer software . . . regardless of the medium by means of which such software is conveyed to a purchaser"); Graham Packaging Co. v. Commonwealth, 882 A.2d 1076, 1087 (Pa. Commw. Ct. 2005) (holding canned software is taxable regardless of method of delivery); ALA. ADMIN. CODE r. 810-6-1.37(3)-(4) ("Canned computer software includes all software, except custom software programming, regardless of its function and regardless of whether it is transferred to the purchaser in physical form, via telephone lines, or by another alternative form of transmission.").

219. *See, e.g.*, VA. CODE ANN. § 58.1-609.5 (2004) (exempting computer software "not involving an exchange of tangible personal property"); Fla. Technical Assistance Advisement 07A-022 (July 19, 2007) (exempting electronically delivered software). In some states, canned software that is downloaded is taxable only if it is received in a tangible medium (i.e., online software purchases are nontaxable unless the purchaser receives a backup copy or manual in addition to the downloaded software). OKLA. ADMIN. CODE § 710:65-19-156(b)(5) (2010 Supp.) ("The levy of sales tax does not apply to . . . sales of prewritten computer software that is delivered electronically[,] . . . mean[ing software which is] delivered to the purchaser by means other than tangible storage media.").

220. One issue not addressed here is the proper sourcing of tax among multiple taxing jurisdictions. States vary as to their sourcing rules, with most states sourcing tax by destination of the tangible personal property, and with a

vendor is required to collect sales tax on taxable sales only if it has a substantial nexus (usually defined as physical presence) in the state where the buyer resides.[221] What establishes a sales tax nexus in a state has been the subject of considerable controversy. The Supreme Court has concluded that a sales tax nexus exists if an out-of-state company has offices or agents to market, solicit, bill, or conduct business activities directly related to sales of goods or services offered online in a state.[222] A number of lower courts have concluded that a sales tax nexus can also arise through the activities of affiliate stores or third parties in a state.[223] It is also possible that a sales tax nexus will be found if an online company stores its data on a server in a state[224] or if images appear on a computer screen in a state.[225] In the current era of budget deficits, it is likely that states will continue to push the limits of when an online

---

minority of states sourcing tax by origin of the tangible personal property. *See* HELLERSTEIN & HELLERSTEIN, *supra* note 199, ¶ 19A.06.

221. *See* Quill Corp. v. North Dakota, 504 U.S. 298, 311-12, 315, 317-18 (1992). An out-of-state software vendor without substantial nexus in the customer's state is not required to collect the sales tax. In such case, the customer is typically required to remit a "use" tax to his or her state of residence. *See id.*

222. *See* D.H. Holmes Co. v. McNamara, 486 U.S. 24, 32-34 (1988); Tyler Pipe Indus., Inc. v. Wash. State Dep't of Revenue, 483 U.S. 232, 249-51 (1987); Nat'l Geographic Soc'y v. Cal. Bd. of Equalization, 430 U.S. 551, 562 (1977); Scripto, Inc. v. Carson, 362 U.S. 207, 211 (1960); Gen. Trading Co. v. State Tax Comm'n, 322 U.S. 335, 337-38 (1944); Felt & Tarrant Mfg. Co. v. Gallagher, 306 U.S. 62, 67-68 (1939).

223. Some courts have found substantial nexus through affiliate presence in the state. *See, e.g.*, Borders Online, LLC v. State Bd. of Equalization, 29 Cal. Rptr. 3d 176, 178, 190-92 (Ct. App. 2005). Other courts, however, have held no substantial nexus when there exists an offline affiliate. *See, e.g.*, St. Tammany Parish Tax Collector v. Barnesandnoble.com, 481 F. Supp. 2d 575, 580-81 (E.D. La. 2007).

224. Congress carved out a substantial nexus provision in the Internet Tax Freedom Act wherein a state may require an Internet company to collect a sales tax if the company stores its website on an in-state server or owns an in-state server. *See* Pub. L. No. 105-277, tit. XI, 112 Stat. 2681-719, 2681-724 to 2681-725 (1998); *see also* 34 TEX. ADMIN. CODE § 3.286(2)(E) (2011) ("A person is engaged in business in Texas if the person has nexus with the state as evidenced by [the ownership or use of] tangible personal property that is located in this state, including a computer server or software . . . .").

225. *See* Ind. Dep't of State Revenue, Letter of Findings 09-0411 (Jan. 27, 2010).

company without an actual physical presence in the state is required to collect and remit sales taxes.[226]

3. *Cloud Computing—Software as a Service.* As discussed earlier, "cloud computing" (also known as "software as a service" or "SaaS") allows customers to pay for the use of web-based software instead of licensing or purchasing the software.[227] Because cloud computing is a relatively recent phenomenon, states are only now beginning to address the taxability of cloud computing, often in the form of administrative pronouncements and rulings.[228] In characterizing transactions occurring "in the cloud," states are addressing difficult questions such as: Does cloud computing involve the sale or license of tangible personal property (or pre-written software treated as tangible personal property)? Is cloud computing really a service, and, if so, is it one of the enumerated taxable services?[229] As described below, the current trend appears to be to attempt to fit cloud computing into existing sales tax frameworks.

Some states treat SaaS applications as "tangible personal property" (or pre-written software, which is treated as tangible personal property) subject to tax.[230] The

---

226. At least ten states have passed so-called "Amazon laws" designed to make online retailers collect sales taxes (e.g., where online retailers have online affiliates or advertising partners in the state, or where an online retailer owns stock in a subsidiary corporation that supports the online retailers operations). Amazon is currently challenging the legislation. *See* Stu Woo, *Amazon Battles States Over Sales Tax*, WALL ST. J., Aug. 3, 2011, at A1.

227. *See supra* notes 27-30 and accompanying text.

228. For commentary, see Michelle Andre, *What's New in Tax: Sales and Use Taxation in the Clouds*, KPMG 5 (July 12, 2010), http://www.us.kpmg.com/microsite/taxnewsflash/2010/Jul/sales_and_use.pdf; Brian Balingit, *Taxing Software as a Service (SaaS): Lessons Learned*, GRANTTHORNTON (Sept. 2010), http://www.gt.com/staticfiles/GTCom/Technology/Techdashboard/SaaS_%20article.pdf [hereinafter Balingit, *Lessons Learned*]; Brian Balingit, *Taxing Software as a Service: What SaaS Providers May Not Know About Their Tax Liabilities*, GRANTTHORNTON (July 2009), http://www.grantthornton.com/staticfiles/GTCom/Technology/TBD_SaaS_July_2009.pdf [hereinafter Balingit, *SaaS Providers*].

229. Other difficult tax issues relate to substantial nexus and sourcing concerns that were discussed above in connection with Internet transactions. *See supra* notes 221-26 and accompanying text.

230. *See, e.g.*, Ariz. Dep't of Revenue, Priv. Taxpayer Rul. LR05-008 (Sept. 8, 2005); Ariz. Dep't of Revenue, Priv. Taxpayer Rul. LR04-010 (Nov. 15, 2004).

rationale is that the amounts paid for the license to use a provider's software programs are receipts from the sale of pre-written software subject to tax; the sale of the use constitutes a transfer of possession of the software (i.e., the user is deemed to have constructive possession of the software because of the right to use or control the software).[231]

Other states treat SaaS as one of the enumerated taxable services, such as taxable data processing services, information services, or digital automated services.[232] Data processing services, for example, are generally defined as services involving the manipulation or storage of a customer's data, the processing or compiling of records, data retrieval, data search, and other computerized data and information storage or manipulation.[233]

New York construes an SaaS transaction as a taxable sale of software. *See* Taxpayer Guidance Div., N.Y.S. Dep't of Taxation & Fin., TSB-M-10(7)S (July 19, 2010) (ruling that the provision of online services is taxable as a sale of canned software); Taxpayer Guidance Div., N.Y.S Dep't of Taxation & Finance, TSB-A-09(44)(S) (Sept. 24, 2009); Taxpayer Guidance Div., N.Y.S Dep't of Taxation & Fin., TSB-A-09(37)S (Aug. 25, 2009) (ruling that charges for a license to use a provider's software were receipts from the sale of prewritten software); Taxpayer Guidance Div., N.Y.S Dep't of Taxation & Finance, TSB-A-08(40)S (Aug. 28, 2008).

231. *See* Taxpayer Guidance Div., N.Y.S Dep't of Taxation & Fin., TSB-A-09(37)S (Aug. 25, 2009).

232. *See, e.g.*, S.C. Dep't of Revenue, Rev. Rul. 05-13 (Oct. 14, 2005) ("[C]harges by the Application Service Provider are similar to charges by database access services and are therefore subject to the sales and use tax . . . ."); 34 TEX. ADMIN. CODE § 3.330(a)(1) (2011); Tax Policy Div., Tex. Comptroller of Pub. Accounts, Ltr. Rul. 20080509L (May 28, 2008) (ruling that an out of state "software as a service" provider (also known as an application service provider) is providing taxable data processing services); Tax Policy Div., Tex. Comptroller of Pub. Accounts, Ltr. Rul. 200401223L (Jan. 14, 2004) ("[ASP application] is subject to Texas sales tax and has data processing and information service components."); Tax Policy Div., Tex. Comptroller of Pub. Accounts, Ltr. Rul. 200009754L (Sept. 28, 2000) (ruling an application service provider, or ASP, is subject to sales tax); Tax Policy Div., Tex. Comptroller of Pub. Accounts, Ltr. Rul. 200002080L (Feb. 29, 2000) (ruling an application service provider is providing taxable data processing services).

233. *See, e.g.*, TEX. TAX CODE ANN. § 151.0035 (West 2008) (defining "data processing service").

It should be noted that the taxation of SaaS applications is not a given in all states.[234] The vast majority of states have yet to address the character of cloud-based transactions under existing laws.[235]

IV. DEBUGGING TAX CODE AND ENCOURAGING INNOVATIONS

As the above analysis of federal and state tax approaches to computer software shows, federal and state governments have not always been consistent in their tax treatments of computer software. Such inconsistency creates two major problems. First, the inconsistent tax treatment of software creates tax distinctions that run counter to the tax principles of fairness and efficiency. Second, the inconsistent tax treatment of software development often frustrates the congressional intent to encourage desirable software activities that underlies many of the provisions related to software taxation.

A. *Analysis of Tax Distinctions for Software Under the Tax Principles of Fairness and Efficiency*

Both the federal and state governments have adopted a number of seemingly incongruous tax distinctions for computer software. For federal income tax purposes, different tax rules apply to internal use software and non-internal use software,[236] off-the-shelf software and custom software,[237] software acquired with a business and software acquired separately,[238] and software protected as a patent and software protected as a copyright.[239] For state sales tax purposes, states have adopted different tax classifications for custom software,[240] electronically delivered canned

234. *See, e.g.*, Kan. Office of Policy & Research, Priv. Ltr. Rul. P-2009-005 (June 26, 2009).

235. *See* Ballingit, *Lessons Learned*, *supra* note 228; Ballingit, *SaaS Providers*, *supra* note 228.

236. *See supra* notes 146-48 and accompanying text.

237. *See supra* notes 172-82 and accompanying text.

238. *See supra* note 173 and accompanying text.

239. *See supra* notes 195-96 and accompanying text.

240. *See supra* notes 211-13 and accompanying text.

software,[241] and SaaS applications.[242] The tax distinctions for software, at both the federal and state levels, run counter to sound principles of tax policy.[243]

1. *Tax Fairness.* Perhaps the most widely accepted principle of taxation is that "persons who are similarly situated should be taxed in a similar fashion."[244] Under this fairness principle, two software owners who are similarly situated should be taxed in a similar fashion. And if they are not, the different tax treatments might signal that there is a flaw in the tax systems, suggesting a need for legislative or administrative changes. Examples of these violations of horizontal equity exist in each of the different categories of tax treatment discussed in Part III, and we consider the fairness of each, in turn.

Consider the federal tax treatment of expenses incurred in the development of computer software. In 1986, Congress modified the definition of "qualified research" eligible for the 20% research and development tax credit, to exclude internal-use software.[245] Thus, two taxpayers are treated differently for tax credit purposes if one develops software that is for internal use by the taxpayer and the other develops software that is not for internal use. Much of the commentary and scholarly debate thus far has focused on producing a workable definition of "internal-use" software that "[c]an be readily applied by taxpayers and readily administered by the IRS; and [i]s flexible enough to provide

241. *See supra* notes 218-19 and accompanying text.

242. *See supra* notes 230-35 and accompanying text.

243. While questionable tax distinctions exist at both the federal and state levels, tax distinctions also exist between the two levels. As described earlier, for federal income tax purposes, non-exclusive licenses of software are treated differently from sales. For state sales tax purposes, however, non-exclusive licenses of software are often treated as sales. In addition, at the federal level, custom software is treated as intangible property, whereas, at the state level, custom software is often viewed as a service. *See supra* Part III.

244. Jeffrey A. Maine & Xuan-Thao Nguyen, *The Unequal Tax Treatment of Intellectual Property*, 130 TAX NOTES 931, 931-32 (2011); *see also* Xuan-Thao Nguyen & Jeffrey A. Maine, *Equity and Efficiency in Intellectual Property Taxation*, 76 BROOK. L. REV. 1, 3-4 (2010).

245. I.R.C. § 41(d)(4)(E) (2006); Tax Reform Act of 1986, Pub. L. No. 99-154, § 231(b), 100 Stat. 2085, 2173-75.

continuing application into the future."[246] The federal government has not been forthcoming with a workable definition, and several commentators have expressed skepticism over whether the federal government might seriously attempt to promulgate final internal-use software rules.[247] The difficulty of establishing a workable definition, and the uncertainties and considerable controversies caused by the tax distinction, raises an interesting point. Perhaps current debate should not center on *where* the line should be drawn, but rather the debate should focus on *whether*, as a normative matter, a line should be drawn in the first place. Since the exception for internal-use software was added twenty-five years ago, we have witnessed:

> [T]echnological advancements and changes to the role of computer software in business activities . . . including the increased development of computer software by taxpayers, the increased use of computer software in all aspects of business activity, and the role of computer software (often integrated across a business) in providing goods and services in addition to the internal operations of a business.[248]

These changes suggest that a tax distinction for internal-use software might not be warranted or, at least, that a narrow definition of internal-use software should apply.

Likewise, inequities in the federal tax treatment of software acquisition costs also exist, and they, too, lack theoretical justification. For example, computer software acquired as part of a business acquisition is subject to ratable fifteen-year depreciation, as are trademarks, trade names, and goodwill.[249] Software acquired separately, however, benefits from a more rapid depreciation period of

---

246. I.R.S. Announcement 2004-9, 2004-1 C.B. 441, 445-46 (Jan. 2, 2004) (requesting comments from the public specifically concerning a definition of internal-use software).

247. *See, e.g.*, Christopher J. Ohmes et al., *Final Research Credit Regulations Expected to Immediately Affect IRS Examinations*, 102 TAX NOTES 1015, 1015-16 (2004).

248. I.R.S. Announcement 2004-9, 2004-1 C.B. 441, 444 (Jan. 2, 2004) (summarizing arguments by commentators who support a narrow definition of internal-use software).

249. *See supra* notes 167-74.

three years.[250] This disparate tax treatment raises interesting questions: Is it logical that all computer software—regardless of its underlying intellectual property protection and regardless of its actual useful life—is depreciated over either fifteen years or three years depending on its method of procurement? If computer software derives its value from its relationship to a product, service, or goodwill of a business, as do trademarks and trade names, it might be justifiable to use the same fifteen-year recovery period applicable to trademarks and trade names to avoid messy valuation and purchase price allocation problems. However, the value of software acquired as part of the purchase of a trade or business is not necessarily tied to the goodwill of the acquired trade or business. Rather, software can be freely sold, assigned, or transferred without any associated goodwill or other business assets. A strong argument can be made that if computer software is capable of reasonable valuation, it should be subject to one depreciation schedule, regardless of how it is acquired.

The federal tax treatment of the sale of computer software also raises equity concerns, as illustrated by the following two examples. In the first example, Individual A and Individual B each develop a computer software program, with Individual A obtaining a patent on his software, and Individual B relying on copyright protection for his software. Individuals A and B then sell their software for the same price, both realizing substantial gain. Although one might expect the federal tax system to treat Individuals A and B similarly, that is not the case. Individual A's gain will be treated as capital gain (under the Code's safe harbor provision governing patents), while Individual B's gain will be treated as ordinary income (under the Code's general characterization provisions).[251] In the second example, Individual C sells for $10,000 copyrighted software that she created. The company ABC, Inc., sells for $10,000 similar copyrighted software that was created by its employees. Although one might anticipate the federal tax system to treat Individual C and ABC, Inc., the same, this is not the case. Individual C will have ordinary gain on the sale of her software, but ABC, Inc. will have

250. *See supra* notes 167-74.

251. *See supra* Part III.A.3.

capital gains treatment on the sale of the software created by its employees.[252]

The federal tax distinctions highlighted in these two examples involving software sales are theoretically and analytically unjustifiable. The tax treatment of software sales should not vary based on the type of intellectual property protecting the software (patent versus copyright), and should not vary based on the status of the software owner (individual versus corporation). Neither of these distinctions is relevant in determining the tax treatment of software development and acquisition costs; it follows that neither should be relevant in determining the tax results of software sales. In 2006, Congress adopted a special rule allowing capital gains treatment for sales of musical compositions and the copyrights on them, regardless of whom the creator was and regardless of whether the musical compositions were inventory.[253] A similar bright-line rule for "software" might be justified.

At the state level, there are perhaps even more inequities in the tax treatment of software than there are at the federal level. In many states, purchasers of canned software and purchasers of custom software are treated differently for sales tax purposes based on incongruous classifications of both types of software (canned software as taxable tangible personal property and custom software as non-taxable services).[254] Commentators have challenged the tax distinction, which requires one "to determine whether the 'true object' or 'dominant purpose' of a transaction was the purchase of tangible personal property or services, when both the property and services constitute inseparable elements of a single transaction."[255] They argue that there is "no sound principle of tax policy" to support the distinction, and, that "consequently, no sound analytical basis exists for drawing a line that should not, as a normative matter, be drawn in the first place."[256] If the state sales tax is truly a "consumption" tax, then the focus should be not on whether software is tangible or services, but rather on whether the

252. *See supra* Part III.A.3.

253. I.R.C. § 1221(b)(3) (2006).

254. *See supra* Part III.B.

255. HELLERSTEIN & HELLERSTEIN, *supra* note 199, ¶ 12.06[2].

256. *Id*.

software (whether property or services) was purchased for consumption.[257]

An even more draconian example of inequity in sales taxation of software can be in found in those states that tax differently two purchasers of similar canned software, where one purchases the software off the shelf at a retail store while the other purchases the software over the Internet. If the sales tax is a true consumption tax, a tax distinction for software based on the method by which the software is delivered, or based on whether a floppy disk or tape was transferred, seems unjustified.[258]

It should be noted that some modern tax theorists might question the utility of the above critique; they question, in general, the utility of equity in tax policy analysis, often pointing to the difficulty of determining relevant likeness (i.e., the comparison of taxpayers and economic activities).[259] Requiring equal treatment for equals, they argue, merely raises the question of what "equals" actually are. It is conceded that it is difficult to evaluate tax systems governing computer software from an equity perspective because software involves such a broad range of economic activities that no two taxpayers will be situated exactly equally. For example, is a developer of internal-use software similar to a developer of software to be commercially leased or licensed? Should a seller of a software patent be viewed similarly situated to a seller of a software protected as a copyright? And should a purchaser of off-the-shelf software be viewed as similarly situated to a purchaser of custom software? But criticism of equity in tax policy analysis rests on an "exaggerated view of the level of

---

257. *See id.*

258. *See* John Wei-Ching Kuo, *Sales/Use Taxation of Software: An Issue of Tangibility*, 2 HIGH TECH. L.J. 125, 145 (1987).

259. *See* Louis Kaplow, *A Note on Horizontal Equity*, 1 FLA. TAX REV. 191, 192-93 (1992); Paul R. McDaniel & James R. Repetti, *Horizontal and Vertical Equity: The Musgrave/Kaplow Exchange*, 1 FLA. TAX REV. 607, 612-13 (1993); *see also* Eric M. Zolt, *The Uneasy Case for Uniform Taxation*, 16 VA. TAX REV. 39, 95 (1996) ("Defining horizontal equity as requiring equal tax treatment for individuals who are, in all relevant aspects, equal accomplishes little. It just begs the question of what is relevant. . . . The principle of horizontal equity does nothing to determine which differences justify different tax treatment."). *But see* Richard A. Musgrave, *Horizontal Equity: A Further Note*, 1 FLA. TAX REV. 354, 359 (1993).

precision required in order for equality to have meaning."[260] Equity is concerned with taxpayers who are "similarly situated," not with those who are "identically situated."[261] Even if criticism of equity is valid, equity can nevertheless serve as a useful tool to uncover potential problems in a tax system. The tax distinctions for software uncovered above signal that flaws might exist in federal and state tax approaches to software. At a minimum, the tax distinctions challenge us to justify disparate tax treatments.

2. *Tax Efficiency*. Another important criterion of sound tax policy is "efficiency."[262] Efficiency means "various things in various contexts," and it can be measured by different—sometimes contradictory—standards.[263]

"Administrative efficiency" can be measured by the costs of administering and complying with software tax rules.[264] Many of the tax distinctions for computer software uncovered in Part IV of this Article have been the source of significant controversy,[265] which has only increased the costs

---

260. John A. Miller, *Equal Taxation: A Commentary*, 29 HOFSTRA L. REV. 529, 545 (2000) ("All our major tax schemes have found ways to determine likeness (or difference) that are generally recognized as fair.").

261. David Elkins, *Horizontal Equity as a Principle of Tax Theory*, 24 YALE L. & POL'Y REV. 43, 44 (2006) (internal quotation marks omitted).

262. *See* Nguyen & Maine, *supra* note 244, at 5-6.

263. *See* MILLER & MAINE, *supra* note 128, at 4; *see also* MICHAEL J. GRAETZ & DEBORAH H. SCHENK, FEDERAL INCOME TAXATION: PRINCIPLES AND POLICIES 29-30 (6th ed. 2009) (summarizing various meanings of the efficiency criterion).

264. *See* GRAETZ & SCHENK, *supra* note 263, at 30 ("Complex tax rules are inefficient because taxpayers must divert time from other activities in order to calculate their taxes . . . and because the government must maintain a large agency to interpret these complex rules and to ensure that taxes are calculated correctly.").

265. As described earlier in this Article, for example, the costs of developing internal-use software are ineligible for the federal research and development tax credit. This particular rule has produced considerable uncertainty and controversy as the government itself has failed to promulgate final regulations with respect to internal-use software. In explaining the reason for not including internal-use software rules in the final regulations, the Treasury Assistant Secretary for Tax Policy stated: "The world has changed significantly since the statutory provision for internal use software was enacted in 1986. . . . A number of important issues still must be resolved before we can issue final regulations that both carry out the statute's purpose and provide clear and meaningful rules for software development today." Press Release, U.S. Dep't of the Treasury, Treasury Issues Final Research Credit Regulations (Dec. 22, 2003) (internal

of compliance and administration. Eliminating the irrational tax distinctions for software would provide certainty and clarity, minimizing these costs. The simplified tax system that would result might also be better equipped to handle emerging software products and new modes of delivery in the current era of technological advances.

Efficiency can also be measured in terms of the degree of neutrality a tax provision has on the decision-making of taxpayers. Thus, an efficient tax system governing software would not distort or interfere with software owners' economic behavior, and it would avoid the deadweight losses that accompany tax provisions that discourage tax-conscious taxpayers from engaging in software transactions that best suit their non-tax needs.[266] At present, the federal and state tax systems governing computer software are not efficient because they adopt various tax distinctions that favor one approach to structuring a transaction over another, preventing taxpayers from making tax-neutral decisions.

At the federal level, software owners are often forced to adopt inefficient strategies to minimize federal income taxes. For example, software developers are swayed by current tax rules to choose patent or trade secret protection for their software to ensure preferential capital gains treatment on a later assignment.[267] Developers of internal-use software are swayed to engage in the façade of leasing or licensing their software to customers to ensure the benefit of the research and development tax credit.[268] Purchasers of off-the-shelf software are swayed to refrain from modifying their off-the-shelf software, lest they risk forfeiting their expense deduction.[269] Purchasers of custom

---

quotation marks omitted), *available at* http://www.treasury.gov/press-center/press-releases/Pages/js1064.aspx.

266. *See* GRAETZ & SCHENK, *supra* note 263, at 29 ("[E]fficiency . . . requires that a tax interfere as little as possible with people's economic behavior."); Elkins, *supra* note 261, at 47 (stating that efficient taxes minimize deadweight losses caused by taxpayer actions to reduce tax burden by choosing courses of action that minimize tax); Zolt, *supra* note 259, at 63 (stating "[e]fficient taxes distort as little as possible," and describing three forms in which distortion come).

267. *See supra* Part III.A.3.

268. *See supra* Part III.A.1.

269. *See supra* notes 179-82 and accompanying text.

software are swayed to purchase software separately from the purchase of assets constituting a trade or business, in order to obtain a shorter cost-recovery period.[270] These decisions should be tax-neutral, but under the present tax system, they are not.

Similar problems plague state taxation systems. For example, taxpayers are often swayed to purchase custom software (exempt from sales taxation) over prewritten software (subject to sales taxation), or at least to have prewritten software modified enough to fall within the classification of non-taxable custom software.[271] And because the mode of transmission of canned software can affect tax results, taxpayers are swayed to choose electronic delivery instead of purchasing disks or tapes.[272] These tax distinctions are not neutral, because "[b]usinesses are not free to make efficient, economic choices among what should be equal and value-free modes of transfer."[273]

As demonstrated above, the federal and state tax distinctions for software violate sound tax principles of fairness and efficiency. In addition, such tax distinctions also have the potential to "become a stumbling block to an efficient market for software," by "creat[ing] barriers and disincentives which may eventually hinder the continued growth and development of [the software] industry."[274] It is thus worthy to evaluate the tax subsidies for software development and their general effectiveness in promoting economic growth.[275]

---

270. *See supra* notes 175-78 and accompanying text.

271. *See supra* notes 211-15 and accompanying text.

272. *See supra* Part III.B.2.

273. Kuo, *supra* note 258, at 145 ("The California legislature has not yet expressly announced any policy reasons for favoring electronic transmissions while penalizing floppy disks or tapes, yet business are nevertheless swayed by the present tax structure to adopt the former mode of transmission over the latter. Businesses are not free to make efficient, economic choices among what should be equal and value-free modes of transfer. Instead, the tax system introduces a hidden, uncontemplated prejudice favoring one alternative which is not necessarily the most efficient. Thus, the tax structure unnecessarily intrudes into business decisions and forces businesses to adopt inefficient strategies.").

274. *Id*. at 125-26.

275. In this Article, efficiency has been measured in terms of administrative costs and neutrality. Efficiency can also be measured in terms of economic

B. *Evaluation of Tax Preferences for the Software Development Industry*

As discussed earlier, federal tax preferences for software development exist principally in the form of a 100% deduction for software development spending and a 20% tax credit for an increase in certain software development spending.[276] Some commentators have argued that these tax preferences for software development "provide too much subsidy" and that they "favor investments that have no special merit" and "encourage a waste of capital."[277] But these arguments ignore the spillover effects or external benefits of the software development industry. While economists disagree on almost everything, most economists do believe that technological innovation accounts for a major share of long-term economic growth in the United States.[278] And most economists also agree that the "average social returns to private [research and development] investments greatly exceed the average private returns."[279] Thus, the fact that tax preferences might encourage software development that otherwise would not

---

growth. In other words, tax systems governing software would be viewed as efficient if they "promoted economic growth and inefficient if [they] inhibit[ed] such growth." GRAETZ & SCHENK, *supra* note 263, at 29; *see also* Edward Yorio, *The President's Tax Proposals: A Major Step in the Right Direction*, 53 FORDHAM L. REV. 1255, 1262-63 (1985) (examining economic growth as a principal criterion of sound federal income tax policy).

276. *See supra* Part III.A.1.

277. Calvin H. Johnson, *Capitalize Costs of Software Development*, 124 TAX NOTES 603, 603, 606 (2009) (providing proposals for software development as part of the Shelf Project, a collaboration by tax professionals to develop revenue raising proposals for Congress). In 2008, more than 12,000 corporations claimed more than $8.3 billion in research tax credits. *Table 1: Corporations Claiming a Credit for Increasing Research Activities on Form 6765*, INTERNAL REVENUE SERVICE (2008), http://www.irs.gov/taxstats/article/0,,id=164402,00.html.

278. GARY GUENTHER, CONG. RESEARCH SERV., RESEARCH AND EXPERIMENTATION TAX CREDIT: CURRENT STATUS AND SELECTED ISSUES FOR CONGRESS 1 (2009) (citing Linda R. Cohen & Roger G. Noll, *Privatizing Public Research*, SCI. AM., Sept. 1994, at 72).

279. *Id*. (citing Edwin Mansfield, *Microeconomics of Technological Innovation*, *in* THE POSITIVE SUM STRATEGY 307, 307-25 (Ralph Landau & Nathan Rosenberg eds., 1986)); *see also* Charles I. Jones & John C. Williams, *Measuring the Social Return to R&D*, 113 Q. J. ECON. 1119, 1132-35 (1998).

be made can be viewed as an acceptable violation of the tax neutrality ideal.

Although most commentators do not argue for eliminating tax preferences for software development,[280] many policy analysts criticize flaws in the design of existing tax benefits for software development. For example, many take issue with the 100% deduction for software development, which applies only if the research costs are incurred "in connection with" the developer's trade or business.[281] In a recent case, the Ninth Circuit affirmed a case in which a court denied deductions to a computer software developer because he licensed the technology to another company for use in that company's trade or business, rather than marketing the developed technology himself.[282] Such an approach fails to recognize the importance of software licensing in today's economy and favors only software development activities of a sufficiently sustained character.[283]

Similarly, the 20% tax credit for software development is also designed in ways that limit its effectiveness.[284] For example, the incremental nature of the credit means that many firms cannot utilize it all; this could be the case if a company's gross sales grew faster than its qualified research spending.[285] In addition, the nonpermanent nature

---

280. *But see* Johnson, *supra* note 277, at 603 (proposing the capitalization of software development costs and replacing the research and development tax credit with a competitive award administered by the National Science Foundation for ground-breaking work). In fact, most commentators endorse tax preferences to spur desirable research spending. GUENTHER, *supra* note 278, at 17 ("Most policy analysts and lawmakers endorse the use of tax incentives to spur increased domestic business R&D investment").

281. I.R.C. § 174(a)(1) (2006).

282. Saykally v. Comm'r, 85 T.C.M. (CCH) 1401, 1410-11 (2003), *aff'd*, 247 F. App'x 914 (9th Cir. 2007).

283. *See* Maine & Nguyen, *supra* note 244, at 932-33 (Example 1).

284. For considerations that limit the credit's effectiveness, see OFFICE OF TAX POLICY, INVESTING IN U.S. COMPETITIVENESS: THE BENEFITS OF ENHANCING THE RESEARCH AND EXPERIMENTATION (R&E) TAX CREDIT 4-5 (2011); Michael D. Rashkin, *The Dysfunctional Research Credit Hampers Innovation*, 131 TAX NOTES 1057, 1062-64 (2011).

285. The basic research credit is 20% of qualified research spending above a base amount. I.R.C. § 41(a). The base amount, which can be thought of as a firm's normal level of research investment, is a fixed-base percentage (research

of the credit makes it difficult for firms to plan ahead for research activities. The credit is only temporary, and although Congress has extended the credit numerous times, efforts to make it permanent have failed due to revenue concerns.[286]

There are two significant points to note about both the 100% deduction and the 20% credit. First, they are aimed at the computer software development market only; they do not apply to purchases of software by third parties for further experimentation and development. As it stands, software purchase costs must be capitalized and then depreciated over either fifteen years (if acquired with a trade or business) or three years if acquired separately.[287] There has been little debate over whether software purchase costs might justify a departure from normative capitalization—although the government does permit the immediate expensing of costs of purchasing tangible personal property (such as new equipment and machinery).[288] Likewise, little attention has been paid to what might be the ideal cost-recovery period for capitalized software purchase costs.

A fifteen-year depreciation period for purchased software is inappropriate considering the relatively risky nature of software compared to other intangibles. As some economists have argued, "depreciation schedules for

---

expenditures during 1984-1988 divided by gross receipts during 1984-1988, I.R.C. § 41(c)(3)) multiplied by average annual gross revenues for the past four years. I.R.C. § 41(c)(1). A problem with the current determination of "base amount" is that even though research spending increases, the credit is not available if gross sales have grown faster than research spending. Another problem with the credit is that it is calculated based on research spending relative to receipts in the years 1984 to 1988, which does not reflect realities of today's economic and technological world, and which could penalize a firm that had high research spending levels during the 1984 to 1988 base period (unless the alternative credit provided a benefit). *See supra* Part III.A.1.

286. A one-year extension of the credit, for example, was estimated to cost the government almost $9 billion over ten years. STAFF OF JOINT COMM. ON TAXATION, 110TH CONG., ESTIMATED REVENUE EFFECTS OF H.R. 6049, at 4 (Comm. Print 2008).

287. There is a temporary Code provision that permits the immediate deduction of off-the-shelf purchase costs. *See supra* notes 179-82 and accompanying text.

288. For such a proposal, see Xuan-Thao Nguyen & Jeffrey A. Maine, *Acquiring Innovation*, 57 AM. U.L. REV. 775, 808-13 (2008).

relatively risky assets should be accelerated to compensate the owners of such assets for bearing a disproportionably large share of the capital price risk."[289] One might argue that a lengthy fifteen-year recovery period is justified due to the fact that ex post adjustments are available to software owners upon later sale or retirement of purchase software.[290] But ex ante slow depreciation (fifteen years) with substantial ex post adjustments are not necessarily favored over ex ante accelerated depreciation schedules (three years) with fewer ex post adjustments.[291]

A second point to note about the 100% deduction and the 20% credit is that neither contains any restriction on the location of the resulting intellectual property.[292] Thus, even though software developed in the United States is eligible for federal tax benefits, there is no requirement that the software stay in the United States. Unsurprisingly, in recent years, numerous technology companies (such as Google, Microsoft, and even Facebook) have reduced their U.S. taxes on worldwide sales of software by transferring the manufacture and marketing of their developed software to low-tax and low-wage countries.[293]

---

289. Ethan Yale, *When Are Capitalization Exceptions Justified?*, 57 TAX L. REV. 549, 572 (2004) (citing Jeremy I. Bulow & Lawrence H. Summers, *The Taxation of Risky Assets*, 92 J. POL. ECON. 20, 37-38 (1984)); *see* Roger H. Gordon & John Douglas Wilson, *Measuring the Efficiency Cost of Taxing Risky Capital Income*, 79 AM. ECON. REV. 427, 438 (1989).

290. Software owners may take a tax loss deduction on either the sale or retirement of software, with the deductible amount on sale being the excess of the adjusted basis in the software over the amount realized in the trade, and the deducible amount on retirement or obsolescence being the unrecovered adjusted basis in the software. *See* I.R.C. §§ 165, 1001 (2006).

291. As noted by one commentator, "an accelerated depreciation system . . . reduces strategic loss-taking. Under an accelerated schedule adjusted basis is lower at any given point in time. It is less likely that adjusted basis will ever exceed market value by enough to make strategic loss-taking profitable net of trading costs." Jeff Strnad, *Tax Depreciation and Risk*, 52 SMU L. REV. 547, 597 (1999).

292. *See supra* note 151 and accompanying text.

293. *See* Martin A. Sullivan, *Microsoft Moving Profits, Not Jobs, Out of the U.S.*, 129 TAX NOTES 271, 273-74 (2010); Jesse Drucker, *Google 2.4% Rate Shows How $60 Billion Lost to Tax Loopholes*, BLOOMBERG (Oct. 21, 2010, 6:00 AM), http://www.bloomberg.com/news/2010-10-21/google-2-4-rate-shows-how-60-billion-u-s-revenue-lost-to-tax-loopholes.html; *see also* Martin A. Sullivan, *Medtronic Moves Jobs, Profits Out of U.S.*, 128 TAX NOTES 687, 690-93 (2010).

As a general rule, the United States does not tax foreign profits of subsidiaries of American companies until the profits are repatriated (e.g., in the form of dividends).[294] Thus, as a general matter, a U.S. software company may transfer its recently developed software (or non-U.S. rights to the software), for which it has received substantial tax benefits, to a subsidiary company formed in a low-tax jurisdiction (such as Ireland or The Netherlands).[295] The foreign subsidiary company will subsequently produce software products and sell them to China, India, and the rest of the world, and the profits will not be subject to immediate U.S. taxation.

However, Congress has enacted a series of rules designed to prevent U.S. companies from avoiding or deferring tax.[296] Unfortunately for Congress, these "anti-deferral tax regimes" are easily avoidable.[297] For example, Congress enacted "subpart F" of the Code to deal with controlled foreign corporations.[298] Under subpart F, a U.S. corporation is currently subject to U.S. tax on the "subpart F income" received by its controlled foreign subsidiaries, whether or not such income is actually received by the U.S. corporation.[299] Sales of software products by controlled

294. Foreign corporations are generally taxed only on U.S. source income, not worldwide income. The test for determining whether a corporation is a domestic corporation (taxed on worldwide income) or a foreign corporation (taxed only on U.S. source income) is determined by where the corporation was created or organized. I.R.C. § 7701(a)(3)-(5).

295. *See supra* note 152 and accompanying text.

296. *See* I.R.C. §§ 367(d), 482, 951-965. Under Section 367(d), a U.S. company that transfers intangible property to its foreign subsidiary corporation is *deemed* to have sold the property in exchange for contingent payments "commensurate with the income attributable to the intangible." I.R.C. § 367(d)(2) (flush language). As with most of the anti-deferral tax provisions, however, "the impact of Code Section 367 on software sales can be largely mitigated at the present time through careful planning." Darby & Lemaster, *supra* note 152, at 12.

297. *See, e.g.*, Darby & Lemaster, *supra* note 152, at 2.

298. I.R.C. §§ 951-965.

299. Subpart F income is defined to include so-called "foreign base company income." I.R.C. § 952(a)(2). Foreign base company income includes a company's foreign personal holding company income, foreign base company sales income, foreign base company services income, and foreign base company oil related income. I.R.C. § 954. Each of these categories of foreign base company income is defined in the Code. *Id*.

foreign corporations would seem to fall within the scope of subpart F of the Code.[300] There is an important exception, however, for controlled foreign corporations that "manufacture" the products they sell.[301] In short, income earned by a controlled foreign corporation is income from the sale of property manufactured in the foreign country, and, therefore, the income is not considered subpart F income. The income will be subject to U.S. tax only if—and when—that income is repatriated, again, often in the form of dividends.[302] The manufacturing exception makes it relatively easy to avoid subpart F of the Code, and is partly to blame for the migration of software offshore.[303]

As Congress continues to address income shifting caused by international tax loopholes, the tax benefits available for U.S. software development should be evaluated. It seems only fair, for example, that the research and development credit should come with certain caveats, such as: research activities must occur in the United States, research costs must be incurred in the United States, and the resulting intellectual property must be retained in the United States. A number of foreign countries impose such restrictions on their research and development tax incentives.[304] Thus, if U.S. software companies desire to shift software manufacturing overseas, the transfer-pricing rules should be implicated to ensure that most of the subsidiary's foreign profits from manufacturing and selling software products are paid currently to the U.S. parent company in the form of royalties for the use of the software. Software is a valuable asset, and the royalty payment should reflect that.

---

300. *See* I.R.C. § 954(a)(2), (d).

301. Treas. Reg. § 1.954-3(a)(4) (2011).

302. *See* I.R.C. § 61(a)(7).

303. Under the transfer pricing rules of Section 482, the initial transfer of software from the U.S. parent company to the foreign subsidiary company must be for arm's-length consideration—that is, the foreign subsidiary must make a buy-in payment for the right to exploit the software outside the United States. *See* I.R.C. § 482 (2006). Companies have attempted minimize the impact of the transfer pricing rules by entering into cost sharing arrangements with the foreign subsidiary for the co-development of the software code.

304. *See* DELOITTE, *supra* note 151, at 4, 38.

CONCLUSION

The growth of technology companies (such as Google, Microsoft, and even Facebook) brings to forefront the intersection of intellectual property and taxation. Technological innovation, in particular software innovation, is crucial to economic growth and provides invaluable social returns. To stimulate software innovation for the public good, however, it is important that the government create strong intellectual property laws and adopt sound tax policy for software. The intersection of intellectual property and taxation presents unique challenges, as software contains tangible and intangible elements, software can encompass some combination of the traits of copyrights, trade dress, patents, and trade secrets, and software can be delivered through various media. In addressing these challenges, the federal and state governments should strive to develop tax schemes that are theoretically and analytically satisfactory.