

## University of Florida Levin College of Law UF Law Scholarship Repository

---

UF Law Faculty Publications

Faculty Scholarship

---

2009

# Measuring Compliance with Compulsory Licensing Remedies in the American Microsoft Case

William H. Page

*University of Florida Levin College of Law*, [page@law.ufl.edu](mailto:page@law.ufl.edu)

Seldon J. Childers

Follow this and additional works at: <http://scholarship.law.ufl.edu/facultypub>

 Part of the [Antitrust and Trade Regulation Commons](#), and the [Computer Law Commons](#)

---

### Recommended Citation

William H. Page & Seldon J. Childers, *Measuring Compliance with Compulsory Licensing Remedies in the American Microsoft Case*, 76 *Antitrust L.J.* 239 (2009), available at <http://scholarship.law.ufl.edu/facultypub/583>

This Article is brought to you for free and open access by the Faculty Scholarship at UF Law Scholarship Repository. It has been accepted for inclusion in UF Law Faculty Publications by an authorized administrator of UF Law Scholarship Repository. For more information, please contact [outler@law.ufl.edu](mailto:outler@law.ufl.edu).

MEASURING COMPLIANCE WITH COMPULSORY  
LICENSING REMEDIES IN THE AMERICAN  
*MICROSOFT* CASE

WILLIAM H. PAGE  
SELDON J. CHILDERS\*

I. INTRODUCTION

In January 2008, Judge Colleen Kollar-Kotelly extended the term of all but one of the provisions of the *Microsoft* final judgments<sup>1</sup> for two years on the grounds that there had been “extreme and unforeseen delay”<sup>2</sup> in the implementation of a troublesome provision of the judgments that the parties had already agreed to extend. The provision at

---

\* Professor Page is the Marshall M. Criser Eminent Scholar, University of Florida Levin College of Law. Mr. Childers is a 2008 graduate (J.D.) from the Levin College of Law at the University of Florida and was previously a software developer and management consultant (1993–2004). We would like to thank Mark Lemley, David Heiner, Danny Sokol, and the participants in the ABA Section of Antitrust Law Conference on Remedies for Dominant Firm Misconduct at the University of Virginia School of Law (June 2008).

<sup>1</sup> There are two final judgments, a consent decree negotiated by the United States (joined by some of the state plaintiffs) and a judgment entered by the court for a group of states that opposed the consent decree. *United States v. Microsoft Corp.*, No. 98-1232, 2006 WL 2882808, § IV.A (D.D.C. Sept. 7, 2006) (amending the United States and settling states’ final judgment entered Nov. 12, 2002) (*Consent Decree 2006*), available at <http://www.usdoj.gov/atr/cases/f218300/218339.pdf>; *New York v. Microsoft Corp.*, No. 98-1233, 2006 WL 3949168, § IV.B (D.D.C. Sept. 21, 2006) (amending the nonsettling states’ final judgment entered Nov. 1, 2002). The judgments are very similar. For convenience, in the remainder of the article, we will cite only to the Consent Decree and the relevant section number.

Because of the many opinions in the *Microsoft* litigation, the custom of numbering opinions by Roman numeral for citation (e.g., *Microsoft V*) is impractical. Consequently, we adopt the citation strategy of WILLIAM H. PAGE & JOHN E. LOPATKA, *THE MICROSOFT CASE: ANTITRUST, HIGH TECHNOLOGY, AND CONSUMER WELFARE* 203–42 (2007), which includes the court, date, and a citation of the volume and page number (e.g., *D.D.C. States Remedy 2002*, 224 F. Supp. 2d at 269).

<sup>2</sup> *New York v. Microsoft Corp.*, 531 F. Supp. 2d 141, 144 (D.D.C. 2008) (*D.D.C. Decree Extension 2008*). In April 2009, the parties agreed to extend the “Surviving Provisions” of the final judgments, including § III.E, at least until May 12, 2011. Joint Motion to Modify Final Judgment and Supporting Memorandum of Points and Authorities, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Apr. 16, 2009), available at <http://www.usdoj.gov/atr/cases/f244900/244921.pdf>. Judge Kollar-Kotelly agreed to the extension. Second

issue, Section III.E, requires Microsoft to “make available,” including by licensing its intellectual property (IP),<sup>3</sup> certain communications protocols that Windows client operating systems use to interoperate with Microsoft’s server operating systems.<sup>4</sup> Interestingly, the extension came at the request of only some of the state plaintiffs<sup>5</sup> and over the objection of the United States.<sup>6</sup>

In extending the judgments,<sup>7</sup> the court found that Microsoft’s failure to comply with the (already extended) protocol licensing requirement, in particular its five-year delay in supplying satisfactory technical documentation supporting the protocols, constituted “changed circumstances.”<sup>8</sup> Moreover, according to Judge Kollar-Kotelly, because of the provision’s “forward-looking” character, it had “paramount significance in the Final Judgments’ scheme,”<sup>9</sup> not only in itself but also in its relationship to other provisions. Consequently, she also extended the other provisions at least until November 2009 to give the judgments a better chance of having their planned effect.

We do not discuss whether or in what circumstances shortcomings in compliance with one provision of a consent decree should justify the extension of others. Instead, we will focus on the issue of compliance with the documentation element of the protocol licensing requirement.

---

Modified Final Judgment, *United States v. Microsoft Corp.*, No. 98-1232, § V.A (D.D.C. Apr. 22, 2009), available at <http://www.usdoj.gov/atr/cases/f245100/245110.pdf>.

<sup>3</sup> For discussion of policy issues in compulsory licensing of IP as an antitrust remedy, see Colleen Chien, *Cheap Drugs at What Price to Innovation: Does the Compulsory Licensing of Pharmaceuticals Hurt Innovation?*, 18 *BERKELEY TECH. L.J.* 853 (2003); Daniel Kanter, *IP and Compulsory Licensing on Both Sides of the Atlantic—An Appropriate Antitrust Remedy or a Cutback on Innovation*, 2006 *EUR. COMPETITION L. REV.* 27(7), 351; Lawrence Schlam, *Compulsory Royalty-Free Licensing as an Antitrust Remedy for Patent Fraud: Law, Policy and the Patent-Antitrust Interface Revisited*, 7 *CORNELL J.L. & PUB. POL’Y* 467 (1998).

<sup>4</sup> *Consent Decree 2006*, *supra* note 1, § III.E.

<sup>5</sup> The moving states included some states from the original New York Group, which joined the United States in the settlement with Microsoft, and some from the California Group, which did not settle. *Id.* at 143 n.2. The California movants were California, Connecticut, Iowa, Kansas, Minnesota, Massachusetts, and the District of Columbia. The New York Movants were New York, Maryland, Louisiana, and Florida. *Id.*

<sup>6</sup> Brief of the United States as Amicus Curiae in Opposition to the Motions to Extend the States’ Final Judgments, *New York v. Microsoft Corp.*, No. 98-1233 (D.D.C. filed Nov. 9, 2007), available at <http://www.usdoj.gov/atr/cases/f227500/227585.pdf>. The disagreement among the plaintiffs, once again, casts doubt on the merits of fragmenting authority to enforce national antitrust policy. See PAGE & LOPATKA, *supra* note 1, at 246.

<sup>7</sup> The motions and the eventual extension did not include § III.B, which required Microsoft to license Windows to computer manufacturers on uniform terms, with certain exceptions. *D.D.C. Decree Extension 2008*, 531 F. Supp. at 144 n.5.

<sup>8</sup> *Id.* at 144, 184. She did not find that Microsoft’s actions constituted “a pattern of willful and systematic violations.” *Id.* at 167 (quoting *Consent Decree 2006*, *supra* note 1, § V.B).

<sup>9</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 181.

As we explain more fully below, this provision was problematic from the outset because it did not respond directly to any proven antitrust violation by Microsoft. Monopolization remedies should usually aim to remove impediments that proven violations place in the way of entry, innovation, and expansion. The protocol licensing remedy, by contrast, imposed an affirmative obligation, essentially unrelated to any proven violations, to facilitate possible future entry by unknown firms and technologies. Many of the problems the court and the parties have encountered in enforcing the provision can be traced to this high ambition.

Even assuming that the protocol licensing requirement was warranted, we question whether Judge Kollar-Kotelly applied the proper measure of compliance. She held that Microsoft had delayed implementation of the decree by failing to produce “certifiably complete, accurate, and useable”<sup>10</sup> documentation of its protocols. In practice, this standard requires something approaching perfection: Microsoft must deliver documentation that generates no technical issues during testing by the plaintiffs’ engineers. Microsoft, for whatever reason, did not formally contest this standard. Nevertheless, we argue that this standard is unnecessarily costly and divorced from market realities. In software development, as in any other practical endeavor, “the perfect is the enemy of the good.”<sup>11</sup> To illustrate this point, we compare the court’s standard of compliance with the sort of practices a firm would adopt to bring a software development kit to market. We propose a market-based standard that focuses the question of compliance on the defendant’s responsiveness to the interoperability needs of real-world developers. Finally, we note that the approach applied by the European Commission in evaluating Microsoft’s compliance with its 2004 decision is similar to our proposed standard.

## II. THE FINAL JUDGMENTS

In June 2001, the D.C. Circuit held that Microsoft had monopoly power in the market for PC operating systems and had violated Section 2 of the Sherman Act by trying to thwart a threat to its monopoly posed by the Netscape Web browser and by Sun’s Java technologies.<sup>12</sup> The browser and Java, separately or together, were “middleware”—applications with platform capabilities—that might have evolved into general

<sup>10</sup> *Id.* at 144, 163, 170, 175 (twice), 181, 183 n.34, 184.

<sup>11</sup> Scott Kieff, *IP Transactions: On the Theory & Practice of Commercializing Innovation*, 42 Hous. L. Rev. 727, 731 (2005) (quoting THE OXFORD DICTIONARY OF QUOTATIONS 716 (Angela Partington ed., 4th ed. 1996) (attributing the saying to VOLTAIRE, DICTIONNAIRE PHILOSOPHIQUE (1770))).

<sup>12</sup> *United States v. Microsoft Corp.*, 253 F.3d 34, 49 (D.C. Cir. 2001) (*D.C. Circuit 2001*).

platforms for software applications and, thus, diminished the “applications barrier to entry” that protected Microsoft’s monopoly in PC operating systems.<sup>13</sup> Some of Microsoft’s contractual and design measures, the court held, prevented potential competitors from achieving the critical mass of users necessary for them to evolve into rival platforms.<sup>14</sup> The court of appeals also reversed some of the district judge’s liability rulings<sup>15</sup> and his remedy, which included a breakup of Microsoft.<sup>16</sup> Consequently, the court of appeals remanded the case to a new judge with instructions to formulate a remedy more closely tailored to the surviving grounds of liability.<sup>17</sup>

After remand, with the encouragement of newly assigned Judge Kollar-Kotelly, the United States and some of the states reached agreement with Microsoft on the terms of a consent decree,<sup>18</sup> while other state plaintiffs continued the litigation in hopes of gaining more extensive relief.<sup>19</sup> In two parallel proceedings, Judge Kollar-Kotelly considered whether the proposed consent decree was in the public interest<sup>20</sup> and whether the nonsettling states were entitled to more relief.<sup>21</sup> Ultimately, she approved the consent decree<sup>22</sup> and entered a separate judgment granting essentially identical relief to the nonsettling states.<sup>23</sup> The twin judgments went into effect in November 2002 and were due to expire five years later.<sup>24</sup> The D.C. Circuit affirmed the judgments in their entirety in 2004.<sup>25</sup>

In this Part, we briefly describe the terms of the judgments and the problems that Microsoft and the plaintiffs have encountered in securing compliance with their terms. In doing so, we suggest that, the judg-

---

<sup>13</sup> *Id.* at 53–54.

<sup>14</sup> *Id.* at 60–74 (describing measures affecting Netscape); *id.* at 74–78 (describing measures aimed at Java).

<sup>15</sup> *Id.* at 75, 80–81.

<sup>16</sup> *Id.* at 98.

<sup>17</sup> *Id.* at 105–07.

<sup>18</sup> Competitive Impact Statement, United States v. Microsoft Corp., No. 98-1232, 8–9 (D.D.C. Nov. 15, 2001), available at <http://www.usdoj.gov/atr/cases/f222900/222994.pdf>.

<sup>19</sup> *New York v. Microsoft Corp.*, 209 F. Supp. 2d 132 (D.D.C. 2002).

<sup>20</sup> *United States v. Microsoft Corp.*, 215 F. Supp. 2d 1 (D.D.C. 2002).

<sup>21</sup> *New York v. Microsoft Corp.*, 209 F. Supp. 2d 132 (D.D.C. 2002).

<sup>22</sup> *United States v. Microsoft Corp.*, 231 F. Supp. 2d 144, 192 (D.D.C. 2002) (*D.D.C. Tunney Act 2002*).

<sup>23</sup> *New York v. Microsoft Corp.*, 224 F. Supp. 2d 76, 266–77 (D.D.C. 2002) (*D.D.C. States Remedy 2002*), *aff’d sub nom. Massachusetts v. Microsoft Corp.*, 373 F.3d 1199 (D.C. Cir. 2004) (*D.C. Circuit 2004 Remedy*).

<sup>24</sup> See *supra* note 1.

<sup>25</sup> *D.C. Circuit 2004 Remedy*, 373 F.3d at 1224, 1250.

ments' adoption of mandatory licensing in the absence of a proven market need made the difficulties of enforcement that lay ahead predictable.

#### A. TERMS

The terms of the final judgments corresponded closely, but not perfectly, to the liability rulings the court of appeals affirmed.<sup>26</sup> The judgments, for example, seek to prevent Microsoft from contracting with computer manufacturers or designing Windows in ways that limit its rivals' ability to install or promote middleware on new Windows PCs.<sup>27</sup> Parallel provisions aim to prevent Microsoft from limiting its rivals' other access to channels of distribution by exploiting its control over software developers, hardware vendors, Internet access and content providers, and end users.<sup>28</sup> Each of these provisions is responsive, directly or indirectly, to some contractual or design action that the court of appeals held unlawful.<sup>29</sup> Some of Microsoft's conduct that the court found to be unlawful is not addressed in the final judgments at all.<sup>30</sup>

In two provisions that the court described as "forward-looking," however, the final judgments exceeded the liability rulings. One of these provisions requires Microsoft to disclose to affected firms the applications programming interfaces (APIs) that Windows uses to interoperate with Microsoft's own middleware products, like Internet Explorer and Microsoft Media player.<sup>31</sup> This provision is designed to assure that Microsoft does not use special knowledge of its APIs to gain an advantage over rivals in the markets for middleware products. The "most forward-looking provision,"<sup>32</sup> Section III.E, requires Microsoft to "make available" to interested firms:

on reasonable and non-discriminatory terms (consistent with Section III.I [which requires licensing of necessary intellectual property<sup>33</sup>]) any Communications Protocol that is . . . (i) implemented in a Windows

<sup>26</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 150–54.

<sup>27</sup> The decree defines various categories of Microsoft and non-Microsoft middleware. *Consent Decree 2006*, *supra* note 1, §§ VI.J, VI.K, VI.M, VI.N. The decree also preserves computer manufacturers' flexibility to delete Microsoft products and to install and enable access to those of rivals. *Id.* §§ III.C, III.H.

<sup>28</sup> *Id.* §§ III.F, III.G.

<sup>29</sup> For more discussion, see PAGE & LOPATKA, *supra* note 1, at 71–76.

<sup>30</sup> *Id.* at 77 (discussing actions that may have deceived developers into writing to the Windows-specific version of Java).

<sup>31</sup> *Consent Decree 2006*, *supra* note 1, § III.D.

<sup>32</sup> *D.D.C. States Remedy 2002*, 224 F. Supp. 2d at 173.

<sup>33</sup> This section requires Microsoft to license its intellectual property with reasonable and nondiscriminatory royalties and limitations.

Operating System Product installed on a client computer, and (ii) used to interoperate, or communicate, natively [i.e., without the addition of other software] with a Microsoft server operating system product.<sup>34</sup>

Implementation of this provision has posed so many difficulties that the court felt compelled to extend most of the judgments' other provisions.

As we have explained elsewhere,<sup>35</sup> Section III.E has little to do with the liability holdings in the case.<sup>36</sup> It and the API disclosure provision are based on a widely held belief that Microsoft manipulates its Windows interfaces to disadvantage its rivals in applications markets.<sup>37</sup> An early version of this charge was the popular saying "[MS-] DOS isn't done until Lotus [1-2-3, once a rival spreadsheet program] won't run."<sup>38</sup> A later version of the charge appeared in the famous Netscape White Paper that provided the impetus and conceptual foundation for the government's case.<sup>39</sup> The complaint, however, did not include the charge.<sup>40</sup> The charge only resurfaced after the trial on liability, during the negotiations over the appropriate remedy.<sup>41</sup> The government apparently believed that Microsoft had already ceased most of the conduct at issue in the case. However, the government insisted on including a provision intended to hinder Microsoft's ability to extend its dominance into network computing because the computer market was moving in the direction of network-based applications that users can access either within corporate local networks or over the Internet.<sup>42</sup>

The rationale for the protocol licensing provision was that server operating systems are platforms for middleware comparable to client operating systems.<sup>43</sup> Middleware running on non-Microsoft servers might

---

<sup>34</sup> *Id.* at 269.

<sup>35</sup> William H. Page & Seldon J. Childers, *Software Development as an Antitrust Remedy: Lessons from the Enforcement of the Microsoft Communications Protocol Licensing Requirement*, 14 MICH. TELECOMM. & TECH. L. REV. 77 (2007), available at <http://www.mttr.org/volfourteen/page.pdf>.

<sup>36</sup> *D.D.C. 2002 Tunney Act*, 231 F. Supp. 2d at 189 (observing that "this aspect of the remedy plainly exceeds the scope of liability"); *D.C. Circuit 2004 Remedy*, 373 F.3d at 1223.

<sup>37</sup> Page & Childers, *supra* note 35, at 93.

<sup>38</sup> For a Microsoft employee's investigation of this charge, see Adam Barr, *DOS Ain't Done til Lotus Won't Run?* (Aug. 1, 2005), [http://www.proudlyserving.com/archives/2005/08/dos\\_aint\\_done\\_t.html](http://www.proudlyserving.com/archives/2005/08/dos_aint_done_t.html).

<sup>39</sup> Gary Reback & Susan Creighton, *White Paper Regarding Recent Anticompetitive Conduct of Microsoft Corporation* (July 1996) (unpublished manuscript, on file with authors). For discussion, see PAGE & LOPATKA, *supra* note 1, at 28-29.

<sup>40</sup> Page & Childers, *supra* note 35, at 95-96.

<sup>41</sup> See KEN AULETTA, *WORLD WAR 3.0: MICROSOFT AND ITS ENEMIES* 340-62 (2001).

<sup>42</sup> Page & Childers, *supra* note 35, at 96-103.

<sup>43</sup> *D.D.C. Tunney Act 2002*, 231 F. Supp. 2d at 189-90.

evolve into a platform rival for Windows.<sup>44</sup> Consequently, it was thought, developers of middleware that would run on those server operating systems should have access to the same communications protocols that Windows client operating systems use to communicate with applications running on Microsoft server operating systems. The provision, according to Judge Kollar-Kotelly, “acknowledges the continuing change in the industry [toward network computing] and expands appropriately from the imposition of liability” to assure that “the core of the decree [will not] prove prematurely obsolete.”<sup>45</sup> In her opinion extending the judgments, she characterized Section III.E as “the cornerstone” of the remedies and “the basis on which the parties and the Court aspired to have the applications barrier to entry broken down over time.”<sup>46</sup>

Recognizing that Section III.E and other provisions would pose technical challenges, the consent decree provides for the creation and maintenance, at Microsoft’s expense,<sup>47</sup> of a Technical Committee (TC) of “experts in software design and programming”<sup>48</sup> to “monitor Microsoft’s compliance”<sup>49</sup> and report to the plaintiffs regularly.<sup>50</sup> The TC can investigate complaints,<sup>51</sup> gather information from Microsoft,<sup>52</sup> and study Microsoft’s source code.<sup>53</sup> Although its members must be unbiased<sup>54</sup> in the sense of not having financial or professional interests for or against Microsoft, the TC “exists to assist the government in enforcing the decree.”<sup>55</sup> It is thus an instrumentality of the plaintiffs and not a neutral arbiter of disputes over enforcement. Although the nonsettling states’ judgment, at the states’ own request, did not provide for a technical committee,<sup>56</sup> those plaintiffs hired a technical consultant who has

---

<sup>44</sup> *D.D.C. States Remedy 2002*, 224 F. Supp. 2d at 129 (observing that § III.E keeps open “the new model of the ‘platform threat’”).

<sup>45</sup> *D.D.C. Tunney Act 2002*, 231 F. Supp. 2d at 202 (holding that § III.E is “appropriately forward-looking,” and “closely connected with the theory of liability in this case”).

<sup>46</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 181.

<sup>47</sup> *Consent Decree 2006*, *supra* note 1, § IV.B.6–8.

<sup>48</sup> *Id.* § IV.B.2.

<sup>49</sup> *Id.* § IV.B.8.a.

<sup>50</sup> *Id.* § IV.B.8.e.

<sup>51</sup> *Id.* § IV.B.8.d.

<sup>52</sup> *Id.* § IV.B.8.b.

<sup>53</sup> *Id.* § IV.B.8.c. Unlike the final judgment proposed to the trial judge, the consent decree does not require the disclosure of Windows source code to third parties. Judge Kollar-Kotelly held that this provision allowed all of the access necessary to the legitimate concerns of the decree. *D.D.C. Tunney Act 2002*, 231 F. Supp. 2d at 193.

<sup>54</sup> The decree disqualifies individuals with ties to Microsoft or its opponents. *Consent Decree 2006*, *supra* note 1, § IV.B.1. It provides that each side will select a member and those two will select the third, all to serve five-year, renewable terms. *Id.* § IV.B.3–4.

<sup>55</sup> *D.D.C. 2002 Tunney Act*, 231 F. Supp. 2d at 199.

<sup>56</sup> *D.D.C. States’ Remedy 2002*, 224 F. Supp. 2d at 182.



worked so closely with the TC that court and the parties view them as a single entity.<sup>57</sup>

Several characteristics of Section III.E should have warned the parties and the court of the enforcement problems that lay ahead. Antitrust recognizes a presumptive right to refuse to deal, a right that courts override only in narrowly defined circumstances.<sup>58</sup> This qualified right rests on a recognition that mandatory contracting is less likely than free contracting to produce wealth-enhancing agreements,<sup>59</sup> and that “[e]nforced sharing . . . requires antitrust courts to act as central planners, identifying the proper price, quantity, and other terms of dealing—a role for which they are ill suited.”<sup>60</sup> Courts will compel a defendant to deal only in limited circumstances. For example, it is a necessary, but not a sufficient, condition for mandatory dealing that rivals “cannot effectively compete without [the resources] and that duplication or practical alternatives are not available.”<sup>61</sup> Mandatory dealing may also be inappropriate if a defendant has not “voluntarily engaged in a course of dealing with its rivals,”<sup>62</sup> which it terminated for no plausible business reason.<sup>63</sup>

The protocol licensing provision runs counter to these considerations. It is forward-looking, but not merely in the sense that it aims to foreclose the defendant’s illegal acts in a different, future technological context. It aims to foreclose acts that the government did not prove had

---

<sup>57</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 157 n.25.

<sup>58</sup> *NYNEX Corp. v. Discon, Inc.*, 525 U.S. 128, 137 (1998) (observing that “the freedom of the individual right to contract when not unduly or improperly exercised [is] the most efficient means for the prevention of monopoly”) (quoting *Standard Oil Co. v. United States*, 221 U.S. 1, 62 (1911)); *United States v. Colgate & Co.*, 250 U.S. 300, 307 (1919) (observing that a trader has a right “freely to exercise his own independent discretion as to parties with whom he will deal” absent an intent to monopolize).

<sup>59</sup> William H. Page, *Mandatory Contracting Remedies in the American and European Microsoft Cases*, 75 ANTITRUST L.J. 787 (2009); John E. Lopatka & William H. Page, *Bargaining and Monopolization: In Search of the “Boundary of Section 2 Liability” Between Aspen and Trinko*, 73 ANTITRUST L.J. 115, 124–26 (2005).

<sup>60</sup> *Verizon Commc’ns Inc. v. Law Offices of Curtis V. Trinko, LLP*, 540 U.S. 398, 408 (2004).

<sup>61</sup> Phillip Areeda, *Essential Facilities: An Epithet in Need of Limiting Principles*, 58 ANTITRUST L.J. 841, 852 (1989). For criticism of Areeda’s approach, see Spencer Weber Waller, *Areeda, Epithets, and Essential Facilities*, 2008 Wis. L. Rev. 359.

<sup>62</sup> *Trinko*, 540 U.S. at 409.

<sup>63</sup> *Compare Covad Commc’ns Co. v. BellSouth Corp.*, 374 F.3d 1044, 1049 (11th Cir. 2004) (“*Trinko* now effectively makes the unilateral termination of a voluntary course of dealing a requirement for a valid refusal-to-deal claim under *Aspen*.”), with *Helicopter Transp. Servs., Inc. v. Erickson Air-Crane Inc.*, No. 06-3077-PA, 2008 WL 151833, at \*9 n.10 (D. Or. Jan. 14, 2008) (“*Trinko* said no such thing.”).

ever occurred in any context.<sup>64</sup> Even if we assume that the remedy responds, in a broad sense, to Microsoft's illegal conduct, it goes beyond accepted prudential limitations on mandatory dealing. It involves complex technical questions that are beyond the court's understanding and, thus, required the creation of a technical committee. It imposes a duty of dealing, even though Microsoft had not terminated a profitable course of licensing its protocols; indeed, unlike the API licensing requirement, the affected protocols had never been formally documented, even within Microsoft.<sup>65</sup> The order thus required Microsoft to unearth functionality that "exist[ed] only deep within the bowels" of Windows and that required the creation of altogether "[n]ew systems . . . simply to make . . . access possible."<sup>66</sup> As we note below, Microsoft and two external consulting firms have had to scrutinize the source code of Windows to identify the protocols and protocol elements it contained.

Finally, Section III.E required dealing, even though there was no evidence that there were many firms that wanted what the court was ordering. Granted, few firms could have known enough about the protocols, given Microsoft's control over them, to decide whether they would be useful.<sup>67</sup> But, as Judge Kollar-Kotelly recognized, "[t]here are a variety of methods [other than native interoperation] used to overcome differences between client and server capabilities."<sup>68</sup> Some of these methods, such as reliance on generic industry-standard protocols supported in Windows or addition of new software to the client, are more widely used and, thus, preferable to native interoperation for many developers. Efforts to assess interest in licensing the protocols began in earnest only after the entry of the judgment. As the court of appeals observed in warily affirming the provision, there are "difficulties inherent in crafting

---

<sup>64</sup> If, despite the considerations underlying the presumptive right to refuse to deal, the court finds that a refusal to deal is unlawful, then compulsory licensing may be an appropriate remedy. For discussion of this issue, see U.S. Dep't of Justice & Fed. Trade Comm'n, Antitrust Enforcement and Intellectual Property Rights: Promoting Innovation and Competition 22-23 (2007), <http://www.usdoj.gov/atr/public/hearings/ip/222655.pdf>. In *Microsoft*, none of the proven violations involved a refusal to provide compatibility information.

<sup>65</sup> See *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 164.

<sup>66</sup> *Trinko*, 540 U.S. at 410.

<sup>67</sup> One exception is the Samba project, which uses protocol analysis to create products that emulate Microsoft server operating systems. William H. Page & Seldon J. Childers, *Bargaining in the Shadow of the European Microsoft Decision: The Microsoft-Samba Protocol License*, 102 Nw. U. L. REV. COLLOQUY 332, 335 (2008), available at <http://www.law.northwestern.edu/lawreview/colloquy/2008/16/LRColl2008n16Page&Childers.pdf>.

<sup>68</sup> *D.D.C. States Remedy 2002*, 224 F. Supp. 2d at 122.

a forward-looking provision concerning a type of business conduct as to which there has not been a violation of the law.”<sup>69</sup>

## B. IMPLEMENTATION

Section III.E has proven to be by far the most difficult provision of the final judgments to implement.<sup>70</sup> We have detailed the tumultuous history of enforcement of Section III.E elsewhere,<sup>71</sup> and Judge Kollar-Kotelly, in her opinion extending the judgments, goes over the same ground.<sup>72</sup> In this Part, we will limit our discussion to three principal themes that have emerged from the experience. First, the parties, after years of wrangling, have resolved many issues relating to pricing or terms of access to the licenses and documentation. Microsoft delivered the first version of the license and 5,000 pages of technical documentation in August 2002.<sup>73</sup> The plaintiffs focused their early compliance efforts on making the terms of the license more attractive<sup>74</sup> and on “evangelizing” the protocols to developers.<sup>75</sup> Even though the final judgments allow Microsoft to charge a reasonable royalty and impose reasonable restrictions on the licenses, Microsoft has repeatedly reduced both the price of the license and the restrictions on its use<sup>76</sup> until both now approach zero. In 2006, Microsoft announced a temporary royalty holiday,<sup>77</sup> which was later extended indefinitely.<sup>78</sup> In 2008, Microsoft went

---

<sup>69</sup> *D.C. Circuit 2004 Remedy*, 373 F.3d at 1223.

<sup>70</sup> Judge Kollar-Kotelly identified six enforcement issues that have arisen apart from § III.E. *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 158.

<sup>71</sup> Page & Childers, *supra* note 35, at 112–26.

<sup>72</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 158–64.

<sup>73</sup> U.S. Dep’t of Justice, Microsoft Consent Decree Compliance Advisory (Apr. 21, 2003), available at <http://www.usdoj.gov/atr/cases/f200900/200957.pdf>; Joint Status Report on Microsoft’s Compliance with the Final Judgments, *United States v. Microsoft Corp.*, No. 98-1232, at 22 (D.D.C. filed July 17, 2003) [hereinafter *JSR July 2003*], available at <http://www.usdoj.gov/atr/cases/f201100/201135.pdf>.

<sup>74</sup> *JSR July 2003*, *supra* note 73, at 8–9.

<sup>75</sup> Interim Joint Status Report on Microsoft’s Compliance with the Final Judgments at 10–11, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Oct. 17, 2003), available at <http://www.usdoj.gov/atr/cases/f201300/201386.pdf>.

<sup>76</sup> Joint Status Report on Microsoft’s Compliance with the Final Judgments at 15, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Jan. 16, 2004), available at <http://www.usdoj.gov/atr/cases/f202100/202129.pdf>.

<sup>77</sup> Joint Status Report on Microsoft’s Compliance with the Final Judgments at 12, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Aug. 30, 2006) [hereinafter *JSR August 2006*], available at <http://www.usdoj.gov/atr/cases/f218000/218096.pdf>.

<sup>78</sup> Joint Status Report on Microsoft’s Compliance with the Final Judgments at 5–6, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. June 19, 2007), available at [http://ag.ca.gov/cms\\_pdfs/press/2007-06-19\\_Final\\_JSJ.pdf](http://ag.ca.gov/cms_pdfs/press/2007-06-19_Final_JSJ.pdf).

further still and made all of the documentation available free on its Web site as part of its newly adopted "Interoperability Principles."<sup>79</sup>

Second, the parties have resolved any issues concerning technical support for licensees. The plaintiffs have insisted that Microsoft provide whatever technical assistance any developer might want to implement the protocols. Microsoft has offered increasing amounts of technical support to licensees,<sup>80</sup> apparently far more than any licensee has actually used. It has given licensees access to Windows source code<sup>81</sup> as well as other forms of support, including dedicated account managers,<sup>82</sup> "plug-fests,"<sup>83</sup> and an interoperability lab,<sup>84</sup> that have sometimes gone begging for participants.<sup>85</sup>

Third, despite the expenditure of staggering amounts of money and brainpower over six years, the parties have not resolved issues concerning the documentation of the protocols. The government has, from the outset, interpreted Section III.E to require disclosures "sufficient . . . to allow licensees fully to utilize all the functionality of each Communications Protocol."<sup>86</sup> The plaintiffs have argued, and Judge Kollar-Kotelly

<sup>79</sup> Joint Status Report on Microsoft's Compliance with the Final Judgments at 12, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Feb. 29, 2008) [hereinafter *JSR February 2008*], available at <http://www.usdoj.gov/atr/cases/f230600/230647.pdf>. The program commits to licensing specified proprietary protocols at lower rates than they had been licensed under the MCPP. See *Microsoft Corp., Interoperability Principles: Open Connections, Standards Support, Data Portability* (Feb. 21, 2008), <http://www.microsoft.com/interop/principles/default.mspix>.

<sup>80</sup> Supplemental Joint Status Report on Microsoft's Compliance with the Final Judgments at 10, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Nov. 18, 2005), available at <http://www.usdoj.gov/atr/cases/f213100/213109.pdf> (offering 500 hours of free technical support and consulting package to each licensee).

<sup>81</sup> Joint Status Report on Microsoft's Compliance with the Final Judgments at 4, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Feb. 8, 2006) [hereinafter *JSR February 2006*], available at <http://www.usdoj.gov/atr/cases/f214500/214518.pdf>.

<sup>82</sup> More than half of the licensees have signed up for this form of support. Joint Status Report on Microsoft's Compliance with the Final Judgments at 9-10, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. June 17, 2008) [hereinafter *JSR June 2008*], available at <http://www.usdoj.gov/atr/cases/f234100/234119.pdf>.

<sup>83</sup> *Id.* at 14. A plug-fest is an event in which Microsoft engineers help developers test and debug protocols the developers use with their products. *JSR August 2006, supra* note 77, at 6.

<sup>84</sup> *JSR June 2008, supra* note 82, at 13. An interoperability lab is a facility that provides "training, best practices, trouble-shooting and technical support for licensees implementing protocols from the MCPP documentation." *JSR August 2006, supra* note 77, at 6.

<sup>85</sup> *JSR February 2008, supra* note 79, at 19-20; Joint Status Report on Microsoft's Compliance with the Final Judgments at 17-18, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Mar. 6, 2007) [hereinafter *JSR March 2007*], available at <http://www.usdoj.gov/atr/cases/f221700/221759.pdf>.

<sup>86</sup> Competitive Impact Statement at 36, *United States v. Microsoft Corp.*, No. 98-1232, (D.D.C. Nov. 15, 2001), available at <http://www.usdoj.gov/atr/cases/f222900/222994.pdf>.

has agreed, that Microsoft must provide “certifiably complete, accurate, and useable” documentation. The TC has measured Microsoft’s compliance with this requirement against three successive sets of technical standards the parties have adopted over the years.<sup>87</sup> The TC’s principal method of assuring the adequacy of the documentation under these standards has been to develop prototype implementations of each protocol.<sup>88</sup> When the TC staff encounters a problem in the development process, it reports the problem to Microsoft as a bug or Technical Documentation Issue (TDI), which is given a priority level, depending upon its perceived importance. The TC’s measure of the adequacy of Microsoft’s documentation is the success of the TC’s software engineers in creating working prototypes: the accumulation of unresolved TDIs quantifies Microsoft’s shortcomings in compliance.<sup>89</sup> Microsoft has contributed to the testing process with a number of initiatives, including developing “protocol parsers” that allow engineers to view protocols in real-time network traffic,<sup>90</sup> and “test suites” that parallel the prototype implementations.<sup>91</sup>

In the spring of 2006, the accumulation of unresolved TDIs became so great that the parties were forced to rethink the process. Microsoft’s senior vice president Robert Muglia concluded that the “process of trying to fix issues identified by the TC one at a time” was not working and that Microsoft must “rewrite substantial portions of the documentation, taking advantage of what it has learned during the last several years, including all of the specific reports from the TC.”<sup>92</sup> This “reset” required the parties to agree on a new “overarching specification” with a series of five “milestones” for Microsoft’s delivery of rewritten documentation.<sup>93</sup>

---

The final judgments require Microsoft to make the protocols available “for use” in interoperation.

<sup>87</sup> Page & Childers, *supra* note 35, at 122.

<sup>88</sup> See, e.g., Joint Status Report on Microsoft’s Compliance with the Final Judgments at 3–4, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Jan. 25, 2005) [hereinafter *JSR January 2005*], available at <http://www.usdoj.gov/atr/cases/f207200/207283.pdf>.

<sup>89</sup> See, e.g., *JSR February 2006*, *supra* note 81, at 8.

<sup>90</sup> *JSR January 2005*, *supra* note 88, at 5. The grander version of the parser plan, called Project Troika, turned out to be overly ambitious and was later scaled back. Page & Childers, *supra* note 35, at 119–20. Judge Kollar-Kotelly identified these failures as evidence of Microsoft’s failure to commit adequate resources to the project. *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 160–61.

<sup>91</sup> *JSR March 2007*, *supra* note 85, at 17. The test suites resemble the prototype implementations, but test the documentation against the Windows code rather than an implementation.

<sup>92</sup> Joint Status Report on Microsoft’s Compliance with the Final Judgments at 6, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. May 12, 2006), available at <http://www.usdoj.gov/atr/cases/f216100/216127.pdf>.

<sup>93</sup> *JSR August 2006*, *supra* note 77, at 13.

Microsoft has completed that process and delivered the new documentation,<sup>94</sup> which now exceeds 25,000 pages.<sup>95</sup> The parties also agreed to modify the final judgment to extend the protocol licensing requirement for at least two years and to require Microsoft to keep Muglia in charge of the documentation project.<sup>96</sup> The process of testing both the pre-reset and post-reset documentation by the use of prototype implementations continues, and new TDIs continue to arise. Microsoft's own testing also continues.<sup>97</sup> The process has been complicated by Microsoft's discovery of several new protocols in Windows source code.<sup>98</sup> Microsoft and two external consulting firms have scoured Windows source code to identify any other overlooked protocols.<sup>99</sup>

In a recent status report, the plaintiffs referred to a relatively new commitment from Microsoft to produce "overview documents that explain how the MCPP protocols work together" in "complex scenarios."<sup>100</sup> The plaintiffs emphasized that the documentation was "essential" to resolve "numerous" issues with the documentation and that it "disagrees" with Microsoft about the necessity of the documentation under the Final Judgments.<sup>101</sup> The disagreement is a reference to Microsoft's position, expressed in its portion of the status report, that the TC's request for "subsystem documents" exceeds "the scope of the Final Judgments because it requires Microsoft to document system information that is not necessary for licensees to make use of the Communications protocols."<sup>102</sup> Nevertheless, Microsoft has agreed to produce nineteen of these "complex[ ], volum[inous], and novel[ ]" documents by March 2009.<sup>103</sup> This new undertaking has the potential to generate significant conflict, depending upon how far the TC presses its demands for documentation of multiple scenarios. The plaintiffs warned in a 2008 status report that

---

<sup>94</sup> *JSR February 2008*, *supra* note 79, at 3.

<sup>95</sup> Wolfgang Grieskamp et al., Model-Based Quality Assurance of Windows Protocol Documentation § 1, <http://research.microsoft.com/users/wrwg/ICST08.pdf>.

<sup>96</sup> *Consent Decree 2006*, *supra* note 1, § IV.E.

<sup>97</sup> *JSR June 2008*, *supra* note 82, at 13. For a description of Microsoft's testing process, see Grieskamp et al., *supra* note 95.

<sup>98</sup> *JSR March 2007*, *supra* note 85, at 11.

<sup>99</sup> *JSR February 2008*, *supra* note 79, at 5–6. Microsoft calls its audit process "Project Sydney." The first external consulting firm generated a report that led to additional training of Microsoft staff involved in the project. The second firm applied separate "programmatic methods for searching the Windows source code to identify communications protocols that should have been identified by Project Sydney." *Id.* at 5. This process had uncovered additional "protocol elements."

<sup>100</sup> *Id.* at 3.

<sup>101</sup> *Id.* at 3–4.

<sup>102</sup> *Id.* at 15.

<sup>103</sup> *JSR June 2008*, *supra* note 82, at 10–11.

“experience in preparing the system documents over time may result in changes to the current list of nineteen documents such as adding additional documents, combining documents, or shifting subject matter between documents.”<sup>104</sup>

### III. THE COURT’S MEASURE OF COMPLIANCE

Surveying the experience of Section III.E enforcement in her January 2008 opinion, Judge Kollar-Kotelly saw a history of Microsoft’s failures:

[A]t the time that the Court entered the Final Judgments, all parties involved anticipated that the technical documentation required under Section III.E would be released, at the latest, by February 2003. Instead, five years later, the rewrite of the technical documentation has only recently been completed, the corresponding overview/reference materials are not available to licensees, and the testing of the revised technical documentation is far from finished. As such, no one can deny that licensees do not yet have access to a set of usable, accurate, and certifiably complete technical documentation, as contemplated by Section III.E and by the Court’s various remedy-related opinions.<sup>105</sup>

These failures, “attributed solely to Microsoft,”<sup>106</sup> constituted the unforeseen change in circumstances that required extending other provisions of the final judgments for two more years.

As we explain in detail below, the court referred to three factors in evaluating compliance: (1) Microsoft’s commitment of resources and cooperation with the plaintiffs; (2) the number of licensees of the protocols and the number and type of products being produced under the licenses; and (3) the number of unresolved TDIs generated by the TC’s testing process. Although the court found the first two of these factors relevant in some respects, it was ultimately the third that was decisive. Microsoft had failed to produce documentation that is *certifiably* complete, usable, and accurate, and will not be in compliance until it does so. We suggest that this standard takes insufficient account of market realities. The shortcomings of the court’s approach become clear in Part IV, where we compare the compliance program with standard industry practice in competitive software development.

---

<sup>104</sup> *Id.* at 5.

<sup>105</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 170.

<sup>106</sup> *Id.* at 174.

## A. RESOURCES AND COOPERATION

Judge Kollar-Kotelly viewed the record of Microsoft's cooperation with the plaintiffs and its commitment of resources to the documentation project as a mixed bag. On the one hand, the court found that:

Microsoft has been overwhelmingly cooperative with the Plaintiffs and the TC over the past five years [and has been willing] to address issues as they arose and to negotiate solutions rather than force litigation. In many respects, Microsoft's conduct has been a model for parties engaged in complex and protracted litigation.<sup>107</sup>

The repeated negotiations "have achieved a worthy goal by obviating the need for compliance-related litigation [so that] the Court has never been asked to find Microsoft out of compliance with Section III.E."<sup>108</sup> Moreover, Microsoft has committed enormous resources to the task. It continuously increased its staff devoted, at least in part, to compliance from ten in July 2003<sup>109</sup> to eighty-three by March 2006<sup>110</sup> to 210 in May 2006,<sup>111</sup> and to 630 by January 2008.<sup>112</sup>

However, more than five years after the judgments called for the protocols to be available, "licensees do not yet have the benefit of a certifiably complete, accurate, and useable set of technical documentation."<sup>113</sup> Whatever the difficulties it faced, "Microsoft is culpable for this inexcusable delay"<sup>114</sup> and "practically speaking . . . has failed to comply with Section III.E."<sup>115</sup> Indeed, according to the court, Microsoft admitted its failure to comply by conceding that, before the reset, it "didn't have the exact right resources [or] the right process in place"<sup>116</sup> until its most recent increase in staffing. Although, admittedly, Section III.E "required Microsoft to document protocols that had never before been documented . . . Microsoft should have recognized the problems with its documentation attempts earlier, and directed the appropriate resources

---

<sup>107</sup> *Id.* at 181.

<sup>108</sup> *Id.* at 163.

<sup>109</sup> *Id.* at 161.

<sup>110</sup> *Id.* at 164 n.28.

<sup>111</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 161.

<sup>112</sup> *Id.* at 162 n.26. The number had grown to 750 by June 2008. *JSR June 2008*, *supra* note 82, at 14.

<sup>113</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 170.

<sup>114</sup> *Id.* at 163.

<sup>115</sup> *Id.* at 181.

<sup>116</sup> *Id.* at 164 (citations omitted).



towards creating the necessary documentation.”<sup>117</sup> Now, however, it apparently has “finally committed the resources necessary”<sup>118</sup> to comply.<sup>119</sup>

This characterization of Microsoft’s culpability depends upon the criterion of compliance. It is certainly the case that Microsoft, even with the latest commitment of resources, has failed to produce documentation that does not yield large numbers of TDIs that are difficult to resolve. It is not clear, however, that this criterion is the appropriate one. The number of TDIs is presumably related to the quality of the documentation in an abstract sense. But that number is also related to the TC’s choice to test the documentation by creating prototype implementations of each protocol, then reporting every problem it encountered in that process as a TDI. Even though Microsoft has agreed to make best efforts to comply with the plaintiffs’ demands, if the TDI’s testing model is inefficient, Microsoft can hardly be said to be solely culpable for failing to produce a “certifiably complete” response. As we will discuss below, a more appropriate criterion for compliance with the documentation requirement should be drawn from industry standards for competitive firms and developer needs.

#### B. LICENSEES AND PRODUCTS

Despite the improvements in the documentation after the reset, the relaxation of the terms of the license, and the provision of unlimited technical support, the number of licensees under the program has remained low. In July 2003, one year after the release of the initial version of the technical documentation, there were only four licensees.<sup>120</sup> By February 2008, the number had increased to forty-seven, of which only sixteen were actually producing products.<sup>121</sup> All of those products relate to specific server tasks. Although six companies have obtained general server licenses, none has produced a “general server product,” presumably the kind most likely to develop into a rival platform, the procompetitive scenario envisioned in the decree.<sup>122</sup>

In the extension proceedings, the moving states argued that the failure of licensees to produce a general server product justified extending

---

<sup>117</sup> *Id.* at 164.

<sup>118</sup> *Id.* at 182.

<sup>119</sup> Microsoft noted that “[s]ignificant attention to and involvement in the technical documentation and the MCPP extend through all levels of the Microsoft organization and draw upon the resources of numerous product engineering, business, technical, and legal groups, as well as company management.” *JSR February 2008*, *supra* note 79, at 20.

<sup>120</sup> *JSR July 2003*, *supra* note 73, at 22.

<sup>121</sup> *JSR February 2008*, *supra* note 79, at 14.

<sup>122</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 175.

the final judgments. Judge Kollar-Kotelly disagreed. Although Section III.E was premised on the possibility that a middleware platform might run on a non-Microsoft server, she noted, the fact that present licensees had not developed a product with those capabilities did not mean that the provision had failed. First, a present or future licensee may yet develop a general server product. She expected “that the MCPPE will continue to attract licensees . . . and that once certifiably complete, accurate, and useable technical documentation is available to those licensees they will develop products that ‘further the ability of [ ] non-Microsoft server operating systems to provide a platform which competes with Windows itself.’”<sup>123</sup> More important, the provision was “aspirational, rather than premised upon certain benchmarks”<sup>124</sup> in product development. The goal of the final judgments generally was to foreclose illegal conduct, not necessarily to eliminate or to reduce Microsoft’s lawfully acquired monopoly power.<sup>125</sup> Thus, for Judge Kollar-Kotelly, Microsoft’s compliance depends upon the completeness and accuracy of the documentation, not necessarily upon the number of licensees or the types of products they produce. If Microsoft produces complete and accurate documentation, the fact that there are no licensees with platform potential would apparently not prevent a finding of compliance. Once again, the standard by which the court measures the completeness and accuracy of the documentation is decisive.

This reasoning recognizes, yet oddly fails to address, the glaring disconnect between plaintiffs’ enforcement activities and any real-world benefit to developers or users. If it were the case that improving the documentation would attract more licensees, one would have expected the number of licensees to increase more rapidly in recent years. Despite the enormous increases in staffing by Microsoft and the TC, the improvements in the documentation, the provision of unlimited technical support, the reduction in the price of the license to zero, and extensive efforts in evangelizing the protocols, fewer than fifty firms have signed licenses, a rate of fewer than ten per year. Microsoft asserted in late 2005, without contradiction, that it was “unaware of any existing or potential licensee that has been unable to use any Communications Protocol because of flaws in the documentation.”<sup>126</sup> That statement apparently remains true. All of the developers with an interest in using

---

<sup>123</sup> *Id.*

<sup>124</sup> *Id.*

<sup>125</sup> *Id.* at 178.

<sup>126</sup> Joint Status Report on Microsoft’s Compliance with the Final Judgments at 14–15, *United States v. Microsoft Corp.*, No. 98-1232 (D.D.C. Oct. 8, 2004) [hereinafter *JSR October 2004*], available at <http://www.usdoj.gov/atr/cases/f205700/205751.pdf>.

Microsoft's protocols have access to them, with voluminous technical documentation and unlimited technical support. Thus, any actual licensee has the tools, with the assistance of Microsoft's technical support, to overcome any flaws in the documentation in its own implementation. In a recent status report, Microsoft listed only twelve outstanding TDIs of a total of 1,276 that had been identified by licensees.<sup>127</sup> The conclusion is inescapable that the TDI-based standard the plaintiffs and the court are applying is inappropriate to the needs of any real-world licensee.

The most plausible explanation for the dearth of licensees is that most developers writing to non-Microsoft servers do not need the protocols and documentation, no matter how complete, accurate, and usable they are. As the court recognized in its remedy opinion, there are various ways of achieving interoperability between Windows clients and applications running on Windows servers, only one of which is to use Microsoft's proprietary, native communications protocols. Developers may, for example, add their own software to the Windows client and, thus, achieve interoperability using the standard Windows API.<sup>128</sup> Alternatively, they may use standard protocols supported by Windows without a license. It is difficult to say how significant an increment in interoperability the native protocols provide. Now that Microsoft has posted the documentation on its Web site for free, the number of official licensees will become even less related to the competitive impact of the decree.<sup>129</sup>

From a competitive point of view, there is an even more fundamental flaw in the final judgments' focus on documenting native protocols. As

---

<sup>127</sup> *JSR June 2008*, *supra* note 82, at 12–13. Microsoft added that:

As to the category "TDIs identified by licensees," in most cases licensees do not open TDIs themselves. Licensees generally ask Microsoft questions about the documentation. Most questions do not result in any TDIs. In some cases, questions from licensees result in a TDI being filed by the Microsoft employees involved in answering the licensees' questions. In these circumstances, Microsoft categorizes the TDI as a licensee TDI.

*Id.* at 11 n.7. This passage implicitly makes clear that Microsoft resolves all but a handful of issues raised by licensees through its technical support.

<sup>128</sup> Interestingly, the states that were moving to extend the term of other provisions of the final judgments raised "the possibility that an MCPP licensee who writes an application using software hosted on a server will also wish to add 'software to the client to take advantage of the functionality that the server provides,'" observing that "certain companies, including Apple, Yahoo!, and Google, are already making use of a product approach that involves both server-side and client-side components, albeit via internet standard protocols rather than Microsoft's proprietary protocols." *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 171–72. The court endorsed this scenario, even though, as Microsoft noted, products that involve adding software to the client or that rely on industry standard protocols would not be affected by the § III.E disclosures. *Id.* at 172.

<sup>129</sup> Parts of the documentation had been downloaded over 146,000 times, as of June 2008. *JSR June 2008*, *supra* note 82, at 9.

we have seen, the goal of Section III.E is to preserve the middleware threat to the applications barrier to entry that protects the Windows monopoly in client operating systems. To the extent that threat ever existed for middleware running on either the client or server operating systems, it has been largely superseded by far more potent platform threats: Apple and Web services. Because Apple now uses Intel chips, it is undeniably a direct competitor of Microsoft. Its latest operating system is a dangerous rival for all versions of the Windows client, at least in the consumer space. Perhaps more significantly, versions of many of the most popular client applications run on Web servers rather than the client operating system.<sup>130</sup> Any user with an Internet connection and a standards-compliant browser now can access powerful programs running on non-Microsoft servers.

Microsoft argued in the extension proceedings that this proliferation of Web services had accomplished by a different avenue what the court had sought to achieve by Section III.E. Judge Kollar-Kotelly, however, refused to accept that Web services had made Section III.E irrelevant because new licenses under the provision “will presumably amplify whatever progress the proliferation of Web-based computing has made towards ‘eliminat[ing] the consequences of Microsoft’s illegal conduct.’”<sup>131</sup> But, as we have shown, it is unlikely that further efforts to improve the documentation under the current standards will greatly increase the number of licenses. Consequently, the protocols’ amplification of the competitive effects of Web services in the market for client operating systems is likely to be slight.

Judge Kollar-Kotelly was careful to distinguish competition in the market for client operating systems from competition in the market for server operating systems. Only the former was at issue in the U.S. *Microsoft* case: the goal of Section III.E was to foster the development of middleware running on servers that would reduce the applications barrier to entry into the market for client operating systems. Consequently, she rejected as irrelevant the states’ argument in the extension proceedings “that Microsoft has ‘consolidated its hold on the server market’ dur-

---

<sup>130</sup> See, e.g., Rachael King, *How Cloud Computing Is Changing the World*, BUS. WK., Aug. 4, 2008, available at [http://www.businessweek.com/technology/content/aug2008/tc2008082\\_445669.htm](http://www.businessweek.com/technology/content/aug2008/tc2008082_445669.htm) (reporting one estimate “that 12% of the worldwide software market would go to the cloud in” five years); Miguel Helft, *Google Pushes to Make Browser Applications More Powerful*, N.Y. TIMES, May 28, 2008, available at <http://bits.blogs.nytimes.com/2008/05/28/google-pushes-to-make-browser-applications-more-powerful/index.html> (quoting Google’s vice president of engineering as stating that “[i]f it was Windows versus the Web, the Web has won”).

<sup>131</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 175.

ing the five years that the Final Judgments have been effect."<sup>132</sup> However, the European *Microsoft* case, which did focus on the market for server operating systems, has had one effect that has important implications for the enforcement of the U.S. judgments: the Samba project, which produces server products that emulate Microsoft's, has obtained a license to the protocol documentation.<sup>133</sup> As we argue in the next Part, Samba is such an important player in the server market that its role as a licensee can provide an important benchmark for compliance with the U.S. judgment.

### C. FLAWED DOCUMENTATION

As we have shown in the last two Parts, Judge Kollar-Kotelly found that Microsoft had, "practically speaking," failed to comply with Section III.E because the documentation was not "certifiably complete, accurate, and usable"<sup>134</sup>:

As of the last status conference held in this case on September 11, 2007, Microsoft had released all initial versions of the rewritten technical documentation to the TC for testing, but the TC and Plaintiffs were far from concluding that the documentation was "substantially complete." The parties and the TC continue to conduct intensive testing of the technical documentation, and to review the corresponding overview/reference materials that Microsoft makes available to licensees. In addition, while the first consulting firm's audit of Microsoft's internal audit is complete, the second consulting firm's efforts are ongoing. Further, in its January 15, 2008 Supplemental Status Report, Microsoft reported that it is still in discussions with the TC regarding the overview/reference materials, and that testing of the rewritten documentation continues, with over 900 TDIs outstanding against the new documentation.<sup>135</sup>

The court placed the blame for the flaws in the documentation squarely on Microsoft. The TC, according to the court, had been unerringly fair and competent. Judge Kollar-Kotelly even thought it appropriate to

commend[ ] the members of the TC as . . . one of the most successful aspects of the Final Judgments, because it has been invaluable in facilitating the Plaintiffs' enforcement efforts . . . . [T]he TC has provided the Plaintiffs with crucial technical expertise by providing advice and evaluating Microsoft's compliance with the Final Judgments. The TC

---

<sup>132</sup> *Id.* at 176.

<sup>133</sup> Page & Childers, *supra* note 67, at 346-48.

<sup>134</sup> The court stated this formulation eight times. See *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 144, 163, 170, 175 (twice), 181, 183 n.34, 184.

<sup>135</sup> *Id.* at 163 (citations omitted).

has gone far beyond the simple “monitoring” with which it was tasked in the Settling States’ Final Judgment . . . to providing testing, feedback, and critiques that have proved critical to the Plaintiffs’ efforts to maximize the full potential of the Final Judgments’ remedies.<sup>136</sup>

The court voiced not a single word of criticism of the TC’s efforts, either in testing or in evaluating the documentation. In doing so, the court in essence appointed the TC as the sole arbiter of the sufficiency of the documentation under the standards laid down by the plaintiffs. By including the word “certifiably” in its test for compliance with the documentation requirement, the court evidently meant suitable for certification by the TC.

The plaintiffs have conceded that the quality of the documentation has improved as Microsoft has met the prescribed milestones in the most recent standard. The plaintiffs expressed confidence that “Microsoft will continue to improve the documents over time, based on feedback from the TC’s implementation and validation projects and from Microsoft’s own test suite project.”<sup>137</sup> Judge Kollar-Kotelly appeared to agree:

From all reports, it appears that Microsoft has finally committed the resources necessary, and adopted an approach likely to produce technical documentation that is complete, accurate, and usable to licensees. In light of the current RESET plan schedule and the parties’ most recent reports, the Court is unaware of any reason that the technical documentation required by Section III.E should not be complete, accurate, and usable to licensees long before Section III.E expires [in November 2009].<sup>138</sup>

In the most recent status reports, however, the plaintiffs make no representations about the likely date at which compliance will be achieved. Microsoft’s tables in the reports suggest that TDIs continue to be generated at least as quickly as they are resolved.<sup>139</sup>

Although the court has made the quality of the documentation its current criterion for compliance, it remains unclear whether meeting that standard will be sufficient to justify allowing the decree to expire. Judge Kollar-Kotelly cautioned:

[T]he door remains open for the Court to reassess the need for continued oversight as the expiration of the Final Judgments in November 2009 approaches. At that point, certifiably complete, accurate, and usable technical documentation will, presumably, be available to licen-

<sup>136</sup> *Id.* at 157.

<sup>137</sup> *JSR February 2008*, *supra* note 79, at 3.

<sup>138</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 182–83.

<sup>139</sup> *JSR June 2008*, *supra* note 82, at 12–13; *JSR February 2008*, *supra* note 79, at 18.

sees, and the Court will be in a far better position to evaluate Section III.E's implementation, as well as the role that the other provisions of the Final Judgments play in supporting Section III.E.<sup>140</sup>

This language appears to contemplate a further inquiry beyond compliance with the disclosure and licensing requirements to determine if Microsoft has met its obligations.

#### IV. A MARKET-BASED MEASURE OF COMPLIANCE

Antitrust remedies should ordinarily be confined to stopping proven violations and preventing their recurrence. As the D.C. Circuit wrote in its 2001 opinion, a judgment should be "tailored to fit the wrong creating the occasion for the remedy."<sup>141</sup> If the defendant's illegal conduct ends, the market's own forces usually will act to destroy inefficient monopolies.<sup>142</sup> When courts attempt to go beyond these bounds, the results are likely to be costly and ineffective, and potentially harmful to competitive markets. The protocol licensing provision departs from these standards because it does not respond to proven violations, does not satisfy any demonstrable market need, and because it requires the creation and marketing of an immensely complex product that never existed before.

For the remainder of this article, however, we assume that parties were correct to undertake this task in the consent decree and that the court was correct to approve it. The question remains what standard the court should apply to determine whether the defendant has complied with the provision. We agree with Judge Kollar-Kotelly that the standard should not be whether a licensee has actually created a platform that rivals the Windows client. Once the court has removed artificial barriers to the entry of middleware developers, the market will determine which platform succeeds. We also agree that the fact that the defendant has expended enormous resources should not be a measure of compliance. Finally, we agree that the standard of compliance should be whether the documentation is complete, accurate, and usable. We disagree, however, that the measure of completeness, accuracy, and usability should be whether the plaintiffs certify the documentation as completely free of issues.

---

<sup>140</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 184.

<sup>141</sup> *D.C. Circuit 2001*, 253 F.3d at 107.

<sup>142</sup> Network effects complicate this analysis because they can make market power more durable, but they do not change the general point that markets have powerful self-correcting mechanisms. For further discussion of the relationship between network effects and market power, see PAGE & LOPATKA, *supra* note 1, ch. 3.

Because the goal of any antitrust remedy is to restore competitive conditions to the market, a court should evaluate a remedy by the standards of the market. Accordingly, the project should be judged by the criteria that competitive software developers would apply. The goal should be to remove real obstacles to the entry of platform developers. The software at issue in this case is a form of platform interface. If a competitive producer of a software platform were to choose to extract and document its interfaces in an effort to maximize its products' interoperability with rival platforms, it would not apply a standard of perfection to its documentation. In this Part, we offer a brief outline of what a market-based measure of compliance would entail. We also note that, perhaps ironically, the European antitrust authorities have apparently adopted something like a market-based standard in implementing their protocol licensing order against Microsoft.

#### A. SOFTWARE DEVELOPMENT UNDER MARKET CONSTRAINTS

To create a new product, commercial software developers first formulate a concept of what the software will do and express the concept in a set of "requirements definitions" for programmers.<sup>143</sup> Programmers evaluate the feasibility of developing software that meets the requirements definitions and estimate the time and money needed accomplish that task.<sup>144</sup> Once the programmers have returned a detailed, feature-by-feature budgetary estimate, management can then consider the costs projected to develop the features. At this point, management makes adjustments or negotiates with the programmers to refine the requirements to make the time and expense of the project manageable. The primary consideration in defining requirements is customer acceptance. Every feature the programmers create should be something customers actually want.<sup>145</sup> Features that sound good in initial brainstorming but have no practical use needlessly increase costs and delay. Even a useful feature may be inefficient to produce if the cost of its development out-

---

<sup>143</sup> P.M. HEATHCOTE, 'A' LEVEL ICT 183 (2003) (discussing "establishing the [project's] objectives" in terms of measurable (quantitative) and descriptive (qualitative) terms); JOHN McMANUS, MANAGING STAKEHOLDERS IN SOFTWARE DEVELOPMENT PROJECTS 1 (2004) (describing objectives in terms of "stakes" held by "stakeholders" as real-world interests, concerns, perceptions of rights, expectations, or even ownership).

<sup>144</sup> JEFFREY O. GRADY, SYSTEMS REQUIREMENTS ANALYSIS 7 (2006); DEAN LEFFINGWELL & DON WIDRIG, MANAGING SOFTWARE REQUIREMENTS 7 (2d ed. 2003) (discussing project failures due to poor specifications management).

<sup>145</sup> See *id.* at 93 ("Understanding user and stakeholder needs moves us from the technical domain of bits and bytes, where many developers are most comfortable, into the domain of real people and real-world problems."). For an entertaining discussion of the process of identifying which features to build, see Joel Spolsky, Set Your Priorities (Oct. 12, 2005), <http://www.joelonsoftware.com/articles/SetYourPriorities.html>.



weighs its utility and, hence, its contribution to the value (and the profit potential) of the product.<sup>146</sup>

Once the programmers, the marketing team, and management have approved a final set of requirements, the programmers develop a project plan, which includes a detailed list of tasks, milestones, and objectives. A project manager keeps a current schedule of what tasks the team has completed, what is remaining, and what is overdue. The project manager can adjust the plan based on unforeseen developments.<sup>147</sup> At any time, the project manager should be able to predict when the product will be ready to ship.

In the past, programmers often worked on a product until a complete initial version of the product was ready, and then released that version to testing essentially for the first time.<sup>148</sup> Under the most favored current development method, however, the programming team makes regular daily or weekly releases of live code to an internal testing team.<sup>149</sup> This approach allows internal testers to focus on discrete components of the product as they become available and to gain experience with the product. Testers become an essential part of the development process by providing discipline and regular feedback to the programmers.<sup>150</sup> Management also benefits by receiving feedback from the programming

---

<sup>146</sup> LEFFINGWELL & WIDRIG, *supra* note 144, at 44–45 (describing development as the most expensive alternative, and suggesting other ways to reframe user desires and perceptions in order to avoid the development expense). Marketing people may help the developers determine what potential customers want and, equally important, what they will pay for it.

<sup>147</sup> McMANUS, *supra* note 143, at 1 (describing the project manager's function as planning, executing, measuring monitoring, controlling, and reducing project risk); LEFFINGWELL & WIDRIG, *supra* note 144, at 185 (describing the role of a project manager in a software development company).

<sup>148</sup> PHILIP G. ARMOUR, *THE LAWS OF SOFTWARE PROCESS: A NEW MODEL FOR THE PRODUCTION AND MANAGEMENT OF SOFTWARE* 102 (2003) (discussing "the problems of big process"); *see also id.* at 110 (discussing the benefits of the modern "extreme programming" paradigm featuring frequent small releases).

<sup>149</sup> *Id.* at 97 (describing the ascension of the "Agile" software development methodology including frequent regular releases of code); *id.* at 109 (same regarding "Extreme Programming" methodology); TORGEIR DINGSOYR, *SOFTWARE PROCESS IMPROVEMENT: 11TH EUROPEAN CONFERENCE 1* (2004) ("Extreme programming (XP) . . . is focused on generating early releases of working products and aims to deliver business value from the very beginning of the project."); DWAYNE PHILIPS, *THE SOFTWARE PROJECT MANAGER'S HANDBOOK: PRINCIPLES THAT WORK AT WORK* 173 (2004) (describing the "Agile" software development methodology). Internal testing refers to the iterative process where developers provide regular releases to internal quality assurance testers for feedback. These internal testers are not using the software necessarily the way a customer might, but are confirming that requirements are being met and tolerances achieved. *See also id.* at 39 (defining what is a software "methodology").

<sup>150</sup> *See* PHILIPS, *supra* note 149, at 167 (describing the Microsoft development process emphasizing "testing during development instead of at the end" and "[t]he testers use

team throughout the course of the project. Regular testing thus helps ensure that the project meets its time and quality objectives. Software that meets the memorialized feature requirements (the “specification”) and that has been approved by an external alpha testing team<sup>151</sup> qualifies as a “release candidate,” ready for beta testing<sup>152</sup> by potential customers.

There are industry definitions, like ISO 9126,<sup>153</sup> that measure software testing quality in terms of functionality, reliability, efficiency, portability, maintainability, compatibility, and usability.<sup>154</sup> Developers may also measure the quality of a product under development by comparing code output to pre-written “test cases,” “use cases,” or “test scripts.”<sup>155</sup> Given a formal requirement definition, testing teams at an early stage can create narrative “scripts” describing what actions a user might take using a given set of information,<sup>156</sup> and then describe specifically how the system should respond or process the information.<sup>157</sup> Programmers often use

---

[the specification] to write the tests for the code the programmers are [concurrently writing”).

<sup>151</sup> “Alpha testing” is a limited form of external testing where the code is sent to a dedicated testing team or to a selected potential customer or customers. Software in the alpha testing stage is not considered viable or stable enough for “real” use. HEATHCOTE, *supra* note 143, at 339 (defining alpha testing).

<sup>152</sup> “Beta testing” occurs when software is released to a limited set of potential end-user customers. These customers may or may not be trained in formal testing procedures, but are usually invited to report errors or unexpected behaviors to a designated contact. Beta code may or may not be considered stable enough for “real” use by customers. Sometimes the developer needs feedback that can only be provided by actual use of the product in real-life “battlefield” conditions. “Acceptance testing” is final testing performed by a customer who is purchasing the product. This generally occurs in situations where a customer has ordered a custom software product be designed for its unique purposes. *Id.*

<sup>153</sup> See ISO 9126 Software Quality Characteristics, <http://www.sqa.net/iso9126.html> (for a summary of the standard).

<sup>154</sup> See also PHILIPS, *supra* note 149, at 166–67 (describing Microsoft’s “three dimensions of quality” as reliability (how good the product must be before shipping), feature set (the specifications or requirements), and schedule (the ship date)).

<sup>155</sup> See LEFFINGWELL & WIDRIC, *supra* note 144, at 148. In formal software design terms, a test “script” usually refers to a program (a “script”) that automates some portion of the testing process. But in general parlance, the terms test “case” and test “script” are used interchangeably. See also WILLIAM E. LEWIS, SOFTWARE TESTING AND CONTINUOUS QUALITY IMPROVEMENT 44 (2000) (“[a] test script guides the tester through a test and ensures consistency among separate executions of the test”).

<sup>156</sup> A given set of testing information is often referred to as a “case” or a “use case.”

<sup>157</sup> See, e.g., HEATHCOTE, *supra* note 143, at 189 (providing examples of how to draw up a test plan and how to choose tests for the test plan); see also *id.* at 193 (stating “[t]esting is a vitally important part of the project” and describing how a comprehensive test plan should be created and verified as complete at the outset of the project); LEFFINGWELL & WIDRIC, *supra* note 144, at 157 (describing the concept of “storyboarding” a use case); ARMOUR, *supra* note 148, at 13 (describing need for a rule-based statement of need as well as a specific detailed real-world example); LEWIS, *supra* note 155, at 157 (describing a process for creating process-based test scripts).

these test scripts as guides for their development work. Later, the testers use the scripts to confirm whether the programmers have met the requirements.<sup>158</sup> Scripts can be enhanced or modified during the course of development as the project's needs evolve.

A crucial constraint guiding all aspects of the development process is the ship date.<sup>159</sup> A developer must be ready to sell usable software to customers by the ship date or suffer losses. Before the product ships, software development is a continuing investment that generates no revenue.<sup>160</sup> Thus, from the beginning of the project, the ship date imposes a critical discipline on everyone involved. Because a workable product must ship by a certain date in order to make a profit, a development firm knows that it will certainly ship an imperfect product.<sup>161</sup> Perfection is unattainable, if only because the underlying technologies are themselves imperfect and because customer expectations are fluid.<sup>162</sup> Consequently, the programmers know that to deliver market-acceptable software by the ship date, they must plan to write the most difficult and important pieces of the project first to give the testers more time with those features.<sup>163</sup> The testers, in turn, must prioritize their feedback to the developers to fix the most challenging problems first. The closer a project gets to the ship date, the more the testing team must scrutinize its feedback and evaluate which items are necessary for market accepta-

---

<sup>158</sup> See LEFFINGWELL & WIDRIG, *supra* note 144, at 310 ("Deriving Test Cases from Use Cases").

<sup>159</sup> See MIKE GUNDERLOY, PAINLESS PROJECT MANAGEMENT WITH FOGBUGZ 4 (2007) (describing "evidence based" software development management system designed primarily to predict the ship date); ARMOUR, *supra* note 148, at 66 (discussing decision process regarding selecting an acceptable level of defects in order to meet the ship date); *id.* at 15 (observing that "the only natural goal of a software [development] group should be to put itself out of business as soon as possible"); PHILIPS, *supra* note 149, at 167 (describing one of the three dimensions of quality as the "schedule [which] is the ship date").

<sup>160</sup> See TOM LOVE, OBJECT LESSONS: LESSONS LEARNED IN OBJECT-ORIENTED DEVELOPMENT PROJECTS 243 (1993) (emphasizing the need to "[k]eep development costs under control . . . [s]uccessful software companies spend less than 20% of total expenses on R&D").

<sup>161</sup> MICHAEL A. CUSUMANO & RICHARD W. SELBY, MICROSOFT SECRETS: HOW THE WORLD'S MOST POWERFUL SOFTWARE COMPANY CREATES TECHNOLOGY, SHAPES MARKETS, AND MANAGES PEOPLE 89 (1995) (noting the "trade-off between functionality and ship-date"); PHILIPS, *supra* note 149, at 167 (describing the primary concept of "[r]eliability [as] how good the product must be before shipping").

<sup>162</sup> See PHILIPS, *supra* note 149, at 167 ("'Good enough' software is a variation of the test completion measure. Shipping a product containing known errors is not new . . ."); RICK D. CRAIG & STEFAN P. JASKIEL, SYSTEMATIC SOFTWARE TESTING 267 (2002) (observing that "[s]ometimes the risk of not shipping (due to competition, failure of an existing system, etc., may exceed the (business) risk of shipping a flawed product").

<sup>163</sup> See CRAIG & JASKIEL, *supra* note 162, at 69 (noting the need to "intelligently choos[e] what not to test (i.e., low-risk features), rather than just running out of time and not testing whatever was left on the ship date").

bility, even if they must leave a desirable feature on the table.<sup>164</sup> This imperative explains why “version 1.0” of a software project invariably ships with a feature set that is not yet fully developed.<sup>165</sup> Some of the most successful software products shipped unremarkable, or even notorious, first versions that improved only after marketplace acceptance and use.<sup>166</sup> Maintenance of the software after shipping provides additional real-world testing and opportunities for improvement.

Of course, no ship date is absolute. If the demands are exigent, managers can extend the date. Indeed, some software firms claim that they will not ship software before it is “ready,” suggesting that any ship date will always give way to concerns about quality. Still, no matter how well-intentioned the developer, practical economic concerns govern every project. Venture capital does not flow indefinitely without tangible economic results—the shipment of a commercial product.<sup>167</sup>

One type of product commonly associated with platform software is the software development kit (SDK), which enables developers of software applications to write compatible programs. Intel, for example, in promoting its platform distributed tools like software libraries “along with specifications information contained in informal working documents such as white papers . . . to train [software] designers and help them develop appropriate complements.”<sup>168</sup> Intel’s engineers developed these tools “primarily for their own benefit,” but disseminated them in SDKs to a growing network of developers, with necessary technical support.<sup>169</sup> The development of documentation for an API is analogous to development of an SDK.<sup>170</sup> The process of developing and supporting

---

<sup>164</sup> *Id.* at 266 (observing that “[a]t some point . . . the cost of continuing to test will exceed the value derived from the additional testing”).

<sup>165</sup> See PHILIPS, *supra* note 149, at 167 (observing that “Microsoft [normally] ships software containing known errors . . . but the software works well enough for the vast majority of intended users . . . Microsoft will both work into the long tail of the 90/10 rule-of-work curve”); CRAIG & JASKIEL, *supra* note 162, at 267 (observing that “[m]any of us have certainly recommended the release of a product that we felt was fairly poor in quality because it was better than what the user[s] currently had [—] Remember, we’re not striving for *perfection*, only *acceptable risk*”) (emphasis added).

<sup>166</sup> See PHILIPS, *supra* note 149, at 167.

<sup>167</sup> See, e.g., CRAIG & JASKIEL, *supra* note 162, at 267 (describing a case where the software development company wrote a best of breed tax application, but held it from release until very high-quality standards were met, and by the time of shipment, most potential customers had already purchased an inferior competitive product and were out of the market).

<sup>168</sup> ANNABELLE GAWER & MICHAEL A. CUSUMANO, PLATFORM LEADERSHIP: HOW INTEL, MICROSOFT, AND CISCO DRIVE INDUSTRY INNOVATION 61 (2002).

<sup>169</sup> *Id.*

<sup>170</sup> Not least is the concept that an SDK, like API documentation, is developed primarily for the use of developers.

such an SDK would follow the same methods of design, testing, and management as other software development projects.

#### B. EVALUATING THE PROTOCOL DOCUMENTATION PROJECT BY MARKET STANDARDS

The implementation of Section III.E is analogous to the creation of a software development kit for Microsoft's native communications protocols. The primary difference is that the goal of the project is to assist developers in writing applications for Microsoft's rivals' server platforms rather than for Microsoft's own. Thus, rather than providing libraries of code for standard tasks, Microsoft must write pseudo-code to describe the functionality of the interfaces. To evaluate the success of such a project, a court should look to the best practices that a competitive platform producer would use to accomplish the same goal. The process would follow the principles we have outlined above: definition of requirements based on real-world needs, testing directed to those requirements, and the establishment of a meaningful ship date with binding consequences for late shipment for both parties. Once the product has shipped, under that standard, compliance should be measured by meeting the needs of actual developers who are using the software in real-world scenarios. As we explain more fully below, this strategy is particularly appropriate now that one of those licensees is the Samba project, which is extraordinarily well placed to identify relevant flaws in both protocols and documentation. Under our suggested approach, the TC and the court can monitor compliance by the defendant's resolution of development problems in concrete cases.

Microsoft's failure to produce compliant documentation by the original expiration date of the final judgments amounts to a costly budget overrun on a software development project. Judge Kollar-Kotelly blamed the overrun on Microsoft's inability to foresee the level of resources required to produce acceptable interoperability documentation. But another shortcoming of the project was a lack of discipline that would have been imposed by practical commercial considerations in a market-based scenario, particularly consumer acceptance and the ship date. The set of possible features for any software product is enormous. To succeed, a development project must limit its features to commercially useful ones.

A commercial developer would establish a detailed set of requirements by evaluating real-world customer needs and meeting them efficiently within appropriate time constraints. By contrast, the Microsoft documentation is measured against a standard of perfection without meaningful time limits. The TC's staff has functioned like a carefully isolated alpha-testing facility. Rather than developing testing scripts or

use cases based on what developers actually want, however, the TC has developed prototype implementations. From available descriptions, these implementations appear to be more like automated tests based on the documentation Microsoft already produced than standalone scenarios describing tasks real-world licensees would have to accomplish. Thus, the development of test cases by the TC, rather than real-world needs, defines what the Microsoft developers are required to do. The seeming endless generation of TDIs bears no necessary relationship to the needs of developers. Instead of a ship date, the parties faced the expiration of the final judgments. Under the court's standard of compliance, however, if the documentation is not certifiably complete, accurate, and useable, the deadline must be extended. Since the plaintiffs control that certification, the deadline imposes no constraint on them. Indeed, because Microsoft bears all of the costs of enforcement, there is little incentive for the plaintiffs, who favor extension of the final judgments at least until 2012,<sup>171</sup> to bring the project to a close.

Microsoft has claimed, without contradiction by the plaintiffs, that it is "unaware of any existing or potential licensee that has been unable to use any Communications Protocol because of flaws in the documentation."<sup>172</sup> Moreover, although Microsoft has not formally contested the plaintiffs' standards of compliance, it "firmly believes that the current protocol documentation available to implementers is fully useable and complies completely with the Final Judgments."<sup>173</sup> The competitive model supports this view, primarily because developers are already using the documentation to produce real-world products. Indeed, as we have noted, under the current program, only a handful of issues developers raised are ever reported as TDIs: most are simply resolved by Microsoft's technical support. Prototype implementations might be justified as a form of alpha testing because of the plaintiffs' special needs in implementing a supervised judgment. Once the product has been released to licensees, however, the focus should shift to the suitability of the documentation to their actual and potential needs. Microsoft and the TC can identify and address flaws in the protocols or the documentation based

---

<sup>171</sup> *D.D.C. Decree Extension 2008*, 531 F. Supp. 2d at 181–84 (declining to extend the judgments beyond November 12, 2009, but leaving open "the possibility of doing so in the future"). Microsoft has agreed that the plaintiffs have the sole discretion to move to extend the enforcement of Section III.E until 2012, and that Microsoft would not oppose the motion. *Id.* at 162. Because the court's refusal to extend the other provisions judgments the extra three years is contingent on Microsoft's creation of compliant documentation, the plaintiffs would seem to have an incentive to demonstrate the documentation remains flawed indefinitely.

<sup>172</sup> *JSR October 2004*, *supra* note 126, at 14–15.

<sup>173</sup> *JSR February 2008*, *supra* note 79, at 15.

upon issues generated by development. One source of information could involve monitoring support measures. The announcement of the interoperability principles and the publication of the documentation on Microsoft's Web site have allowed anyone interested in the documentation to use it for their development projects. Users can post issues to Microsoft Developer Network user forums and receive assistance from Microsoft or from other developers.<sup>174</sup> The court and the TC can monitor Microsoft's resolution of these issues.

Interestingly, the European Commission has apparently applied something like the approach we propose in its evaluation of the documentation Microsoft has created for the Workgroup Server Protocol Program, which includes server-to-server protocols. Last fall, the European Commission announced that Microsoft was in compliance:

In line with the 2004 Decision Microsoft now provides the interoperability information on reasonable and non-discriminatory terms. The interoperability information made available by Microsoft also appears to be complete and accurate *to an extent that a software development project can be based on it*. Therefore, the Commission considers that Microsoft is now complying with its obligations under the 2004 Decision. However, licensees may raise additional issues when they obtain access to the information and Microsoft must keep the interoperability information updated and fix errors on an ongoing basis.<sup>175</sup>

This passage explicitly measures compliance based on the usability of the software by a real development project, subject to a condition of continuing support.

As we noted in the last Part, it is significant that the Samba project, which produces server software that emulates Microsoft's server operating systems, obtained a license to Microsoft's protocols and documentation following the resolution of the 2007 European *Microsoft* decision.<sup>176</sup> The Samba team's use of Section III.E is likely to lead to better practical documentation than anything the TC's staff might do at this stage. The Samba project is particularly qualified in this regard because its entire development model is based on protocol analysis. As the Samba team members explore and then apply the information disclosed by Microsoft, they will ask questions, report problems, and provide external reinforcement of the documentation's usefulness in their own forums.

---

<sup>174</sup> See MSDN Forums: Open Protocol Specifications, <http://forums.microsoft.com/MSDN/default.aspx?ForumGroupID=573&SiteID=1>.

<sup>175</sup> Press Release, European Comm'n, Antitrust: Commission Ensures Microsoft's Compliance with the 2004 Decision—Frequently Asked Questions (emphasis added), available at <http://europa.eu/rapid/pressReleasesAction.do?reference=MEMO/07/420&format=HTML&aged=0&language=EN&guiLanguage=EN>.

<sup>176</sup> Page & Childers, *supra* note 67, at 346–48.

Customers using Samba-based networks all over the world will provide feedback to the Samba team, who in turn will attempt to rely on the Microsoft documentation to solve those real problems. Should the documentation prove insufficient or be of poor quality, the Samba team will certainly complain. Microsoft will, as it is organized to do as a commercial software and technical documentation developer, respond to those developers' complaints and improve the documentation in successive versions.

## V. CONCLUSION

Section III.E of the final judgments in the American *Microsoft* case has been the focus of the parties' efforts to implement the final judgments during their initial five-year term. The problems of enforcing the provision became so acute that the parties agreed in 2006 to extend that provision for up to five more years. The same problems have now provided grounds for extension of most of the other provisions in judgments for at least two years. The enormous costs of this process coupled with its meager results suggest that the court should reconsider the standards of compliance with Section III.E. The current approach focuses on identifying flaws in the technical documentation by using it to create prototype implementations of the protocols with no necessary relationship to market needs. We suggest that the parties and the court should redirect the inquiry to whether the defendant has produced documentation and support methods that meet the standards of the market and that are sufficient to meet the needs of developers.



