Washington Law Review

Volume 94 | Number 1

3-1-2019

Crashworthy Code

Bryan H. Choi

CRASHWORTHY CODE

Bryan H. Choi*

Abstract: Code crashes. Yet for decades, software failures have escaped scrutiny for tort liability. Those halcyon days are numbered: self-driving cars, delivery drones, networked medical devices, and other cyber-physical systems have rekindled interest in understanding how tort law will apply when software errors lead to loss of life or limb.

Even after all this time, however, no consensus has emerged. Many feel strongly that victims should not bear financial responsibility for decisions that are entirely automated, while others fear that cyber-physical manufacturers must be shielded from crushing legal costs if we want such companies to exist at all. Some insist the existing liability regime needs no modernist cure, and that the answer for all new technologies is patience.

This Article observes that no consensus is imminent as long as liability is pegged to a standard of "crashproof" code. The added prospect of cyber-physical injury has not changed the underlying complexities of software development. Imposing damages based on failure to prevent code crashes will not improve software quality, but will impede the rollout of cyber-physical systems.

This Article offers two lessons from the "crashworthy" doctrine, a novel tort theory pioneered in the late 1960s in response to a rising epidemic of automobile accidents, which held automakers accountable for unsafe designs that injured occupants during car crashes. The first is that tort liability can be metered on the basis of mitigation, not just prevention. When code crashes are statistically inevitable, cyber-physical manufacturers may be held to have a duty to provide for safer code crashes, rather than no code crashes at all. Second, the crashworthy framework teaches courts to segment their evaluation of code, and make narrower findings of liability based solely on whether cyber-physical manufacturers have incorporated adequate software fault tolerance into their designs.

Requiring all code to be perfect is impossible, but expecting code to be crashworthy is reasonable.

---

* Assistant Professor of Law and Computer Science & Engineering, The Ohio State University. I thank Christopher Yoo for leading me to this project. I also thank Matt Cooper, Paul Gatz, and Natasha Landon for excellent research assistance. Special thanks as well to Mikhail Belkin, Kiel Brennan-Marquez, Martha Chamallas, Jane Chong, Margot Kaminski, Deborah Merritt, Christina Mulligan, Efthimios Parasidis, Guy Rub, Andrew Selbst, Marc Spindelman, Chris Walker, and Rebecca Wexler for close reads of early drafts. This project benefitted immensely from input from Harry Surden, Nick Diakopoulos and the participants of the Junior Faculty Forum for Law & STEM. This work was supported in part by NSF CNS-1505799 and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy. All opinions, errors, and omissions are my own.

IN CODE WE TRUST 40
I. THE UNBEARABLE LIGHTNESS OF SOFTWARE LIABILITY 47
A. Consumer Protectionism: A Heavy Hand 50
B. Technology Protectionism: A Light Touch 58
C. Doctrinal Conventionalism: An Invisible Hand 60
II. THE UNREASONABLENESS OF CRASHPROOF CODE 62
A. Software's Intangible Form 65
B. Software's Innovation Function 71
C. Software's Complexity Anomie 79
III. CRASHWORTHY CODE: A RULE OF EQUANIMITY 86
A. On the Origin of Crashworthiness 91
B. Software Fault Tolerance: Translation from Cars to Code 100
1. Redundancy 103
2. Adjudication 106
3. Recovery 108
C. The Reasonable Fault-Tolerant System 110
CONCLUSION: MEMENTO MORI 115

IN CODE WE TRUST

In October 2004, Paramjit Singh entered the operating room for a routine heart bypass surgery.[1] A catheter was inserted into his heart, and a heart monitor device was used to ensure the catheter would not overheat. During the operation, the software controlling the heart monitor crashed, causing the catheter to burn and destroy Singh's heart. The hospital placed Singh into an artificial coma for eleven weeks, during which he suffered anoxic brain damage. Singh then received a heart transplant, but the anti-rejection drugs caused him to develop blood cancer, which required subsequent treatment by chemotherapy. Total medical bills were estimated at $2.7 million.

At trial, the evidence showed that the manufacturer of the heart monitor was aware of and had developed a fix for the software bug as early as 1998, but made a calculated business decision not to issue a recall or warning to any customers. Instead, monitors were patched on a rolling

1. For a fuller factual account, see *Singh v. Edwards Lifesciences Corp.*, 151 Wash. App. 137, 210 P.3d 337 (2009).

basis only when sent in for repair, and so the one used during Singh's operation had not yet been patched. The jury awarded Singh $31.75 million in compensatory damages plus an additional $8.35 million in punitive damages. The verdict was upheld on appeal.[2]

Singh's case was extraordinary on many dimensions, not least of which was the extremity of his injuries. The fact that Singh was helplessly under anesthesia at the time likely contributed additional opprobrium.[3] But by the same token, the accident was exceedingly rare: in more than six years of operation, the software had never before crashed mid-operation while a catheter was inserted in a patient's body.[4] Even so, the deliberate concealment of a simple bug fix may have made the manufacturer's cost-benefit decision seem especially callous.[5]

Perhaps the most surprising aspect of the case, however, is how unusual it is for a plaintiff like Singh to recover any legal damages at all. Tort liability for software failures is a rarity.[6] When Microsoft Word crashes and loses one's work, the only remedy is to restart, not to hire a lawyer.[7] If lucky, the software in question might provide some form of "crash

---

2. *Id.*

3. *Cf.* Ybarra v. Spangard, 154 P.2d 687 (Cal. 1944) (lending generous inferences where plaintiff was harmed while rendered unconscious for surgical treatment).

4. *Singh*, 151 Wash. App. at 141–42, 210 P.3d at 339–40 (noting only one other incident in Japan in October 2002). *But cf.* S. Austin Drive-In Theatre v. Thomison, 421 S.W.2d 933, 950 (Tex. Civ. App. 1967) ("We think that negligence simply means creating a risk that a reasonably prudent person would avoid and is not related to a statistical table of frequency of harm."); Huggins v. Stryker Corp., 932 F. Supp. 2d 972, 988 (D. Minn. 2013) ("The test is not whether the precise nature and manner of the plaintiff's injury was foreseeable, but whether the possibility of an accident was clear to the person of ordinary prudence." (quoting Domogala v. Rolland, 805 N.W.2d 14, 27 (Minn. 2011)).

5. *See* Nora Freeman Engstrom, *When Cars Crash: The Automobile's Tort Law Legacy*, 53 WAKE FOREST L. REV. 293, 330–32 (2018) (describing concealment of the Cobalt ignition switch problem); Gary T. Schwartz, *The Myth of the Ford Pinto Case*, 43 RUTGERS L. REV. 1013 (1991); W. Kip Viscusi, *Corporate Risk Analysis: A Reckless Act*, 52 STAN. L. REV. 547 (2000); *see also* Gen. Motors Corp. v. Johnston, 592 So. 2d 1054, 1060–61, 1064 (Ala. 1992) (approving $7.5 million award for punitive damages (reduced from $15 million) where automaker knew of engine stalling problems caused by defective computer chips, developed a solution, but concealed it from the public and did not issue a notice or recall).

6. *See* Jane Chong, *Bad Code: Exploring Liability in Software Development*, *in* CYBER INSECURITY: NAVIGATING THE PERILS OF THE INFORMATION AGE 69 (Richard M. Harrison & Trey Herr eds., 2016) [hereinafter CYBER INSECURITY]; Michael L. Rustad & Thomas H. Koenig, *The Tort of Negligent Enablement of Cybercrime*, 20 BERKELEY TECH. L.J. 1553, 1567, 1579 n.139 (2005); Michael Scott, *Tort Liability for Vendors of Insecure Software: Has the Time Finally Come?*, 67 MD. L. REV. 425, 430, 469 (2008) ("To date, there are no reported decisions in the United States holding a software vendor liable under a strict [products] liability theory.").

7. *See* Emily Kuwahara, Note, *Torts v. Contracts: Can Microsoft Be Held Liable to Home Consumers for Its Security Flaws?*, 80 S. CAL. L. REV. 997, 998 (2007).

recovery" that preserves one's documents and data, but the absence of such failsafe features triggers no legal penalty. Courts uniformly dismiss claims of software defect, often because there is no physical injury at stake,[8] but also for a broad range of other disqualifying reasons.[9] And even when the plaintiff alleges an eligible injury, it remains exceedingly difficult to prove whether the software *caused* the injury, and whether that cause was due to some *defect* intrinsic to the software.[10] The very fact that the manufacturer elected not to settle Singh's case suggests it believed it had a plausible chance of winning—even in the face of such troubling facts.[11]

---

8. *See, e.g.*, Taxes of P.R., Inc. v. Tax Works, Inc., No. 14-00279, 2014 WL 6604056 (W.D. Mo. June 16, 2014); Cotton Patch Café, Inc. v. Micros Sys., Inc., No. 09-3242, 2012 WL 5986773 (D. Md. Nov. 27, 2012); Hodell-Natco Indus., Inc. v. SAP Am., Inc., No. 1:08-cv-02755, 2010 WL 6765522 (N.D. Ohio Sept. 2, 2010); Shema Kolainu—Hear Our Voices v. ProviderSoft, LLC, 832 F. Supp. 2d 194 (E.D.N.Y. 2010); *In re* All Am. Semiconductor, Inc., 490 B.R. 418 (Bkrtcy. S.D. Fla. 2013). *But see In re* Facebook Inc. IPO Sec. & Derivative Litig., 986 F. Supp. 2d 428, 460–62 (S.D.N.Y. 2013) (denying motion to dismiss negligent design claims involving stock exchange software); Claridge v. RockYou, Inc., 785 F. Supp. 2d 855, 866 (N.D. Cal. 2011) (negligence claims for data breach); Clark Street Wine & Spirits v. Emporos Sys. Corp., 754 F. Supp. 2d 474, 480–82 (E.D.N.Y. 2010) (gross negligence claims for data breach); Sotelo v. DirectRevenue, LLC, 384 F. Supp. 2d 1219, 1235–36 (N.D. Ill. 2005) (negligence claims for damages caused by spyware). These latter cases denying motions to dismiss are very much the exception, not the rule.

9. *See, e.g.*, Getz v. Boeing Co., 654 F.3d 852 (9th Cir. 2011) (government contractor immunity); Johnston v. Multidata Sys. Int'l Corp., 523 F.3d 602 (5th Cir. 2008) (no jurisdiction where injuries occurred in Panama); Motorola Mobility, Inc. v. Myriad France SAS, 850 F. Supp. 2d 878 (N.D. Ill. 2012) (contractual limitation of liability); Pirozzi v. Apple Inc., 913 F. Supp. 2d 840, 851–52 (N.D. Cal. 2012) (no "special relationship" between manufacturer and customer); Rock Creek Lumber Co. v. Valley Mach. Works, Ltd., No. 3:08-cv-0967, 2010 WL 2891535 (M.D. Pa. July 21, 2010) (lack of expert testimony); F. Patrick Hubbard, *"Sophisticated Robots": Balancing Liability, Regulation, and Innovation*, 66 FLA. L. REV. 1803, 1839, 1841–42 (2014) (collecting cases involving factory workers injured by industrial robots, as well as cases involving robotic surgery devices, and finding these claims generally unsuccessful); Chong, *supra* note 6; Scott, *supra* note 6. *But see In re* Fort Totten Metrorail Cases Arising Out of Events of June 22, 2009, 895 F. Supp. 2d 48, 73–76, 84 (D.D.C. 2012) (rejecting government contractor and derivative sovereign immunity defenses).

10. *See, e.g.*, Winters v. Fru-Con Inc., 498 F.3d 734 (7th Cir. 2007); Scott v. White Trucks, 699 F.2d 714 (5th Cir. 1983); West v. Bell Helicopter Textron, Inc., 967 F. Supp. 2d 479 (D.N.H. 2013); Wendorf v. JLG Indus., Inc., 683 F. Supp. 2d 537 (E.D. Mich. 2010); Graves v. CAS Med. Sys., Inc., 735 S.E.2d 650 (S.C. 2012); Bailey v. Disney Worldwide Shared Servs., No. 113072/08 (N.Y. Sup. Ct. Feb. 10, 2012); *see also* Rebecca Wexler, *Life, Liberty, and Trade Secrets: Intellectual Property in the Criminal Justice System*, 70 STAN. L. REV. 1343 (2018) (describing use of trade secret privilege to withhold software source code from discovery process).

11. Settlement is a well-worn tactic used to smooth liability risks for new technologies. *See* Nora Freeman Engstrom, *Sunlight and Settlement Mills*, 86 N.Y.U. L. REV. 805 (2011); Samuel Issacharoff & John Fabian Witt, *The Inevitability of Aggregate Settlement: An Institutional Account of American Tort Law*, 57 VAND. L. REV. 1571 (2004); Nancy Leveson, *Medical Devices: The Therac-25*, *in* SAFEWARE: SYSTEM SAFETY AND COMPUTERS 515 (1995), https://www.bowdoin.edu/~allen/courses/cs260/readings/theraclong.pdf [https://perma.cc/9UD9-

As the software industry ventures from purely cyber systems toward cyber-*physical* systems such as self-driving cars, delivery drones, and networked medical devices,[12] anticipation has been building that the rules for cyber-physical liability will be different.[13] Traditional software does not kill, at least not without opportunity for human intervention.[14] But when code controls physical systems directly, code crashes will cause physical crashes.[15] "Common sense" suggests courts would "revolt" at the idea of "killer bots."[16]

Yet precisely how liability should work for cyber-physical systems has remained in limbo. Part I identifies the three major moves proposed in the

---

3GDJ]; Jaclyn Trop & Ben Protess, *Toyota in Talks on Final Settlements over Car Recalls*, N.Y. TIMES, Feb. 10, 2014, at B2, https://www.nytimes.com/2014/02/10/business/toyota-in-talks-on-final-settlements-over-car-recalls.html [https://perma.cc/54TL-92UE] (reporting settlements in hundreds of sudden-acceleration cases after Toyota won three trials and then lost one); Bernie Woodall, *Uber Avoids Legal Battle with Family of Autonomous Family Victim*, REUTERS, Mar. 28, 2018, https://www.reuters.com/article/us-autos-selfdriving-uber-settlement/uber-avoids-legal-battle-with-family-of-autonomous-vehicle-victim-idUSKBN1H5092 [https://perma.cc/YCT8-E7RH].

12. Cyber-physical systems involve a tight coupling between embedded systems and their physical environment. *See generally* Ayan Banerjee et al., *Ensuring Safety, Security, and Sustainability of Mission-Critical Cyber-Physical Systems*, 100 PROC. IEEE 283, 283 (2012) ("Systems that use the information from the physical environment, and in turn can affect the physical environment during their operation, are called cyber-physical systems (CPSs)"); Edward A. Lee, *Cyber-Physical Systems—Are Computing Foundations Adequate?* (Position Paper for NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Oct. 2006).

13. *See* Kenneth S. Abraham & Robert L. Rabin, *Automated Vehicles and Manufacturer Responsibility for Accidents: A New Legal Regime for a New Era*, 105 VA. L. REV. (forthcoming 2019); Mark A. Geistfeld, *A Roadmap for Autonomous Vehicles: State Tort Liability, Automobile Insurance, and Federal Safety Regulation*, 105 CALIF. L. REV. 1611 (2017); Donald G. Gifford, *Technological Triggers to Tort Revolutions: Steam Locomotives, Autonomous Vehicles, and Accident Compensation*, 11 J. TORT LAW 71 (2018); Andrzej Rapaczynski, *Driverless Cars and the Much Delayed Tort Law Revolution* (Colum. Law and Econ., Working Paper No. 540, 2016). For a broader discussion of the historical salience of physical injury to tort law, see Thomas C. Grey, *Accidental Torts*, 54 VAND. L. REV. 1225 (2001).

14. *See* Rebecca Crootof, *A Meaningful Floor for "Meaningful Human Control"*, 30 TEMP. INT'L & COMP. L.J. 53 (2016); M.C. Elish, *Moral Crumple Zones: Cautionary Tales in Human-Robot Interaction* (Data & Soc'y Research Inst., Working Paper No. 1 V2, 2016).

15. *Compare* West v. Bell Helicopter Textron, Inc., 967 F. Supp. 2d 479 (D.N.H. 2013), *and* Wendorf v. JLG Indus. Inc., 683 F. Supp. 2d 537 (E.D. Mich. 2010), *with* Getz v. Boeing Co., 654 F.3d 852 (9th Cir. 2011), *and* Bailey v. Disney Worldwide Shared Servs., No. 113072/08 (N.Y. Sup. Ct. Feb. 10, 2012).

16. *Cf.* Lawrence Lessig, *Laws that Choke Creativity*, TED (Mar. 2007), https://www.ted.com/talks/larry_lessig_says_the_law_is_strangling_creativity/transcript [https://perma.cc/J5EE-6CQ9] ("Common sense—a rare idea in the law, but here it was, common sense—revolts at the idea" (quoting United States v. Causby, 328 U.S. 256, 261 (1946)); *see also* BENJAMIN WITTES & GABRIELLA BLUM, THE FUTURE OF VIOLENCE: ROBOTS AND GERMS, HACKERS AND DRONES (Lara Heimert ed., 2015); Rebecca Crootof, *The Killer Robots Are Here: Legal and Policy Implications*, 36 CARDOZO L. REV. 1837 (2015).

literature thus far: *consumer protectionism*, *technology protectionism*, and *doctrinal conventionalism*. The first two sound a call-to-arms and gravitate toward an all-or-nothing stance: either all the liability should fall on cyber-physical manufacturers, or none of it should. The appeal of both approaches is their obvious simplicity—yet this simplicity masks brittle assumptions, such as the expected rate of injury or the amount of available funds. A more sustainable approach demands a middle path—a mediating principle capable of distinguishing those cyber-physical injuries that merit remedy from those that do not. Here, the third camp counsels patience, maintaining that the current tort regime for product safety already offers robust balancing principles and that no drastic corrections are needed to accommodate cyber-physical technologies.[17]

Contrary to such reassurances, Part II builds the case that the conventional approaches to software safety will continue to stall even in the cyber-physical context. A fundamental attribute of software—computational complexity—confounds the usual tort calculus refined for ordinary manufactured goods. Unlike mechanical errors, software errors occur in an arbitrary manner that cannot be reasonably prevented via *ex ante* design or *ex post* testing.[18] In the commercial-grade software systems being built today, it is impossible to guarantee complete correctness of code. Yet once discovered, many software errors can be painted as careless or obvious.[19] Given this perplexing antinomy—simultaneous impossibility and triviality—it is no accident that courts have struggled to articulate an appropriate tort liability framework for buggy software. The transition to cyber-*physical* systems does not solve that basic riddle.

This Article argues that as long as software errors remain inevitable, the software liability paradigm must shift from *prevention* to *mitigation*.[20]

---

17. *Cf.* RONALD DWORKIN, LAW'S EMPIRE 124 (1986) (defining "conventionalism" as the idea that "the law of a community includes everything within the implicit extension of [legal] conventions" like legislation and precedent).

18. *See infra* Section III.C.

19. *See infra* note 196; Hubbard, *supra* note 9, at 1854 ("Applying the 'reasonable alternative design' test to software will also present problems because a programming error in the software will constitute a defect that, having been discovered, might be easily fixed by a reprogrammed version of the software."). These easy fixes distinguish software from "unavoidably unsafe" products such as pharmaceutical drugs. *See* RESTATEMENT (SECOND) OF TORTS, § 402A cmt. k (1965); James A. Henderson, Jr. & Aaron D. Twerski, *Drug Design Liability: Farewell to Comment k*, 67 BAYLOR L. REV. 521, 543 (2015) ("However, in the last several decades, the notion that courts have no role to play in reviewing drug designs has fallen into disrepute.").

20. *See* Jane Chong, *We Need Strict Laws If We Want More Secure Software*, NEW REPUBLIC (Oct. 30, 2013), https://newrepublic.com/article/115402/sad-state-software-liability-law-bad-code-part-4

To that end, Part III proposes adapting the doctrine of "crashworthiness" from the automotive context to the software context. The crashworthy doctrine holds that a vehicle manufacturer owes a duty to "use reasonable care in the design and manufacture of a product to minimize injuries to its users and not to subject its users to an unreasonable risk of injury *in the event of a collision or impact*."[21] It originally grew out of the physics concept—popularized by Ralph Nader—of the "second collision."[22] The first collision is the one between the car and another external object such as a tree; the second collision occurs when the momentum of the car stops abruptly but the passengers do not.[23] Those who documented crash sites from the 1920s to the 1960s recorded with numbing frequency victims' eyes impaled on jutting dashboard knobs, necks broken by rigid steering columns, jagged "glass collars" where heads had burst through windshields, severed arms from rollovers, and on and on without legal solutions in sight.[24]

So long as the problem of car crashes was defined solely in terms of prevention, courts remained trapped in a zero-sum tradeoff between safety and usability. If bad drivers were the primary cause of crashes,[25] then there was very little courts could require automakers to do to prevent crashes—short of selling "square" cars that "nobody" wanted.[26] What Nader and his allies accomplished was to redefine the problem by partitioning off the

---

[https://perma.cc/8Q9P-ZZER] (arguing that courts should "conceive[] of software as a product that could be designed to minimize, though not eliminate, security vulnerabilities").

21. Larsen v. Gen. Motors Corp., 391 F.2d 495, 504 (8th Cir. 1968) (emphasis added).

22. RALPH NADER, UNSAFE AT ANY SPEED 81–146 (1965).

23. *See* MICHAEL R. LEMOV, CAR SAFETY WARS: ONE HUNDRED YEARS OF TECHNOLOGY, POLITICS, AND DEATH 50 (2015) (explaining that the "basic physics" of the second collision theory were described by Hippocrates in the fourth century BCE and again by Sir Isaac Newton in 1687).

24. LEMOV, *supra* note 23, at 13–16, 54–55 (describing early publications documenting the "sickening details" of car wrecks); NADER, *supra* note 22, at 93–133; *see also* JOEL W. EASTMAN, STYLING VS. SAFETY: THE AMERICAN AUTOMOBILE INDUSTRY AND THE DEVELOPMENT OF AUTOMOTIVE SAFETY, 1900–1966, at 115–17, 177–89 (1984); JEFFREY O'CONNELL & ARTHUR MYERS, SAFETY LAST: AN INDICTMENT OF THE AUTO INDUSTRY 101–42, 168–90 (1966).

25. *See infra* notes 35, 253.

26. *Compare* O'CONNELL & MYERS, *supra* note 24, at 5 ("Back in 1956 William Mitchell, GM's director of styling, . . . [told] a *Fortune* reporter that completely safe cars would appeal only to 'squares—and there ain't any squares no more.'"), *with id.* at 111 (declaring the argument that "a safe car would be too expensive or so ugly no one would buy it . . . loses its intensity the farther away you get from Detroit"). *See also* JERRY L. MASHAW & DAVID L. HARFST, THE STRUGGLE FOR AUTO SAFETY 104 (1990) (describing a doomed government program that produced Experimental Safety Vehicles that "met or exceeded *all* existing and proposed safety standards issued through mid-1970," but turned out to be so clunky and expensive that it had "virtually no impact on standard setting"); EASTMAN, *supra* note 24, at 184–88, 192–93, 198.

first collision, and inventing a new tort claim focused only on the second collision.[27] By showing statistically that each and every car manufactured is highly likely to crash at some point in its lifetime, car safety advocates persuaded courts to set aside the thorny question of why crashes happen, and ask instead how to minimize harm when the inevitable hits.[28] This reframing helped courts regain their footing, and the crashworthy doctrine soon won uniform consensus across the country.[29]

Today, software code faces the same crossroad. Bugs and vulnerabilities are so rampant across the industry that the question of cybercrashes and cyberattacks is not "whether" but "when."[30] Meanwhile, courts have deferred the issue for decades, leaving consumers to suffer unilaterally the full externalized costs of technical debt.[31] As in the automotive context, rejecting the false idol of "crashproof" code and adopting a new mandate of "crashworthy" code would embolden courts and regulators to weigh in on software safety. Consumers would obtain a coveted cause of action against cyber-physical injuries. At the same time, cyber-physical manufacturers and engineers would also benefit by obtaining clearer guidance on uncertain questions of tort liability.

To be clear, a judicial approach need not exclude other methods of safety governance such as agency regulation, legislative rulemaking, or

---

27. A significant portion of Nader's argument was that cars like the Chevrolet Corsair and Volkswagen Beetle were inherently unstable by design and were likely to fishtail, roll over, and crash at the slightest provocation. NADER, *supra* note 22, at ch.1; CTR. FOR AUTO SAFETY, SMALL-ON SAFETY: THE DESIGNED-IN DANGERS OF THE VOLKSWAGEN ch.2 (Lowell Dodge et al. eds., 1972).

28. *See* Larsen v. Gen. Motors Corp., 391 F.2d 495, 502 (8th Cir. 1968) ("Collisions with or without fault of the user are . . . statistically inevitable.").

29. *See infra* note 243.

30. *See* Jane Chong, *Bad Code: Should Software Makers Pay? (Part 1)*, NEW REPUBLIC (Oct. 3, 2013), https://newrepublic.com/article/114973/bad-code-should-software-makers-pay-part-1 [https://perma.cc/2TKK-B89E]; *cf.* Cahen v. Toyota Motor Corp., 147 F. Supp. 3d 955 (N.D. Cal. 2015). For purposes of this Article, a code "crash" should be understood as any deviation from a software program's expected performance, including but not limited to disruption of service. *See* LAURA L. PULLUM, SOFTWARE FAULT TOLERANCE: TECHNIQUES AND IMPLEMENTATION 4 (2001). A cyberattack is the subset of code crashes involving deliberate intent.

31. Jane Chong, *Why Is Our Cybersecurity So Insecure?*, NEW REPUBLIC (Oct. 11, 2013), https://newrepublic.com/article/115145/us-cybersecurity-why-software-so-insecure [https://perma.cc/E345-Y8TV] (citing Ward Cunningham's concept of "'technical debt' to describe the long-term costs of cutting corners when developing code"); *see also* Jane Chong, *What You Don't Know About Internet Security Will Definitely Hurt You*, NEW REPUBLIC (Oct. 22, 2013), https://newrepublic.com/article/115281/what-you-dont-know-about-internet-security-will-definitely-hurt-you [https://perma.cc/D6V9-THDN] (detailing the market failures of cybersecurity); Chong, *supra* note 6, at 78 ("The idea of levying a Pigovian tax on software, which attempts to correct the cost of insecure code to the market, is not so counterintuitive if we accept that all software contains vulnerabilities and generates negative externalities.").

private self-governance. In fact, the crashworthy code framework can be readily adapted for the regulatory setting. As the history of automotive safety teaches, regulators have often followed the judicial lead in learning how to set safety standards.[32]

Part III concludes with guidance on how to design a cyber-physical system so that its code could be deemed "crashworthy" as a matter of law. The computer science literature has long distinguished "fault prevention" from "fault tolerance," which corresponds neatly to the dichotomy between prevention and mitigation.[33] Accordingly, a doctrine of crashworthy code could assess liability against a cyber-physical manufacturer based on whether it adequately incorporates state-of-the-art techniques in software fault tolerance. By definition, because the crashworthy code doctrine does not aim to prevent all injuries, it does not demand incorporation of techniques that lie beyond the state of the art.

This approach offers three advantages. First, from an engineering standpoint, it reduces the complexity of the problem space by reserving heightened scrutiny for only a small subset of code, while allowing the vast majority of code development practices to continue as is. This strategy also comports with best practices for safety-critical systems, such as including multiple redundancies and anticipating ad hoc failures. Second, from a legal standpoint, judges and juries are better equipped to evaluate the narrower question whether a system's design had reasonable crashworthy measures, rather than whether the entire code base as a whole was reasonably safe. More importantly, severing the pre-crash issues from the post-crash issues offers courts more fine-grained control to resolve these difficult cases. Third, from the consumer standpoint, requiring code to be crashworthy should result in safer system designs, and match consumer expectations for how a cyber-physical system should respond when its code crashes.

I. THE UNBEARABLE LIGHTNESS OF SOFTWARE LIABILITY

Sudden and astounding progress in self-driving car technologies has sparked renewed fervor that cyber-physical software can solve car

32. *See generally* Jerry L. Mashaw & David L. Harfst, *From Command and Control to Collaboration and Deference: The Transformation of Auto Safety Regulation*, 34 YALE J. ON REG. 167 (2017).

33. *See infra* note 298.

accidents simply by removing humans from the loop.[34] This confidence that eliminating "human factors" is the key to elevating society can be traced in part to longstanding beliefs that driver error is the dominant cause of car accidents.[35] It also reflects exuberance about the superior capabilities of computer technologies,[36] as well as a bias among computer engineers that human input is the major obstacle to perfect system design.[37]

---

34. *See generally* Bryant Walker Smith, *Automated Driving and Product Liability*, 2017 MICH. ST. L. REV. 1 (2017); *see also* JAMES M. ANDERSON ET AL., RAND, AUTONOMOUS VEHICLE TECHNOLOGY: A GUIDE FOR POLICYMAKERS 12–16 (2016); David C. Vladeck, *Machines Without Principals: Liability Rules and Artificial Intelligence*, 89 WASH. L. REV. 117, 146 (2014) ("[D]riverless vehicles are likely to be far less hazardous or risky than the products they replace."); Kevin Funkhouser, Note, *Paving the Road Ahead: Autonomous Vehicles, Products Liability, and the Need for a New Approach*, 2013 UTAH L. REV. 437, 451 & n.106 ("If early estimates about autonomous vehicles prove to be even close to accurate, their widespread implementation could lead to one of the greatest safety advances in decades."). A more reliable prediction is that computer errors will be different in kind, rather than that they will be more or less frequent. *See* Kyle Graham, *Of Frightened Horses and Autonomous Vehicles: Tort Law and Its Assimilation of Innovations*, 52 SANTA CLARA L. REV. 1241, 1260 (2012) (positing that early accident cases involving novel devices tend to exhibit a "blam[e] the user" dynamic).

35. *See* MASHAW & HARFST, *supra* note 26, at 29 ("And since the horse was gone, the driver was the only serious candidate for blame. This was simple common sense—so common that virtually every utterance on motor vehicle safety in the critical early years of its social conceptualization focused on the driver as the problem."); *cf.* M.C. Elish & Tim Hwang, *Praise the Machine! Punish the Human!: The Contradictory History of Accountability in Automated Aviation* (Data & Soc'y Research Inst., Working Paper No. 1 V2, 2015). Most commentary cites unquestioningly to traffic statistics attributing more than 90% of accidents to driver error, but there is disagreement as to how much reliance to place on such statistics. *Compare* ANDERSON ET AL., *supra* note 34, at 141 ("Human error causes the vast majority of accidents today."), *with* NADER, *supra* note 22, at 239 ("Investigation stops with the driver in the vast majority of cases because our statutes ascribe all responsibilities to the driver . . . . Accident reporting and statistics also reflect the law's emphasis.").

36. *See, e.g.*, Alexis C. Madrigal, *The Most Important Self-Driving Car Announcement Yet*, ATLANTIC (Mar. 28, 2018), https://www.theatlantic.com/technology/archive/2018/03/the-most-important-self-driving-car-announcement-yet/556712/ [https://perma.cc/J5P3-UQPJ] ("The first million miles took roughly six years. The next million took about a year. The third million took less than eight months. The fourth million took six months. And the fifth million took just under three months. Today, that suggests a rate on the order of 10,000 miles per day. If Waymo hits their marks, they'll be driving at a rate that's *three orders of magnitude* faster in 2020. We're talking about covering each million miles in hours."). *But see* Vijay Kumar, *Irrational Exuberance and the 'FATE' of Technology*, COMM. ACM, Nov. 2018, at 8; Matthew Hutson, *AI Researchers Allege that Machine Learning Is Alchemy*, SCIENCE (May 3, 2018), https://www.sciencemag.org/news/2018/05/ai-researchers-allege-machine-learning-alchemy [https://perma.cc/MGM9-MKLP].

37. *See* Alex Davies, *The Very Human Problem Blocking the Path to Self-Driving Cars*, WIRED (Jan. 1, 2017, 7:00 AM), https://www.wired.com/2017/01/human-problem-blocking-path-self-driving-cars/ [https://perma.cc/C97F-7JKJ]; John Markoff, *Google's Next Phase in Driverless Cars: No Brakes or Steering Wheel*, N.Y. TIMES, May 28, 2014, at B1, https://www.nytimes.com/2014/05/28/technology/googles-next-phase-in-driverless-cars-no-brakes-or-steering-wheel.html [https://perma.cc/U2QL-4PLV]; Kathleen L. Mosier & Linda J. Skitka,

Yet this buoyant optimism is matched by equally heavy pessimism. Just a few high-profile fiascos could "trigger a crisis of confidence."[38] Regulators might shut down projects because of resistance from legacy automakers,[39] and because of public fears—fanned by armchair conceits such as the "trolley problem"[40]—that autonomous software is a recipe for premeditated murder.[41] The reason for pause is not the tragedy of death

---

Human Decision Makers and Automated Decision Aids: Made for Each Other?, *in* AUTOMATION AND HUMAN PERFORMANCE: THEORY AND APPLICATIONS 201 (Raja Parasuraman & Mustapha Mouloua eds., 1996).

38. Joshua Corman & Beau Woods, *Safer at Any Speed: The Roads Ahead for Automotive Cyber Safety Policy*, *in* CYBER INSECURITY, *supra* note 6, at 47; *see also* Daisuke Wakabayashi, *Arizona Orders Uber to Remove Self-Driving Cars from Its Roads*, N.Y. TIMES, Mar. 27, 2018, at B4, https://www.nytimes.com/2018/03/26/technology/arizona-uber-cars.html [https://perma.cc/9ZZB-ZXAH].

39. *See* Vikas Bajaj, *The Distraction of Automated Cars*, N.Y. TIMES, Apr. 1, 2018, at SR8, https://www.nytimes.com/2018/03/31/opinion/distraction-self-driving-cars.html [https://perma.cc/X8Y5-9TG8]; John Lippert et al., *Toyota's Vision of Autonomous Cars Is Not Exactly Driverless*, BLOOMBERG BUSINESSWEEK (Sept. 19, 2018, 4:00 AM), https://www.bloomberg.com/news/features/2018-09-19/toyota-s-vision-of-autonomous-cars-is-not-exactly-driverless (last visited Mar. 17, 2019).

40. *See* Bryant Walker Smith, *Slow Down that Runaway Ethical Trolley*, CTR. FOR INTERNET & SOC'Y (Jan. 12, 2015, 3:42 PM), https://cyberlaw.stanford.edu/blog/2015/01/slow-down-runaway-ethical-trolley [https://perma.cc/GGW8-P5W3] ("Unfortunately, the reality that automated vehicles will eventually kill people has morphed into the illusion that a paramount challenge for or to these vehicles is deciding who precisely to kill in any given crash . . . . Late last year, I was asked the 'who to kill' question more than any other—by journalists, regulators, and academics."); *see also* Ryan Calo, *Is the Law Ready for Driverless Cars?*, COMM. ACM, May 2018, at 34, 35 ("The New Trolley Problem strikes me as a quirky puzzle in search of a dinner party."); Noah Goodall, *Away from Trolley Problems and Toward Risk Management*, 30 J. APPLIED ARTIFICIAL INTELLIGENCE 810 (2016); Rodney Brooks, *Unexpected Consequences of Self Driving Cars*, RODNEY BROOKS: ROBOTS, AI, & OTHER STUFF (Jan. 12, 2017), https://rodneybrooks.com/unexpected-consequences-of-self-driving-cars [https://perma.cc/F7YQ-9REC] (denigrating the trolley problem as "pure mental masturbation dressed up as moral philosophy"); Julian De Freitas et al., *Doubting Driverless Dilemmas*, PSYARXIV (forthcoming 2019), https://psyarxiv.com/a36e5/ [https://perma.cc/EHU7-UJ5R] ("Instead of stoking these flames with distracting thought experiments, we should empower safety engineers to continue improving at the main goal of minimizing harm.").

41. *See, e.g.*, Jean-François Bonnefon et al., *The Social Dilemma of Autonomous Vehicles*, 352 SCIENCE 1573 (2016) (reporting so-called "experimental ethics" results from MIT Media Lab's "Moral Machine" project); Bryant Walker Smith, *The Trolley and the Pinto: Cost-Benefit Analysis in Automated Driving and Other Cyber-Physical Systems*, 4 TEX. A&M L. REV. 197, 198–99 (2017) (collecting media hype). Note that the real technology-in-use relies on continuously updating calculations of collision risk, not split-second judgments of moral preference. *See, e.g.*, Consideration of risks in active sensing for an autonomous vehicle, U.S. Patent No. 8,781,669 (filed May 14, 2012); Controlling vehicle lateral lane positioning, U.S. Patent No. 8,781,670 (filed May 28, 2013).

itself (tens of thousands of Americans die by car every year) but because death by software-car is perceived as "different."[42]

The legal literature thus far has offered only rudimentary guidance on liability for autonomous vehicles.[43] It can be organized broadly into three camps. One perspective is *consumer protectionism*, which seizes on the central fact that end users are no longer in charge when autonomous software takes over control. If human "drivers" are no longer driving, how could it be fair to hold them responsible for car accidents? Surely the fault must fall elsewhere. A counter-response, *technology protectionism*, raises fears that out-of-control liability costs will drive manufacturers out of business. Transformative technologies are considerably more difficult to build than the ordinary consumer could fathom, and liability must be capped if such ventures are to be risked at all; technology companies cannot be used as societal insurers. In its strongest form, the argument is for absolute immunity from tort liability; weaker versions seek merely to cap damages in more limited fashion. The third stance, *doctrinal conventionalism*, cautions against adopting either of the two extremes. Instead, this view holds that existing regimes such as negligence and products liability are flexible and capacious enough to address cyber-physical harms on a case-by-case basis.

A. Consumer Protectionism: A Heavy Hand

The first approach, *consumer protectionism*, draws on one of the richest traditions of twentieth century American legal theory.[44] It takes the general view that the burdens caused by new technologies should not be forced upon hapless victims, but should be borne instead by those best situated to account for those risks.[45] An important battle within this tradition has been over how to apportion costs when the plaintiff shares

---

42. Azim Shariff et al., *Psychological Roadblocks to the Adoption of Self-Driving Vehicles*, NATURE HUM. BEHAV., Sept. 2017, at 694; HILLARY ABRAHAM ET AL., MIT AGELAB, AUTONOMOUS VEHICLES, TRUST AND DRIVING ALTERNATIVES: A SURVEY OF CONSUMER PREFERENCES (2016).

43. *See, e.g.*, DOROTHY GLANCY ET AL., TRANSP. RES. BD., A LOOK AT THE LEGAL ENVIRONMENT FOR DRIVERLESS VEHICLES 79 (2016) (concluding that "forecasts regarding the 'likely' or optimal legal policy responses to driverless vehicles should be made only tentatively").

44. *See* John C.P. Goldberg & Benjamin C. Zipursky, *Torts as Wrongs*, 88 TEX. L. REV. 917, 921 (2010) (discussing OLIVER WENDELL HOLMES, JR., THE COMMON LAW (1881)); John C.P. Goldberg, *Twentieth-Century Tort Theory*, 91 GEO. L.J. 513, 523–24 (2003) (describing modernity's influence on American tort theory).

45. *See generally* Mark Geistfeld, *Negligence, Compensation, and the Coherence of Tort Law*, 91 GEO. L.J. 585 (2003); Mario J. Rizzo, *Law Amid Flux: The Economics of Negligence and Strict Liability in Tort*, 9 J. LEGAL STUD. 291 (1980).

some percentage of fault, e.g., for reckless driving. But when autonomous software takes full control of the wheel, the argument that liability should be shared is significantly weakened.[46] Intuitively, many commentators start from the gut sense that victims of driverless accidents are blameless and should not have to pay for their injuries.[47]

Strict liability has long been the first resort of those seeking to shift the costs of accidents away from victims to other responsible parties.[48] There are two flavors of strict liability: the older vintage is for "ultrahazardous" or "abnormally dangerous" activities such as blasting rocks, keeping vicious animals, or storing toxic chemicals.[49] Some have asked whether lack of human controllability could qualify as abnormally dangerous.[50] However, technological novelty should not be conflated with abnormality. That post-industrial distinction helps explain why this doctrine's scope has become vanishingly thin, and why extending it to consumer-oriented goods and services would be a poor conceptual fit where the intended use is normal, not abnormal.[51] A more helpful heuristic here is whether a given

---

46. *See* Rapaczynski, *supra* note 13. Questions of victim fault do not disappear entirely, however: residual claims might include failure to maintain software updates, intentional tampering by end users, and the child who runs out in front of a moving car.

47. *See, e.g.*, Omri Ben-Shahar, *Should Carmakers Be Liable When a Self-Driving Car Crashes?*, FORBES (Sept. 22, 2016), https://www.forbes.com/sites/omribenshahar/2016/09/22/should-carmakers-be-liable-when-a-self-driving-car-crashes/ [https://perma.cc/7PAT-QKBW].

48. *See* David G. Owen, *Rethinking the Policies of Strict Products Liability*, 33 VAND. L. REV. 681, 703–04 (1980); George L. Priest, *Strict Products Liability: The Original Intent*, 10 CARDOZO L. REV. 2301, 2307 (1989) ("The simple desire of the founders was to ease consumer recovery in cases in which consumers had suffered personal injury from products which obviously had been mismanufactured.").

49. Charles E. Cantú, *Distinguishing the Concept of Strict Liability for Ultra-Hazardous Activities from Strict Products Liability Under Section 402A of the Restatement (Second) of Torts: Two Parallel Lines of Reasoning that Should Never Meet*, 35 AKRON L. REV. 31 (2002). This version of strict liability represents the only remnants that survived the dramatic shift to negligence theory at the end of the nineteenth century. Goldberg & Zipursky, *supra* note 44, at 921–22; *see also* RESTATEMENT (SECOND) OF TORTS, § 520 (1977).

50. *See* Sophia H. Duffy & Jamie Patrick Hopkins, *Sit, Stay, Drive: The Future of Autonomous Car Liability*, 16 SMU SCI. & TECH. L. REV. 453, 459 & n.40 (2013) (citing an early case from 1921 finding an out-of-control car to be a "dangerous instrumentality"); Hubbard, *supra* note 9, at 1862–63 & n.291 (collecting discussion). *But cf.* M. Ryan Calo, *Open Robotics*, 70 MD. L. REV. 571, 610 (2011) ("Despite the inevitability of some injury and damage, there is little reason to assume that personal robots will regularly harm people or property.").

51. *See* Kenneth W. Simons, *The Restatement (Third) of Torts and Traditional Strict Liability: Robust Rationales, Slender Doctrines*, 44 WAKE FOREST L. REV. 1355, 1376 & n.81 (2009) (observing that "very few activities have been found to be abnormally dangerous," not even gas lines, power lines, underground storage of gasoline, or transportation of dangerous chemicals) (citing Gerald W. Boston, *Strict Liability for Abnormally Dangerous Activity: The Negligence Barrier*, 36 SAN DIEGO L. REV. 597, 623–24 (1999)).

cyber-physical system is inappropriate or unfit for the location in which it is deployed.[52] For example, autonomous weapons systems for military use or nanobot swarms for industrial use could be considered abnormally dangerous if deployed in a residential neighborhood.[53] But autonomous taxi fleets intended for urban use almost certainly would not, because that activity is categorically appropriate for the zone.[54]

The more modern version of strict liability, pioneered in the mid-1960s, is for consumer products in "defective condition unreasonably dangerous to the user."[55] After an intense bloom in the 1960s and 1970s, the strict products liability movement has likewise faced stiff cutback in courtrooms and mainstream tort scholarship following the liability insurance crisis of the 1970s and 1980s[56] and the subsequent tort reform movement of the 1990s and 2000s.[57]

Nevertheless, strict products liability has been enjoying a popular revival within the software and robotics literature. The conceptual moves

---

52. *See* Cantú, *supra* note 49, at 35–40; Mark A. Geistfeld, *Should Enterprise Liability Replace the Rule of Strict Liability for Abnormally Dangerous Activities?*, 45 UCLA L. REV. 611, 653–55 (1998); *see also* RESTATEMENT (THIRD) OF TORTS: LIABILITY FOR PHYSICAL HARM ch.4 (AM. LAW INST. 2009); David C. Vladeck, *Machines Without Principals: Liability Rules and Artificial Intelligence*, 89 WASH. L. REV. 117, 146 (2014) (asserting that driverless vehicles cannot be ultrahazardous "for the simple reason that driver-less vehicles are likely to be far less hazardous or risky than the products they replace").

53. *See* WITTES & BLUM, *supra* note 16, at 35–37 (describing research efforts on nanotechnology and nanobots); Henry Fountain & Michael S. Schmidt, *'Bomb Robot' Takes Down Gunman, but Raises Enforcement Questions*, N.Y. TIMES, July 9, 2016, at A15.

54. Even if autonomous cars are not "abnormal," one might well believe they should be held to a supernatural standard of care. This intuition is better explained by the concept of "compliance error" than of "ultrahazardous activity." *See* Mark F. Grady, *Res Ipsa Loquitur and Compliance Error*, 142 U. PA. L. REV. 887, 903, 910 (1994) (observing the "paradox" that "accidents in areas with the most safety equipment are the strongest res ipsa cases" because there are more possibilities for compliance error, which he defines as "an inadvertent failure to use a precaution"). Many thanks to Martha Chamallas for this pointer.

55. RESTATEMENT (SECOND) OF TORTS § 402A (1965).

56. *See* Anita Johnson, *Products Liability "Reform": A Hazard to Consumers*, 56 N.C. L. REV. 677 (1978); Ralph A. Winter, *The Liability Crisis and the Dynamics of Competitive Insurance Markets*, 5 YALE J. ON REG. 455 (1988).

57. *See* John C.P. Goldberg & Benjamin C. Zipursky, *The Easy Case for Products Liability Law: A Response to Professors Polinsky and Shavell*, 123 HARV. L. REV. 1919 (2010); Goldberg, *supra* note 44, at 540 (observing that the strict liability movement "has lost a good deal of its momentum"); James A. Henderson, *Why Negligence Dominates Tort*, 50 UCLA L. REV. 377, 390–97 (2002) (arguing that broad-based strict liability is "not viable" because it is not insurable, due to problems such as adverse selection and moral hazard); Michael L. Rustad & Thomas H. Koenig, *Taming the Tort Monster: The American Civil Justice System as a Battleground of Social Theory*, 68 BROOK. L. REV. 1, 88–93 (2002); Aaron D. Twerski & James A. Henderson, Jr., *Manufacturers' Liability for Defective Product Designs: The Triumph of Risk-Utility*, 74 BROOK. L. REV. 1061 (2009).

are well-established: cyber-physical manufacturers should bear unilateral responsibility because they are the "least cost avoiders" as well as the "best risk spreaders."[58] To the first point, software manufacturers typically maintain tight control over their code and fiercely guard its secrecy.[59] No one else is likely to have better knowledge or ability to certify code quality or to improve code safety than those holding the pen.[60] To the second point, ordinary consumers are less likely to have the financial resources to pursue litigation, whereas well-endowed corporations are better-positioned to "spread the losses" across greater pools of revenue.[61]

And yet there is reason for pessimism: none of those arguments are new, and they have long failed to move any court to extend products liability law to software.[62] Even more damning is the fact that the strict

58. *See, e.g.*, Alan Butler, *Products Liability and the Internet of (Insecure) Things: Should Manufacturers Be Liable for Damage Caused by Hacked Devices?*, 50 U. MICH. J.L. REFORM 913, 916 (2017) ("[H]olding manufacturers liable for downstream harms caused by their insecure devices . . . encourage[es] manufacturers (as a least-cost-avoider) to invest in security measures."); Jeffrey K. Gurney, Note, *Sue My Car Not Me: Products Liability and Accidents Involving Autonomous Vehicles*, 2013 U. ILL. J.L. & TECH. POL'Y 247, 272 (the manufacturer "writes and controls the algorithm for the autonomous technology," so therefore "the easiest method for courts to ensure autonomous vehicle safety would be to hold the manufacturer liable for accidents caused in autonomous mode"); Justin (Gus) Hurwitz, *Cyberensuring Security*, 49 CONN. L. REV. 1495, 1526–29 (2017) ("[P]lacing the risk of cyber-incidents on parties that are better able to mitigate them will likely lead to an overall improvement in the systems that make up the cybersecurity ecosystem, reducing the overall risk for everyone."). *See generally* Jules L. Coleman, *The Morality of Strict Tort Liability*, 18 WM. & MARY L. REV. 259, 262 (1976).

59. *See* Derek E. Bambauer & Oliver Day, *The Hacker's Aegis*, 60 EMORY L.J. 1052 (2011); Harry Surden & Mary-Anne Williams, *Technological Opacity, Predictability, and Self-Driving Cars*, 38 CARDOZO L. REV. 121 (2016); Wexler, *supra* note 10.

60. *See* Bryant Walker Smith, *Proximity-Driven Liability*, 102 GEO. L.J. 1777, 1779, 1794 (2014) (arguing that manufacturer duties should expand with greater proximity to "knowledge about, access to, and control over their products, the people who use them, and the ways in which they are used"); *cf.* NADER, *supra* note 22 (arguing that it is easier to change manufacturer behavior than to change consumer behavior). *But see* Jay P. Kesan & Carol M. Hayes, *Bugs in the Market: Creating a Legitimate, Transparent, and Vendor-Focused Market for Software Vulnerabilities*, 58 ARIZ. L. REV. 753 (2016) (arguing that a free-market system for bug hunting would be an efficient solution).

61. Orly Ravid, Comment, *Don't Sue Me, I Was Just Lawfully Texting & Drunk When My Autonomous Car Crashed Into You*, 44 SW. L. REV. 175, 201 (2014) ("[M]anufacturers should be per se liable for any injury resulting from complete and proper autonomous use."); Jacob D. Walpert, Note, *Carpooling Liability?: Applying Tort Law Principles to the Joint Emergence of Self-Driving Automobiles and Transportation Network Companies*, 85 FORDHAM L. REV. 1863, 1894–95 (2017). Volvo, followed by a few other automakers, proclaimed willingness to accept full liability. Geistfeld, *supra* note 13, at 1629–30 & n.52.

62. *See infra* note 142.

products liability movement has eroded to the point where it is hardly "strict" at all anymore.[63]

No-fault insurance is the new hope for those who acknowledge the shortcomings of strict liability as a doctrinal matter but want to find alternative ways to make it work.[64] Advocates hawk no-fault schemes as a win-win-win: manufacturers can smooth their losses, victims can receive guaranteed payouts, and courts can outsource tricky liability questions to insurers who have expertise at pricing risk.[65] Fringe benefits include that the insurance system is more cost-effective than the tort system,[66] and that the insurance industry will "nudge" manufacturers to improve on a battery of safety metrics.[67]

Other scholars sound a note of caution against invoking insurance as *deus ex machina*, citing problems with prior no-fault schemes.[68] First,

---

63. Hubbard, *supra* note 9, at 1823–26 (noting that only manufacturing defects and distributor liability are truly strict); Twerski & Henderson, *supra* note 57. *But cf.* Douglas A. Kysar, *The Expectations of Consumers*, 103 COLUM. L. REV. 1700 (2003) (drawing on behavioral psychology literature to argue that there may be a role remaining for the stricter "consumer expectations" test).

64. *See* Abraham & Rabin, *supra* note 13 (propounding a "Manufacturer Enterprise Responsibility (MER)" scheme that would be a "manufacturer-financed, strict responsibility bodily-injury compensation system, administered by a Fund created through assessments levied on HAV [highly automated vehicle] manufacturers"); Daniel A. Crane et al., *A Survey of Legal Issues Arising from the Deployment of Autonomous and Connected Vehicles*, 23 MICH. TELECOMM. & TECH. L. REV. 191, 258 (2017) ("In the long run, however, the new products liability risks associated with the shift to ACVs seem entirely insurable, given the size of reinsurance markets and given their ability to handle substantially larger risks."). *But see* Robert D. Cooter, *Economic Theories of Legal Liability*, J. ECON. PERSPECTIVES, Summer 1991, at 11, 26–28 (observing that "liability insurance attempts to provide perfect compensation, while accident insurance covers only risks that victims believe it worthwhile to insure against").

65. Calo, *supra* note 50, at 609–11; Ravid, *supra* note 61, at 203 ("In wanting to avoid the problem of needless complicated and expensive litigation, an insurance solution ought to do the trick."); Vladeck, *supra* note 34, at 147–49; *see also* Hurwitz, *supra* note 58, at 1540–42; Kuwahara, *supra* note 7, at 1010–12.

66. Geistfeld, *supra* note 13, at 1694.

67. *See* Omri Ben-Shahar & Kyle D. Logue, *Outsourcing Regulation: How Insurance Reduces Moral Hazard*, 111 MICH. L. REV. 197 (2012); *see also* MASHAW & HARFST, *supra* note 26, at 209–13 (discussing *Motor Vehicle Mfgrs. Ass'n v. State Farm Mut. Auto. Ins. Co.*, 463 U.S. 29 (1983), and *State Farm Mut. Auto. Ins. Co. v. Dole*, 802 F.2d 474 (D.C. Cir. 1986)). *But see* NADER, *supra* note 22, at 248–57 (summarizing the insurance industry's position as one of calculated indifference: "[t]hey don't want us telling them how to build autos and we don't want them telling us how to sell insurance"); Crane et al., *supra* note 64, at 256–57 ("Of course, over time, as auto products liability insurance premiums increase, those costs will be shifted back to auto makers, who will (again) shift most of those costs back to auto consumers through higher auto prices.").

68. *See* Nora Freeman Engstrom, *An Alternative Explanation for No-Fault's "Demise"*, 61 DEPAUL L. REV. 303 (2012) (attributing the failure of the no-fault movement to evidence that no-fault is associated with increased fatality rates, resistance by the plaintiffs' bar, as well as the rise of auto insurers as primary payers); Efthimios Parasidis, *Recalibrating Vaccination Laws*, 97 B.U. L.

financial constraints call into question the assumption that insurers will always be able and willing to pay.[69] No-fault schemes may be more feasible where claims are rarely needed (as in nuclear energy insurance pools), or where there is a special relationship that reduces moral hazard (as in workers' compensation and victim compensation funds).[70] By contrast, auto injury victims are among the most likely to seek compensation.[71] When claims exceed profit models, insurers reduce payouts, deny claim coverage, or exit the market entirely.[72] Second, political resistance can thwart the enactment of no-fault compensation schemes, or upset the delicate balancing needed to make such systems work as designed.[73] Third, liability insurance is useful for pooling factual risks, but not for resolving uncertainties about the liability standard

---

REV. 2153 (2017); JAMES M. ANDERSON ET AL., RAND, THE U.S. EXPERIENCE WITH NO-FAULT AUTOMOBILE INSURANCE 1 (2010), https://www.rand.org/content/dam/rand/pubs/monographs/2010/RAND_MG860.pdf [https://perma.cc/54S8-DK48]; GLANCY ET AL., *supra* note 43, at 47–61.

69. *See, e.g.*, MASHAW & HARFST, *supra* note 26 (describing insurance crisis of the 1980s); Kenneth S. Abraham, *Making Sense of the Liability Insurance Crisis*, 48 OHIO ST. L.J. 399 (1987); George L. Priest, *The Current Insurance Crisis and Modern Tort Law*, 96 YALE L.J. 1521 (1987); Heather Long, *Where Harvey Is Hitting Hardest, 80 Percent Lack Flood Insurance*, WASH. POST (Aug. 29, 2017), https://www.washingtonpost.com/news/wonk/wp/2017/08/29/where-harvey-is-hitting-hardest-four-out-of-five-homeowners-lack-flood-insurance/ [https://perma.cc/MYJ5-QSXZ] ("Private insurers largely avoid offering flood insurance because it's hard to price the risk and they lose money. The federal program is struggling financially."); Mary Williams Walsh, *Wildfires Move California Closer to Insurance Crisis*, N.Y. TIMES, Nov. 21, 2018, at B1.

70. *See, e.g.*, Mark A. Geistfeld, *supra* note 13 (leaning heavily on the assumption that autonomous vehicles will be substantially safer than conventional vehicles); Gifford, *supra* note 13, at 127 (noting that "many employers promoted the adoption of workers' compensation because of their fear that the fellow-servant rule . . . was about to collapse"). *But see* Gary T. Schwartz, *Waste, Fraud, and Abuse in Workers' Compensation: The Recent California Experience*, 52 MD. L. REV. 983 (1993).

71. *See* Engstrom, *supra* note 5, at 299 (citing DEBORAH HENSLER ET AL., COMPENSATION FOR ACCIDENTAL INJURIES IN THE UNITED STATES 110 (1991)); *cf.* Tom Baker, *Blood Money, New Money, and the Moral Economy of Tort Law in Action*, 35 LAW & SOC'Y REV. 501 (2001) (finding that tort claims are often shaped to match the available insurance coverage).

72. *See* Kenneth S. Abraham, *Tort Luck and Liability Insurance*, 70 RUTGERS U. L. REV. 1, 33 (2017) ("The crisis of the mid-1980s was a shot across the bow of courts and legislatures. These institutions saw for the first time that where tort law went, liability insurance was not always sure to follow. And with that recognition, the expansion of long-tail liability halted."); Nora Freeman Engstrom, *A Dose of Reality for Specialized Courts: Lessons from the VICP*, 163 U. PENN. L. REV. 1631, 1655–58 (2015); Paul Heaton et al., *Victim Compensation Funds and Tort Litigation Following Incidents of Mass Violence*, 63 BUFF. L. REV. 1263 (2015).

73. *See* Engstrom, *supra* note 5, at 312–13 (collecting commentary); Henderson, *supra* note 57, at 383 & nn.36–37 (detailing several failed efforts to expand no-fault insurance, and noting that "'the no-fault movement ground to a halt in 1975'" and that "[n]o state has enacted a no-fault statute since 1975, and several no-fault statutes have been repealed" (citations omitted)); Hubbard, *supra* note 9, at 1859–60.

itself.[74] Professor Mark Geistfeld warns that "when there is a fundamental disagreement about the underlying liability rules, the uncertainty is systemic and cannot be eliminated by the pooling of individual risks within an insurance scheme."[75] Professor Patrick Hubbard also reflects this caution, pointing out that

> [A]ny proposal to impose no-fault liability for accidents caused by fully autonomous cars needs to provide a test for determining *which* accident costs will be imposed on sellers . . . . Simply referring to the manufacturer's ability to spread the cost ignores these tasks as well as the reasons for abandoning cost-spreading as a basis for products liability.[76]

To date, "cyber insurance" policies remain conspicuously limited in their coverage.[77]

Another variation on the theme of consumer protectionism is to bypass the manufacturer and pin liability directly on the autonomous entity

---

74. *See* Robert Martin, *General Aviation Manufacturing: An Industry Under Siege*, *in* THE LIABILITY MAZE: THE IMPACT OF LIABILITY LAW ON SAFETY AND INNOVATION 478, 483–84 (Peter W. Huber & Robert E. Litan eds. 1991) [hereinafter LIABILITY MAZE] ("As one prominent Lloyd's aviation underwriter put it: 'We are quite prepared to insure the risks of aviation, but not the risks of the American legal system.'"). *Compare* Issacharoff & Witt, *supra* note 11 (detailing automobile accident insurance claims from 1920s to 1960s), *with infra* notes 188–90 (denial of Y2K insurance claims). *See generally* MASHAW & HARFST, *supra* note 26, at 216–17 (describing dysfunctions in value-of-life calculations, finding a range of imputed values "from $93,000 to $989,000,000 per life saved" reflecting a "spectacular variance" of a factor of 1,000); Gary T. Schwartz, *Contributory and Comparative Negligence: A Reappraisal*, 87 YALE L.J. 697, 697 n.5 (1978) (observing that auto liability insurers historically have defended harsher rules for tort recovery).

75. Geistfeld, *supra* note 13, at 1618; *see also* John W. Wade, *On the Effect in Product Liability of Knowledge Unavailable Prior to Marketing*, 58 N.Y.U. L. REV. 734, 755 (1983) ("How does one spread the potential loss of an unknowable hazard? How can insurance premiums be figured for this purpose? Indeed, will insurance be available at all? . . . Providing compensation should not be the sole basis for imposing tort liability, and this seems more emphatically so in the situation where the defendant is no more able to insure against unknown risks than is the plaintiff."); *cf.* Kenneth S. Abraham, *Four Conceptions of Insurance*, 161 U. PENN. L. REV. 653 (2013) (cataloging the many interpretive difficulties and market failures that arise within insurance law).

76. Hubbard, *supra* note 9, at 1868–69 (emphasis added).

77. Erik S. Knutsen & Jeffrey W. Stempel, *The Techno-Neutrality Solution to Navigating Insurance Coverage for Cyber Losses*, 122 PENN ST. L. REV. 645 (2018); *see also* GLANCY ET AL., *supra* note 43, at 57 (noting that "insurers are beginning to add cyber exclusions to the policies to avoid any ambiguity with respect to the issue"); Hurwitz, *supra* note 58, at 1536–38 (collecting skeptical commentary and observing that the "cyber-insurance market has, in fact, proceeded along these lines"); Robert Morgus, *Cyber Insurance: A Market-Based Approach to Information Assurance*, *in* CYBER INSECURITY, *supra* note 6, 155, 161–62 (observing that "[m]any cyber insurance policies exclude physical damage," and that "only 50 percent cover the loss associated with restoring systems due to physical damage caused by an incident").

itself.[78] Lured by dreams of sentient robots, these thought experiments often draw analogies to torts committed by animals or children.[79] More pragmatic economic drivers are at work here, too; Professor David Vladeck offers the best example of how these two ideas interrelate.[80] He candidly confesses he favors a strict liability regime for consumer welfare purposes, yet feels uncomfortable "making the manufacturer shoulder the costs alone."[81] Vladeck's compromise is to transfer liability to the robots, in personam, for any injury-producing decisions that cannot be reasonably assigned to their manufacturers.[82] The robot—like the insurer—serves as a pass-through vessel that dissociates the desire of victim compensation from the pain of manufacturer payment. But this invention is an illusion: robots will have owners, so robot liability is respondeat superior by another name.[83]

---

78. *See* EUROPEAN PARLIAMENT RESOLUTION OF 16 FEBRUARY 2017 WITH RECOMMENDATIONS TO THE COMMISSION ON CIVIL LAW RULES ON ROBOTICS § 59(f) (2017) (proposing—unsuccessfully—that "the most sophisticated autonomous robots" should have the status of "electronic persons responsible for making good any damage they may cause"); SAMIR CHOPRA & LAURENCE F. WHITE, A LEGAL THEORY FOR AUTONOMOUS ARTIFICIAL AGENTS 143–44 (Melody Herr ed., 2011); Ryan Abbott, *The Reasonable Computer: Disrupting the Paradigm of Tort Liability*, 86 GEO. WASH. L. REV. 1 (2018); Gary E. Marchant & Rachel A. Lindor, *The Coming Collision Between Autonomous Vehicles and the Liability System*, 52 SANTA CLARA L. REV. 1321, 1326, 1328–29 (2012) ("With a fully autonomous vehicle, however, the responsibility for avoiding an accident shifts entirely to the vehicle."); K.C. Webb, Comment, *Products Liability and Autonomous Vehicles: Who's Driving Whom?*, 23 RICH. J.L. & TECH. 9 (2017); Mark A. Lemley & Bryan Casey, *Remedies for Robots* (Stanford Law and Economics Olin, Working Paper No. 523, 2018), https://ssrn.com/abstract=3223621 [https://perma.cc/M28S-XLE8]; Geistfeld, *supra* note 13, at 1630 ("Lest there be any doubt about the matter, NHTSA has ruled that Google's self-driving car is the equivalent of a human driver for federal regulatory purposes.").

79. *See* F. Patrick Hubbard, *"Do Androids Dream?": Personhood and Intelligent Artifacts*, 83 TEMPLE L. REV. 405 (2011); CHOPRA & WHITE, *supra* note 78, at 11–13; Duffy & Hopkins, *supra* note 50, at 467 (arguing injuries involving autonomous cars should be treated like dog attacks, rather than "dangerous instrumentalities," because "both dogs and autonomous cars think and act independently from their human owners"); Marchant & Lindor, *supra* note 78; *cf.* Christina Mulligan, *Revenge Against Robots*, 69 S.C. L. REV. 579 (2018) (endorsing the psychological "satisfaction" of punishing robots by drawing analogies to animals). *But see* Ryan Calo, *Robotics and the Lessons of Cyberlaw*, 103 CALIF. L. REV. 513, 528–29 & n.108, 539 & n.166 (2015) ("There are analytic and technical reasons to believe robots will never think like people.").

80. Vladeck, *supra* note 34.

81. *Id.* at 146–48.

82. *Id.* at 124 & n.27; *id.* at 150 ("Conferring 'personhood' on these machines would resolve the agency question; the machines become principals in their own right, and along with the new legal status would come new legal burdens, including the burden of self-insurance."). *See generally* Lawrence B. Solum, *Legal Personhood for Artificial Intelligences*, 70 N.C. L. REV. 1231, 1243–48 (1992).

83. Jack M. Balkin, *The Three Laws of Robotics in the Age of Big Data*, 78 OHIO ST. L.J. 1217, 1223–25 (2017) ("I have coined a phrase—the *homunculus fallacy*—to describe . . . the belief that

B. Technology Protectionism: A Light Touch

The second approach, *technology protectionism*, starts from the opposite premise that it is cyber-physical manufacturers who need safeguarding.[84] Alarmed that exposure to mass tort liability might choke innovation, these commentators argue that difficult questions of software liability should continue to be deferred for the greater societal good, citing many prospective benefits such as lives saved, time and money conserved, and toxic emissions reduced.[85]

Professor Ryan Calo has been a prominent proponent of robot exceptionalism. Writing in 2011, Calo proposed a limited form of immunity in which manufacturers would be shielded from tort liability for third-party tinkering, akin to protections for firearms manufacturers and website operators.[86] More recently, he has extended the argument further to encompass "emergent" robot behavior—decisions that are self-learned and self-executed without any direct input from human programmers.[87] While Calo is sensitive to the safety risks that such unsupervised

---

there is a little person inside the program who is making it work—who has good intentions or bad intentions, and who makes the program do good or bad things. But, in fact, there is no little person inside the algorithm . . . . The effects of robotics are always about the relationships of power between human beings or groups of human beings."); *see also* WENDELL WALLACH & COLIN ALLEN, MORAL MACHINES: TEACHING ROBOTS RIGHT FROM WRONG (Martha Ramsey ed., 2009) (autonomous machines are operational rather than functional agents); Calo, *supra* note 79, at 542–43 & nn.184–86; Hubbard, *supra* note 9, at 1862–65 (collecting commentary); *cf.* James Grimmelmann, *There's No Such Thing as a Computer-Authored Work—And It's a Good Thing, Too*, 39 COLUM. J.L. & ARTS 403 (2016); Margot E. Kaminski, *Authorship, Disrupted: AI Authors in Copyright and First Amendment Law*, 51 U.C. DAVIS L. REV. 589 (2017). *But see* Ryan Calo, *Robots as Legal Metaphors*, 30 HARV. J.L. & TECH. 209, 231 (2016) ("Finding a developer, operator, or other person to blame for every action of a robot could be problematic in several ways.").

84. *See* NIDHI KALRA ET AL., RAND, LIABILITY AND REGULATION OF AUTONOMOUS VEHICLE TECHNOLOGIES 46–47 (2009), http://www.dot.ca.gov/newtech/researchreports/reports/2009/prr-2009-28_liability_reg_&_auto_vehicle_final_report_2009.pdf [https://perma.cc/28GZ-DFV5]; Adam Thierer, *When the Trial Lawyers Come for the Robot Cars*, SLATE (June 10, 2016, 7:09 AM), http://www.slate.com/articles/technology/future_tense/2016/06/if_a_driverless_car_crashes_who_is_liable.html [https://perma.cc/ECP2-BUEL].

85. *See, e.g.*, Crane et al., *supra* note 64, at 298–301 (enumerating crash avoidance, increased productivity, decreased congestion, fuel savings, car sharing, increased mobility, and network effects); *id.* at 315 (suggesting possible subsidies for innovation including liability caps, tax credits, and more); Bryant Walker Smith, *supra* note 41; *see also* Calo, *supra* note 50, at 575 ("[T]he potential for crippling legal liability . . . may lead entrepreneurs and investors to abandon open robots in favor of robots with more limited functionality.").

86. Calo, *supra* note 50.

87. Calo, *supra* note 79, at 538–40, 554–55; *cf.* Jack M. Balkin, *The Path of Robotics Law*, 6 CALIF. L. REV. CIRCUIT 45 (2015) (response to Ryan Calo).

autonomy poses,[88] his calculus is that the risks are outweighed by the future societal benefits that robots—and those tinkering with robots—are likely to deliver.[89]

Given the physical risks, Calo and others have suggested that immunity could be paired with a federal agency review model patterned after the Food and Drug Administration (FDA) or the Federal Aviation Administration (FAA)—the gold standard among technology oversight bodies.[90] Ideally, the upfront regulatory cost of this safety review and compliance process would be offset by federal preemption of state tort law on the back end, thereby maximizing consumer safety while minimizing litigation risk.[91] Skeptics of this preclearance model warn, on one side, that a safety review does not guarantee careful review, and on the other side, that a governmental review process can be too onerous and

88. Calo, *supra* note 50, at 603 ("The problem with blanket immunity in the context of robotics is that it would remove not only the legal disincentive to the production of open robots but also an incentive to make them safe."); Calo, *supra* note 79; *see also* Christopher Wing, Note, *Better Keep Your Hands on the Wheel in That Autonomous Car: Examining Society's Need to Navigate the Cybersecurity Roadblocks for Intelligent Vehicles*, 45 HOFSTRA L. REV. 707, 729–31 (2016).

89. Calo, *supra* note 50, at 605. Bryant Walker Smith uses the phrase "newly possible" to express a similar optimism. Smith, *supra* note 41, at 208 n.60. *But see* Hubbard, *supra* note 9, at 1869 (expressing skepticism about the claimed benefits, and criticizing such proposals as "simply ignor[ing] the need to balance innovation with injury costs in a way that incentivizes safety improvements").

90. *See* RYAN CALO, BROOKINGS INST., THE CASE FOR A FEDERAL ROBOTICS COMMISSION (2014); Calo, *supra* note 79, at 555–58; Paul Ohm & Blake Reid, *Regulating Software When Everything Has Software*, 84 GEO. WASH. L. REV. 1672 (2016); Matthew U. Scherer, *Regulating Artificial Intelligence Systems: Risks, Challenges, Competencies, and Strategies*, 29 HARV. J.L. & TECH. 353 (2016); Katherine Booth Wellington, *Cyberattacks on Medical Devices and Hospital Networks: Legal Gaps and Regulatory Solutions*, 30 SANTA CLARA HIGH TECH. L.J. 139, 190–97 (2014); Marchant & Lindor, *supra* note 78, at 1321, 1337–39; *cf.* Woodrow Hartzog, *Unfair and Deceptive Robots*, 74 MD. L. REV. 785 (2015) (embracing the FTC enforcement paradigm for data privacy); Crane et al., *supra* note 64, at 224–27, 240–45 (considering potential regulatory oversight by agencies such as NHTSA, FCC, and FTC, and noting that "calls for the creation of a Federal Robotics Commission" have "yet to gain traction").

91. *See* Catherine M. Sharkey, *Preemption by Preamble: Federal Agencies and the Federalization of Tort Law*, 56 DEPAUL L. REV. 227 (2007); Catherine M. Sharkey, *Products Liability Preemption: An Institutional Approach*, 76 GEO. WASH. L. REV. 449 (2008); Steven Shavell, *Liability for Harm Versus Regulation of Safety*, 13 J. LEGAL STUD. 357, 368–71 (1984); Marchant & Lindor, *supra* note 78, at 1337–39.

cause undesirable delay.[92] Moreover, preemption doctrine is notoriously incoherent and has been a patchy source of manufacturer immunity.[93]

Professor Mark Geistfeld offers a similar vision in which manufacturers could win immunity by disclosing aggregate performance indicators that satisfy a predetermined safety benchmark.[94] This proposal echoes the premarket approval process for pharmaceuticals,[95] but does not depend on the involvement of a federal agency.[96] By tying the liability standard to external test data rather than to internal properties of the code itself, Geistfeld hopes manufacturers will be able to obtain more certainty regarding their liability-risk exposure.

Although calls for manufacturer immunity have been more muted within academic circles, many state legislatures have begun studying the ramifications of enacting such immunities at the behest of manufacturers.[97]

C. Doctrinal Conventionalism: An Invisible Hand

A third approach to the question of liability for autonomous vehicles is *doctrinal conventionalism*, which takes the view that the modern tort regime is sufficiently robust to accommodate any new technology,

---

92. *Compare* Efthimios Parasidis, *Clinical Decision Support: Elements of a Sensible Legal Framework*, 20 J. HEALTH CARE L. & POL'Y 183 (2018) (worrying about insufficient oversight), *with* Elizabeth C. Price, *Teaching the Elephant to Dance: Privatizing the FDA Review Process*, 51 FOOD & DRUG. L.J. 651 (1996) (complaining of regulatory bloat) *and* Lars Noah, *The Little Agency That Could (Act with Indifference to Constitutional and Statutory Strictures)*, 93 CORNELL L. REV. 901 (2008) (criticizing agency overreach).

93. *See* Catherine M. Sharkey, *The Administrative State and the Common Law: Regulatory Substitutes or Complements?*, 65 EMORY L.J. 1705, 1724–33 (2016) (mapping significant rifts within preemption doctrine); Engstrom, *supra* note 72; Hubbard, *supra* note 9, at 1859–60, 1866–67, 1871–72.

94. Geistfeld, *supra* note 13, at 1651 (arguing that a fully autonomous vehicle will "necessarily drive in a reasonably safe manner if prior driving experience shows that [it] at least halves the incidence of crashes relative to conventional vehicles").

95. *See* Richard Nagareda, *FDA Preemption: When Tort Law Meets the Administrative State*, 1 J. TORT L. [ii] (2006) (proposing that FDA preemption should be tied to adequate information disclosures by pharmaceutical and medical device manufacturers).

96. Geistfeld adds, however, that federal standards could reinforce this approach via the regulatory compliance defense. Geistfeld, *supra* note 13, at 1685–88.

97. *See, e.g.*, S.B. 220, 132d Gen. Assemb., Reg. Sess. (Ohio 2018) (enacting affirmative defense for data breach lawsuits); S.B. 998, 98th Leg., Reg. Sess. (Mich. 2016) (exempting mechanics from liability for repairs to automated motor vehicles); S.B. 663, 97th Leg., Reg. Sess. (Mich. 2013) (limitations on manufacturer liability for third-party modifications made to automated motor vehicles). *See generally Autonomous Vehicles: Self-Driving Vehicles Enacted Legislation*, NAT'L CONF. ST. LEGISLATURES (Nov. 7, 2018), http://www.ncsl.org/research/transportation/autonomous-vehicles-self-driving-vehicles-enacted-legislation.aspx [https://perma.cc/VC44-ZLP5].

without special provisions such as strict liability or immunity.[98] Typically, the doctrinal analysis focuses on two dominant theories of tort law: negligence and strict products liability.[99] Both are defined in terms of "reasonableness," though the target differs: negligence measures the reasonable care of one's *conduct*, while products liability measures the reasonable safety of one's *product*.[100] While there are some differences at the margins, there exists broad agreement among tort scholars that the two theories have largely converged in recent decades, especially with respect to product design where conduct and product merge.[101]

Professor Patrick Hubbard offers one of the earliest, and strongest, forms of this genre.[102] After stepping methodically through a comprehensive tour of blackletter law, Hubbard concludes there is absolutely no justification to "abandon[] a system that has provided, and will continue to provide, a fair and efficient balance of innovation and safety in robotic machines."[103] The term "reasonable" is a deliberately flexible concept that can readily accommodate any evolutions in technology to maintain "an efficient balance between the concern for physical safety and the desire for innovation."[104] In his view, proposals for "fundamental change" on either side—no-fault schemes to help

98. *See* Hubbard, *supra* note 9; *see also* GLANCY ET AL., *supra* note 43, at 35–41; Andrew P. Garza, Note, *"Look Ma, No Hands!": Wrinkles and Wrecks in the Age of Autonomous Vehicles*, 46 NEW ENG. L. REV. 581, 583 (2012) ("[P]roducts liability law is capable of handling autonomous vehicles in the same way that it handled new safety technologies in the past."); Alexander Herd, Note, *R2Dford: Autonomous Vehicles and the Legal Implications of Varying Liability Structures*, 5 FAULKNER L. REV. 29 (2013); Jeremy Levy, *No Need to Reinvent the Wheel: Why Existing Liability Law Does Not Need to Be Preemptively Altered to Cope with the Debut of the Driverless Car*, 9 J. BUS., ENTREPRENEURSHIP & L. 355 (2015).

99. GLANCY ET AL., *supra* note 43, at 31–35; Crane et al., *supra* note 64, at 259–61; Scott, *supra* note 6, at 441–50, 467–70.

100. For a detailed discussion of strict products liability as applied to autonomous vehicles, see Geistfeld, *supra* note 13, at 1632–47.

101. *See* David G. Owen, *Design Defect Ghosts*, 74 BROOK. L. REV. 927, 931 (2009) (noting the "open secret" that "while purporting to apply 'strict' liability doctrine to design cases, courts in fact were applying principles that look remarkably like negligence"); Twerski & Henderson, *Triumph of Risk-Utility*, *supra* note 61. *But cf.* Richard C. Ausness, *Product Liability's Parallel Universe: Fault-Based Liability Theories and Modern Products Liability Law*, 74 BROOK. L. REV. 635, 635 (2009) (observing that "plaintiffs now commonly supplement or even replace strict liability with claims that rely on fault-based liability theories").

102. Hubbard, *supra* note 9.

103. *Id.* at 1872.

104. *Id.* at 1861, 1865.

plaintiffs, or limitations on liability to help defendants—are solutions in search of a problem.[105]

A softer version is offered by Professor Dorothy Glancy and her co-authors in their comprehensive report for the Transportation Research Board.[106] They predict that litigation over autonomous vehicle safety will progress in three broad phases: (1) a first stage that adheres closely to claims that are presently successful in conventional automobile litigation; (2) a second stage that witnesses the evolution of more sophisticated legal claims to match developing societal expectations for software performance; and (3) a third stage where claim resolution becomes routinized as the litigation landscape matures.[107] This prediction is modeled closely on a study of the historical arc of early automotive litigation,[108] with the main implication being that tort law will readily absorb cyber-physical technologies despite some false starts. While the report remains carefully agnostic as to the precise content of those future claims, it concludes that there likely will be some blend of strict liability theories and negligence principles, along with regulatory rulemaking.[109]

These critiques are well-taken. Yet, the salient question is not whether but *how* these general tort principles can be adapted to work for cyber-physical systems. To answer that question, one must first articulate a theory of why courts have had great difficulty assessing liability in the software context. This investigation is the subject of the next Part.

II. THE UNREASONABLENESS OF CRASHPROOF CODE

The main puzzle of software liability law has been its curious absence. Two hypotheses have dominated the literature across prior decades. The first is a definitional claim that software does not fit neatly within the four corners of tort law. After all, software blurs the traditional line between intangible information and tangible object,[110] and tort law has long

105. *Id.* at 1865–66 ("Liability law is designed to achieve an efficient balance between the concern for physical safety and the desire for innovation . . . . [T]he persons proposing change simply assume, with little or no argument, that there is a problem that needs to be addressed in a particular way.").

106. GLANCY ET AL., *supra* note 43, at 30–41.

107. *Id.* at 37–40.

108. *See id.* at 35 (citing Graham, *supra* note 34).

109. *Id.* at 40–41.

110. *See* Michael J. Madison, *Law as Design: Objects, Concepts, and Digital Things*, 56 CASE W. RES. L. REV. 381, 414 (2005) (critiquing as "barely coherent" the "search for the essential thingness of the computer program," which has led the law to conclude that "computer programs, by their nature, are simultaneously both intangible things and tangible things"). The informational versus

puzzled over how to administer claims for invisible injuries.[111] Other mechanisms such as contract law, it is argued, are better suited to redress such claims. The second set of objections emerges from the modern law-and-economics movement as well as from innovation incentive theory. Here, the argument is less about form than function—namely that software is a socially beneficial industry that must be shielded from being snuffed out by excessive legal costs.[112]

At first glance, both of these traditional explanations appear to fall away in the cyber-physical context, where the injuries are very much tangible, and the work is being led by mature, well-funded companies. For one, an autonomous vehicle or medical implant that kills its user commits the requisite physical harm that falls squarely within tort's empire. For the

---

functional character of software has long attracted academic interest in the free speech domain, including a recent flurry of renewed attention on so-called "algorithmic speech." *See* Kaminski, *supra* note 83, at 607–08 (collecting discussion). However, when put to the test, courts have been circumspect about extending speech protections to functional aspects of code. *See, e.g.*, Universal City Studios, Inc. v. Corley, 273 F.3d 429, 452 (2d Cir. 2001) ("The functionality of computer code properly affects the scope of its First Amendment protection."); *cf.* Commonwealth v. Carter, No. SJC-12502 (Mass. Feb. 6, 2019) ("It has never been deemed an abridgment of freedom of speech . . . to make a course of conduct illegal merely because the conduct was in part initiated, evidenced, or carried out by means of language, either spoken, written, or printed." (citation omitted)). *But see* Bernstein v. U.S. Dep't of Justice, 176 F.3d 1132, 1142 (9th Cir. 1999) (rejecting the argument that "even one drop of 'direct functionality' overwhelms any constitutional protections that expression might otherwise enjoy").

111. *See generally* David B. Gaebler, *Negligence, Economic Loss, and the U.C.C.*, 61 IND. L.J. 593, 594–96 (1985) ("[A] purchaser of a defective product who has suffered economic injury but no personal injury or property damage may recover only if he can establish fraud, misrepresentation, or a breach of warranty"); Mark Geistfeld, *Placing a Price on Pain and Suffering: A Method for Helping Juries Determine Tort Damages for Nonmonetary Injuries*, 83 CALIF. L. REV. 733 (1995); Daniel Givelber, *The Right to Minimum Social Decency and the Limits of Evenhandedness: Intentional Infliction of Emotional Distress by Outrageous Conduct*, 82 COLUM. L. REV. 42 (1982); John C.P. Goldberg & Benjamin Zipursky, *Unrealized Torts*, 88 VA. L. REV. 1625 (2002); Betsey J. Grey, *The Future of Emotional Harm*, 83 FORDHAM L. REV. 2605 (2015); Nancy Levit, *Ethereal Torts*, 61 GEO. WASH. L. REV. 136 (1992); Robert L. Rabin, *Pain and Suffering and Beyond: Some Thoughts on Recovery for Intangible Loss*, 55 DEPAUL L. REV. 359 (2006); Daniel J. Solove, *A Taxonomy of Privacy*, 154 U. PA. L. REV. 477 (2006).

112. *See, e.g.*, Bruce Schneier, *Liabilities and Software Vulnerabilities*, SCHNEIER ON SECURITY (Oct. 20, 2005, 5:19 AM), https://www.schneier.com/blog/archives/2005/10/liabilities_and.html [https://perma.cc/WYQ4-X2SL] (backpedaling after immediate backlash against the slightest suggestion that software developers and vendors be liable for vulnerabilities in their code); JONATHAN ZITTRAIN, THE FUTURE OF THE INTERNET—AND HOW TO STOP IT 162–63 (2006) (placing legal blame on each product maker "would serve only to propel PC lockdown, reducing generativity"); Elizabeth Pollman, *The Rise of Regulatory Affairs in Innovative Startups*, *in* THE HANDBOOK OF LAW AND ENTREPRENEURSHIP IN THE UNITED STATES (D. Gordon Smith & Christine Hurt eds., forthcoming 2019); Eric Goldman, *Ten Worst Section 230 Rulings of 2016 (Plus the Five Best)*, TECH. & MARKETING L. BLOG (Jan. 4, 2017), https://blog.ericgoldman.org/archives/2017/01/ten-worst-section-230-rulings-of-2016-plus-the-five-best.htm [https://perma.cc/FDZ6-HK6A].

other, market dynamics have shifted dramatically such that software companies dominate the top spots for market valuation.[113] The potential prize for winning the cyber-physical race is expected to be astronomical.

Yet both narratives point to a more basic obstacle that remains unchanged: software complexity.[114] In computer science terms, "complexity" refers not to the magnificence of a program's output, but to the computational impossibility of verifying and validating the correctness of the internal logic of the program.[115] Software complexity grows at an exponential rate, meaning that as the program size increases at a linear rate, the amount of computation needed to prove its correctness grows asymptotically toward infinity. While testing can locate some errors on a piecemeal basis, it cannot comb the entire universe of possible settings (or "machine-states") that the software might encounter in the wild. As a result, readily fixable errors—even embarrassingly trivial ones—regularly pass unnoticed, simply because software testing does not have the capacity to check every corner.

In short, the challenge of software liability is that it is seemingly impossible to identify marginal-cost measures that can or should be taken to improve software safety. The amount of additional effort expended on software certification does not correlate meaningfully with a reduction in risk of software error. If all software errors look alike to jurists, then cracking open the lid to software liability could turn every bug into a

---

113. *See* Shira Ovide & Rani Molla, *Technology Conquers Stock Market*, BLOOMBERG (Aug. 2, 2016), https://www.bloomberg.com/gadfly/articles/2016-08-02/tech-giants-form-fab-five-to-dominate-stock-valuation-chart [https://perma.cc/2UPE-83CX] (listing Apple, Google, Microsoft, Amazon, and Facebook as "the five biggest companies in the world by market value").

114. *See* Chong, *supra* note 6, at 76 ("As Fred Brooks pointed out in his famous 1986 paper distinguishing between essential and accidental complexity, some complexity can be eliminated by way of code optimization. But in other respects complexity is an 'essential property' of software that comes with unavoidable technical and management difficulties and leads to product flaws."); Chong, *supra* note 31 ("Gary McGraw, among the best-known authorities in the field, attributes software's growing security problems to what he terms the 'trinity of trouble': connectivity, extensibility and complexity. To this list, let's add a fourth commonly-cited concern, that of software 'monoculture.'").

115. *See* Robert L. Glass, *Sorting Out Software Complexity*, COMM. ACM, Nov. 2002, at 19; Tom Mens, *On the Complexity of Software Systems*, IEEE COMPUTER SOC'Y, Aug. 2012, at 79; Daniel L. Dvorak, *NASA Study on Flight Software Complexity*, AM. INST. AERONAUTICS & ASTRONAUTICS, 2009, at 35 ("The mathematics that so well describe physics and so well support other engineering disciplines do not apply to the discrete logic that comprises so much of flight software . . . . [P]hysics deals with terribly complex objects, but the physicist labors on in a firm faith that there are unifying principles to be found. However, no such faith comforts the software engineer."). Software "complexity," as used as a term of art by computer scientists, also differs from legal complexity, as used for example by Hubbard, *supra* note 9, at 1851–53.

potential multimillion-dollar lawsuit.[116] To be sure, intangibility and innovation are important clues to understanding the absence of software liability, but they are not its root cause. A successful approach to software liability must first relinquish the canonical assumption that reasonableness can be judged on an individual bug-by-bug basis.

A. Software's Intangible Form

One prevalent explanation for the absence of software liability is software's intangibility. This mistaken belief traces back to the earliest software liability cases, which were brought primarily as breach-of-warranty claims, not as pure tort claims.[117] There, the predominant question was whether software was a "good" or a "service."[118] This dichotomy mattered because Article 2 of the Uniform Commercial Code (UCC)—and its attendant warranties of merchantability and fitness for a particular purpose—applied only to sales of goods.[119] Software's intangibility raised novel and interesting questions about the proper reach of sales law. The fact that intangibility was summarily rejected as a distinguishing feature has largely escaped mention within tort law.

---

116. *See* Grady, *supra* note 54, at 901–02, 905–06 (describing a phenomenon in negligence law where, in cases where "efficient and inefficient lapses are indistinguishable," "[c]ourts seem to require perfect compliance from most defendants" and do not allow a defense of innocent mistake, even though perfect compliance is impossible in the real world).

117. *See* Frances E. Zollers et al., *No More Soft Landings for Software: Liability for Defects in an Industry that Has Come of Age*, 21 SANTA CLARA COMPUTER & HIGH TECH. L.J. 745, 764 (2005) (citing cases); James J. White, *Reverberations from the Collision of Tort and Warranty*, 53 S.C. L. REV. 1067, 1070, 1077 (2002) ("Clearly, Prosser saw the relation between strict tort and warranty.").

118. *See* Amelia H. Boss & William J. Woodward Jr., *Scope of the Uniform Commercial Code; Survey of Computer Contracting Cases*, 43 BUS. LAW. 1513, 1526 (1987) ("The major issue courts have faced in cases involving actions for breach of computer software contracts has been whether such contracts are for sales of goods or services."); Lawrence B. Levy & Suzanne Y. Bell, *Software Product Liability: Understanding and Minimizing the Risks*, 5 HIGH TECH L.J. 1 (1989); Andrew Rodau, *Computer Software: Does Article 2 of the Uniform Commercial Code Apply?*, 35 EMORY L.J. 853 (1986); Scott, *supra* note 6, at 434–41; Bonna Lynn Horovitz, Note, *Computer Software as a Good Under the Uniform Commercial Code: Taking a Byte Out of the Intangibility Myth*, 65 B.U. L. REV. 129 (1985); Kerry M.L. Smith, Comment, *Suing the Provider of Computer Software: How Courts Are Applying U.C.C. Article Two, Strict Tort Liability, and Professional Malpractice*, 24 WILLAMETTE L. REV. 743 (1988).

119. U.C.C. §§ 2-314, 2-315 (AM. LAW INST. & UNIF. LAW COMM'N 2018); Horovitz, *supra* note 118, at 141. *See generally* Peter Alces, *W(h)ither Warranty: The B(l)oom of Products Liability Theory in Cases of Deficient Software Design*, 87 CALIF. L. REV. 269, 276–79 (1999). Sales of services were not *excluded* so much as they were not considered significant enough at the time to be *included*. *See* Raymond T. Nimmer, *Services Contracts: The Forgotten Sector of Commercial Law*, 26 LOY. L.A. L. REV. 725, 727 (1993) (citing Grant Gilmore, *On the Difficulties of Codifying Commercial Law*, 57 YALE L.J. 1341 (1948)).

Warranty law was the natural first resort because those early cases involved bespoke, arms-length transactions between equally sophisticated business entities.[120] In those formative decades, the computer industry was dominated by IBM and a lesser handful of mainframe manufacturers.[121] Hardware was very expensive to manufacture, so sales efforts were limited to high-end business clients who could afford to pay.[122] Personal computers had not yet been invented; nor was there a mass market for standalone software.[123] Instead, software was custom-built by the mainframe manufacturers and bundled in at no additional cost, as a way to entice customers to buy the hardware.[124] Computer salesmen promised to deliver "turnkey systems" that would perform miracles at the metaphorical turn of a key.[125] When expectations were disappointed and deadlines repeatedly missed, frustrated customers sued for breach of contract and warranty.[126]

A "good" was defined by its "movability," so software presented a provocative doctrinal test.[127] On one hand, early software resembled a

---

120. *See* Richard A. Mann & Barry S. Roberts, *The Applicability of Tort Law to Commercial Buyers*, 79 NEB. L. REV. 215, 248–49 (2000) ("When two parties are on roughly equal footing, they are in a position to determine which risks to assume and how costs will be allocated."); Zollers et al., *supra* note 117, at 764 & n.117.

121. PAUL E. CERUZZI, A HISTORY OF MODERN COMPUTING 51–53, 67–69, 143–45 (MIT ed. 1998); MARK A. LEMLEY ET AL., SOFTWARE AND INTERNET LAW 2–9 (1st ed. 2000), https://www.law.berkeley.edu/files/chp1.pdf [https://perma.cc/GMM5-LH2R].

122. CERUZZI, *supra* note 121, at 82 fig.3.1.

123. FRANKLIN M. FISHER, JAMES W. MCKIE & RICHARD B. MANCKE, IBM AND THE U.S. DATA PROCESSING INDUSTRY: AN ECONOMIC HISTORY 322 (1983). *See generally* CERUZZI, *supra* note 121, at 207–41 (describing efforts from 1972 to 1977 leading to the development of the personal computer).

124. CERUZZI, *supra* note 121, at 143–44 (explaining that "most computer dollars continued to be spent on large mainframes" until the advent of personal computers in the 1980s, and that "[t]hose who wished to compete in [the mainframe] business provided everything from bottom to top—hardware, peripherals, system and applications software, and service"); Rodau, *supra* note 118, at 871–73; Horovitz, *supra* note 118, at 153.

125. *See* Diversified Graphics, Ltd. v. Groves, 868 F.2d 293, 297 (8th Cir. 1989) ("The term 'turnkey' is intended to describe a self-sufficient system which the purchaser need only 'turn the key' to commence operation."); USM Corp. v. Arthur D. Little Sys., Inc., 546 N.E.2d 888, 893–94 n.9 (Mass. App. Ct. 1989) (explaining industry usage of the term "turnkey" to mean a system "able to be turned on and function immediately").

126. *See, e.g.*, Chatlos Sys. v. Nat'l Cash Register Corp., 635 F.2d 1081 (3d Cir. 1980); Triangle Underwriters, Inc. v. Honeywell, Inc., 604 F.2d 737 (2d Cir. 1979); IBM Corp. v. Catamore Enters., Inc., 548 F.2d 1065 (1st Cir. 1976); Clements Auto Co. v. Serv. Bureau Corp., 444 F.2d 169 (8th Cir. 1971); Computer Servicenters, Inc. v. Beacon Mfg. Co., 443 F.2d 906 (4th Cir. 1971); *see also* Zollers, *supra* note 117, at 764 n.116 (collecting early cases).

127. U.C.C. § 2-105 (AM. LAW INST. & UNIF. LAW COMM'N 2018) (defining "goods" as "all things (including specially manufactured goods) which are movable at the time of identification to the

"good" in that it was intimately tied to physical hardware as part of an integrated turnkey system; if the hardware was physically movable, then so too was the bundled software.[128] On the other hand, early mainframe software also resembled a "service" in that it was customized for each customer; arguably, what customers were buying was the labor and knowhow of expert computer consultants, not a finished piece of merchandise.[129]

For all the hype, judicial consensus came quickly and quietly.[130] The near-automatic presumption today is that software is a "good" subject to the UCC, with or without the entanglement of a physical machine.[131] To the extent that software transactions include service aspects—such as ongoing maintenance and support—courts have readily ruled that those hybrid features may be folded in as incidental to the sale of the software-as-good.[132] With the invention of the personal PC and rise of the consumer software market, physical props such as boxes, disks, and dongles further cemented the illusion of software as a movable good.[133]

---

contract for sale"); *see also* Alces, *supra* note 119, at 294 ("To be sure, intellectual property, and particularly software, is neither completely goods nor completely services. Software is a hybrid, owing incidents to both the tangible and less tangible.").

128. *See* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2372 (1994); Scott, *supra* note 6, at 434–36.

129. *See* Susan Nycum, *Liability for Malfunction of a Computer Program*, 7 RUTGERS J. COMPUTERS TECH. & L. 1, 3 (1979) (noting that it was unclear "the extent to which the programmer must custom-design the program for the customer's use under the customer's direction and control and the extent to which the program comes as part of a package with . . . hardware").

130. Zollers et al., *supra* note 117, at 766 ("Commentators have addressed in depth the issue of whether software is a good and whether a license is a sale, but the courts spend very little time, if any, debating the concepts.").

131. Michael C. Gemignani, *Product Liability and Software*, 8 RUTGERS COMPUTER & TECH. L.J. 173, 177 n.18 (1981) ("Most courts which have considered the matter seem willing to classify computer programs as goods rather than services." (citation omitted)); David A. Owen, *The Application of Article 2 of the Uniform Commercial Code to Computer Contracts*, 14 N. KY. L. REV. 277, 282 (1987); Jeffrey B. Ritter, *Scope of the Uniform Commercial Code: Computer Contracting Cases and Electronic Commercial Practices*, 45 BUS. LAW. 2533, 2543 (1990) ("In the context of article 2, recent decisions have generally classified software as 'goods'"); Rodau, *supra* note 118, at 883 ("The weight of authority treats computer software as being within the article 2 definition of a good without lengthy analysis or discussion.").

132. *See* Rodau, *supra* note 118, at 913–16; Scott, *supra* note 6, at 434–36. *But see* Stacy-Ann Elvy, *Hybrid Transactions and the INTERNET of Things: Goods, Services, or Software?*, 74 WASH. & LEE L. REV. 77, 109 (2017) (criticizing the "predominant purpose" test as leading to "ambiguous or conflicting results"); *cf.* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 19, Reporters' Notes to cmt. d (AM. LAW INST.1998) ("Under the Code, software that is mass-marketed is considered a good. However, software that was developed specifically for the customer is a service.").

133. *See* Mark A. Lemley, *Intellectual Property and Shrinkwrap Licenses*, 68 S. CAL. L. REV. 1239, 1244–45 n.23 (1995) ("[A]lmost all courts and commentators that have considered the issue have

This battle was important for business-to-business dealings, but for ordinary consumers who lack power to negotiate terms of sale, it was more sound than fury.[134] Boilerplate software licenses rapidly evolved to include sweeping disclaimers of warranties and limitations of liability, and those clauses have been upheld consistently in court.[135] Today, software warranty disputes appear mainly in the form of claims sounding in fraud or misrepresentation, which cannot be disclaimed.[136]

Given the intimate relationship between tort law and warranty law, many commentators expected tort law to pick up where warranty law stopped, as it had done in the past.[137] Here, the parallel issue was whether software is a "product" or a "service."[138] Logic seemed to dictate that if

---

concluded that a shrinkwrap license transaction is a sale of goods . . . covered by Article 2 of the current U.C.C."). *But cf.* Peter A. Alces & Aaron S. Book, *When Y2K Causes "Economic Loss" to "Other Property"*, 84 MINN. L. REV. 1, 24–26 (1999) (clinging to the old view: "Software, either system or application, would not fall within the scope of either the Article 2 or 2A warranty provisions because software is intangible. The fact that it is captured in a tangible form should not be dispositive." (citation omitted)).

134. *See* Rustad & Koenig, *supra* note 6, at 1562–66. Some commentators expressed hope that implied warranties would lead to needed improvements in software quality. Horovitz, *supra* note 118, at 160 ("[T]he UCC will adequately protect software vendees and will not serve as a vehicle for manufacturers to limit their liability."); Smith, *supra* note 118, at 755 ("[B]ecause fairness and reasonableness are fundamental in the Code, application of the U.C.C. would benefit parties unfamiliar with its provisions."); *see also* Gemignani, *supra* note 131, at 178 (describing the early court split on whether to uphold vendor protective clauses in computer contracts). More cynical voices pointed out that a warranty-based approach was no guarantee of accountability. Nycum, *supra* note 129, at 7–8; *see also* Scott, *supra* note 6, at 436–39.

135. Boss & Woodward, Jr., *supra* note 118, at 1540 ("Disclaimers [in software contracts] are generally effective and courts interpret them in the same way as in other contracts."); Lemley, *supra* note 133; Douglas E. Phillips, *When Software Fails: Emerging Standards of Vendor Liability Under the Uniform Commercial Code*, 50 BUS. LAW. 151, 169 (1994). *But see* Zollers et al., *supra* note 117, at 765 (collecting cases allowing recovery despite limitation provision).

136. *See* Methodist Hosps., Inc. v. FTI Cambio, LLC, No. 2:11-cv-036, 2011 WL 2610476 (N.D. Ind. July 1, 2011); Boss & Woodward, Jr., *supra* note 118, at 1533–40 ("[B]uyers often ignore the generally broad disclaimers of express and implied warranties in standard vendor contracts. When they become disappointed and discover that disclaimers foreclose their contract remedies, they turn to the law of misrepresentation for relief."); Zollers et al., *supra* note 117, at 758 (collecting older cases in which misrepresentation claims were successfully used to get around contractual limitations on liability).

137. Alces, *supra* note 119, at 91 ("[W]e must remain aware that it was warranty law's limitations that engendered development of strict products liability."); *see also* Henningsen v. Bloomfield Motors, Inc., 161 A.2d 69 (N.J. 1960), *cited in* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 18 (AM. LAW INST. 1998); Kyle Graham, *Strict Products Liability at 50: Four Histories*, 98 MARQ. L. REV. 555 (2014); William L. Prosser, *The Assault Upon the Citadel (Strict Liability to the Consumer)*, 69 YALE L.J. 1099, 1124–34 (1960) (criticizing courts' contorted reliance on breach of warranty claims to achieve outcomes that sound more properly in tort principles).

138. *See* Roy N. Freed, *Products Liability in the Computer Age*, 17 JURIMETRICS J. 270, 275–79 (1977); Daniel B. Garrie, *The Legal Status of Software*, 23 J. MARSHALL J. COMPUTER & INFO. L.

software were a "good" under warranty law, it should be a "product" under tort law, too.[139] To be sure, the labels are not identical—"products" are defined by "tangibility" while "goods" are defined by "movability"—but as long as software is not a "service," the distinction seems vanishingly small.[140] Besides, commentators were quick to point out that courts have held other intangibles, such as electricity and aeronautical chart data, to be "products."[141]

Instead, courts have studiously avoided answering whether software is a "product,"[142] and have dismissed most software liability claims by

---

711, 714–20 (2005); Gemignani, *supra* note 131, at 197–98; Diane B. Lawrence, *Strict Liability Computer Software and Medicine: Public Policy at the Crossroads*, 23 TORT INS. L.J. 1, 12–15 (1987); Levy & Bell, *supra* note 118, at 2–6; Patrick T. Miyaki, Comment, *Computer Software Defects: Should Computer Software Manufacturers Be Held Strictly Liable for Computer Software Defects?*, 8 SANTA CLARA COMPUTER & HIGH TECH. L.J. 121, 126–28 (1992); Nycum, *supra* note 129, at 16–19; Scott, *supra* note 6, at 461–67; David A. Hall, Note, *Strict Products Liability and Computer Software: Caveat Vendor*, 4 COMPUTER L.J. 373 (1983); Susan Lanoue, Comment, *Computer Software and Strict Products Liability*, 20 SAN DIEGO L. REV. 439, 443–55 (1983).

139. David W. Lannetti, *Toward a Revised Definition of "Product" Under the* Restatement (Third) of Torts: Products Liability, 35 TORT INS. L.J. 845, 857–58 (2000) (outlining three possible scenarios: (1) "good" is more expansive than "product"; (2) "product" is broader than "good"; or (3) the concepts are identical).

140. *Id.* at 875–78. *See generally* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 19(a) (AM. LAW INST. 1998) ("A product is tangible personal property distributed commercially for use or consumption. Other items, such as real property and electricity, are products when the context of their distribution and use is sufficiently analogous to the distribution and use of tangible personal property . . . ."); Lannetti, *supra* note 139, at 865–68 (discussing the longstanding rule that "service providers, unlike product manufacturers or suppliers, are *not* strictly liable for personal injuries resulting from rendered services").

141. *See, e.g.*, Lanoue, *supra* note 138, at 443–47 (noting that "[t]here is no absolute rule that restricts the definition of products to tangible items" and citing case law finding strict products liability for damage to homes caused by electric currents); Scott, *supra* note 6, at 464–67 (discussing cases treating software as "tangible property" for purposes of insurance law, and treating information as "products" for purposes of tort law); Smith, *supra* note 118, at 755–59. *But see* Alces, *supra* note 119, at 301–02 (observing that the comments to section 19 of the Third Restatement of Torts distinguish between intangible information versus intangible forces such as electricity and X-rays).

142. Scott, *supra* note 6, at 462 ("While a majority of courts have held that software is a *good* for the application of the U.C.C. and taxation, that does not mean that software is necessarily a *product* for the application of product liability law."); Zollers et al., *supra* note 117, at 766 ("To date, there have been no reported cases holding a software manufacturer strictly liable for defects in the software."). Over the years, many commentators have cited *Winter v. G.P. Putnam's Sons*, 938 F.2d 1033, 1035 (9th Cir. 1991), as the lone case to suggest that malfunctioning software could be called a defective product—but nothing ever became of this dicta. *See, e.g.*, Calo, *supra* note 79, at 535–36, 536 n.145 (2015); Scott, *supra* note 6, at 466; Zollers et al., *supra* note 117, at 759; Roy W. Arnold, Note, *The Persistence of Caveat Emptor: Publisher Immunity from Liability for Inaccurate Factual Information*, 53 U. PITT. L. REV. 777, 798 (1992); David Berke, Note, *Products Liability in the Sharing Economy*, 33 YALE J. REG. 603, 614–15 (2016); Michael R. Maule, Comment, *Applying Strict Products Liability to Computer Software*, 27 TULSA L.J. 735, 746–51 (1992); Miyaki, *supra*

invoking the "pure economic loss" doctrine.[143] Under this rule, no tort recovery may be obtained for losses that are purely financial, and unaccompanied by bodily injury or property damage.[144] The primary rationale for the economic loss doctrine is to police the conceptual border between contract law and tort law.[145] Various justifications have been proffered for maintaining this rigid wall, all of which reduce in essence to skepticisms about intangible *injuries*, though not necessarily intangible *causes*.[146] The economic loss doctrine sweeps far beyond software, and its wisdom has been hotly debated as a general matter, but its application has had an undeniably profound effect on software litigation.[147]

Thus, it would be easy to conclude that intangibility is the reason for software liability's absence. Yet, there are missing pieces that do not fully add up. The pure economic loss doctrine does not eviscerate all negligence liability[148]; not all software is necessarily a "good" let alone a "product"; software liability cases rarely succeed even where there is physical injury[149]; and those cases that do result in an award—like *Singh*[150]—often turn on additional culpability extrinsic to the code itself. To be sure,

---

note 138, at 126–27 (1992); Lori A. Weber, Note, *Bad Bytes: The Application of Strict Products Liability to Computer Software*, 66 ST. JOHN'S L. REV. 469, 470 (1992). Another case offering tantalizing dicta was *Advent Sys. Ltd. v. Unisys Corp.*, 925 F.2d 670, 673–77 (3d Cir. 1991) (declaring that computer programs are "tangible, moveable and available in the marketplace").

143. Boss & Woodward, Jr., *supra* note 118, at 1535–40 (collecting early software cases that attempted to navigate the line between contract and tort claims); Scott, *supra* note 6, at 470–71.

144. Gaebler, *supra* note 111, at 602–05 ("[A] common statement of the general rule is that there can be no recovery for economic loss in the absence of some physical injury."). There is some variation in how the economic loss doctrine is applied across different states. *See* Vincent R. Johnson, *The Boundary-Line Function of the Economic Loss Rule*, 66 WASH. & LEE L. REV. 523 (2009).

145. *See* Jay M. Feinman, *The Economic Loss Rule and Private Ordering*, 48 ARIZ. L. REV. 813, 813 (2006); Vincent R. Johnson, *The Boundary-Line Function of the Economic Loss Rule*, 66 WASH. & LEE L. REV. 523, 546 (2009); David W. Opderbeck, *Cybersecurity, Data Breaches, and the Economic Loss Doctrine in the Payment Card Industry*, 75 MD. L. REV. 935 (2016); Catherine M. Sharkey, *The Remains of the Citadel (Economic Loss Rule in Products Cases)*, 100 MINN. L. REV. 1845 (2016).

146. *See* Johnson, *supra* note 145, at 541–43 ("A variety of reasons have been offered to justify the economic loss rule, although those reasons 'have not traditionally been clear.' It is said, for example, that liability for negligence that causes only economic harm must be uncompensable under tort law because allowing such recovery would: expose defendants to an unlimited scope of liability; impose liability for damages that are speculative; result in liability that is disproportionate to fault; or have a 'chilling effect on non-negligent conduct.'"); *see also* sources cited *supra* note 111.

147. *See, e.g.*, cases cited *supra* note 8.

148. *See generally* Gaebler, *supra* note 111.

149. *See supra* notes 9–10.

150. *See supra* text accompanying note 1.

intangibility has been an important factor in many cases, but it has not been an explanation.

B. Software's Innovation Function

Even as the debate over software's intangibility receded, a different explanatory movement coalesced around the need to protect software for its importance to innovation and economic growth.[151] This transition to law-and-economics reasoning ramped up as personal computing took off in the 1980s, and then reached max velocity with the explosive growth of the internet in the 1990s.[152] Time and time again, software manufacturers were granted special exemption from liability on the maximalist theory that more software is always better than less. Yet, if intangibility had been a reason in search of a conclusion, the economic rationale was a conclusion in search of a reason. It begged the question: what makes software liability a uniquely existential threat?

The first signs of this rhetorical shift came in the battle over "shrinkwrap" or "tear-open" licenses, which purported to bind customers to contractual obligations as soon as the software was accessed.[153] Enforceability of these licenses was hotly debated in the scholarly literature and loomed over the industry as a question mark.[154] These

---

151. *See* PETER W. HUBER, LIABILITY: THE LEGAL REVOLUTION AND ITS CONSEQUENCES (1988). To be sure, policy discussions concerning intellectual property protection of software began much, much earlier. *See, e.g.*, Dan L. Burk & Mark A. Lemley, *Is Patent Law Technology-Specific?*, 17 BERKELEY TECH. L.J. 1160 n.16 (2002) (collecting articles detailing the "curious history of the patentability of software"); Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 DUKE L.J. 663 (1984) (describing the history of Congress's study concerning the issue of software copyrightability).

152. Concerns regarding the negative effects of tort liability on innovation were broad-based across many sectors of the U.S. economy during this era. *See* Carl T. Bogus, *War on the Common Law: The Struggle at the Center of Products Liability*, 60 MO. L. REV. 1, 77–82 (1995); W. Kip Viscusi & Michael J. Moore, *An Industrial Profile of the Links Between Product Liability and Innovation*, *in* LIABILITY MAZE, *supra* note 74, at 81.

153. *See* Richard H. Stern, *Shrink-Wrap Licenses of Mass-Marketed Software: Enforceable Contracts or Whistling in the Dark?*, 11 RUTGERS COMPUTER & TECH. L.J. 51 (1985); David Einhorn, Note, *The Enforceability of "Tear-Me-Open" Software License Agreements*, 67 J. PAT. & TRADEMARK OFF. SOC'Y 509 (1985).

154. *See generally* RAYMOND T. NIMMER, AM. BAR ASS'N, SOFTWARE LICENSING CONTRACTS: PROPOSAL FOR STUDY BY THE A.B.A. AD HOC COMMITTEE ON THE SCOPE OF THE UCC (1987); Mary Brandt Jensen, *The Preemption of Shrink Wrap Licenses in the Wake of* Vault Corp. v. Quaid Software Ltd., 8 COMPUTER/L.J. 157 (1988); Lemley, *supra* note 133, at 1248 ("Because of the nature of the shrinkwrap license, and because of its potential to rewrite the rules of tort and intellectual property law, courts have viewed such licenses with a skeptical eye."); Ritter, *supra*

license terms carried sweeping limitations on liability and disclaimers of warranty, and this adhesionary abdication of software quality led some courts to question whether shrinkwrap tactics were fair to consumers.[155] Applying a basic UCC analysis, these courts held that customers could not be bound by terms they could not see.[156]

Alarmed by this state of uncertainty, the American Bar Association (ABA) and the National Conference of Commissioners on Uniform State Laws (NCCUSL) launched a joint campaign to modernize UCC Article 2 to offer clearer guidance on shrinkwrap licenses and other software transactions.[157] Professor Raymond Nimmer—who took the lead role in the drafting efforts—flatly rejected as "wrong" the formalist question "whether software or other intangibles constitute goods or whether a contract that licenses use of intangibles constitutes a sale."[158] Instead, Nimmer embraced the functionalist approach of law-and-economics that the law should bend to facilitate the commercial needs of a burgeoning industry offering unique value to the national economy.[159] This emphasis

---

note 131, at 2549 ("These 'shrink-wrap' licenses are a common practice, notwithstanding a recent decision holding such 'contracts' to be unenforceable.").

155. Step-Saver Data Sys., Inc. v. Wyse Tech., 939 F.2d 91, 104 (3d Cir. 1991) (refusing to enforce a box-top license because, *inter alia*, "[w]e are not persuaded that requiring software companies to stand behind representations concerning their products will inevitably destroy the software industry"); Ariz. Retail Sys., Inc. v. Software Link, Inc., 831 F. Supp. 759, 766 (D. Ariz. 1993) (refusing to enforce shrinkwrap license in a suit for breach of warranty); *see also* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1452 (7th Cir. 1996) (finding only three prior cases on the enforceability of shrinkwrap licenses, including *Step-Saver* and *Ariz. Retail*).

156. *See* Novell, Inc. v. Network Trade Ctr., 25 F. Supp. 2d 1218, 1230 (D. Utah 1997); Pamela Samuelson, *Intellectual Property and Contract Law*, 87 CALIF. L. REV. 1, 5 (1999) (asserting that courts "generally hold that the terms contained in such shrinkwrap licenses are unenforceable because the consumer never assented to them").

157. *See* Michael L. Rustad, *Making UCITA More Consumer Friendly*, 18 J. MARSHALL J. COMPUTER & INFO. L. 547, 552–54 (1999); Diane W. Savage, *The Impact of Proposed Article 2B of the Uniform Commercial Code on Consumer Contracts for Information and Computer Software*, 9 LOY. CONSUMER L. REV. 251, 252–53 (1997). For further background on the origins of the UCC recodification efforts, see Ritter, *supra* note 131, at 2534–37.

158. Raymond T. Nimmer, *Intangibles Contracts: Thoughts of Hubs, Spokes, and Reinvigorating Article 2*, 35 WM. & MARY L. REV. 1337, 1343 (1994).

159. *Id.* at 1360–62, 1369 ("Software and other intangibles contracts fit a standard of importance gauged by economic significance under any measure. The information industry accounts for over two percent of the gross national product of this country and affects a broad spectrum of commercial and individual interests. Ongoing developments in information technology promise to continue the exponential growth of that field. Technology (intangibles) contracts underlie virtually all modern areas of commerce driving our present economy."); *see also* Raymond T. Nimmer, *Licensing on the Global Information Infrastructure: Disharmony in Cyberspace*, 16 NW. J. INT'L L. & BUS. 224, 246–47 (1995) ("More so here than in any prior commercial/economic context, an enhanced degree of harmonization and simplification is needed to enable the transactions made possible by the technology

on economic efficiency favored vendor protections above software quality.[160]

Ultimately, the Article 2 revision efforts were rendered moot by the landmark decision *ProCD, Inc. v. Zeidenberg*,[161] which held shrinkwrap licenses enforceable under the *old*, unrevised UCC Article 2.[162] Yet the prevailing argument was identical to Nimmer's, that economic need superseded doctrinal fit. According to Judge Frank Easterbrook, shrinkwrap licenses were enforceable simply because they helped software vendors keep prices low, irrespective of whether the software came wrapped in a box.[163] Low prices benefitted consumers, the court proclaimed—without exploring the second-order costs of externalizing software sloppiness onto consumers.

Meanwhile, internet protectionism kicked into full gear with the launch of Netscape Navigator in 1994.[164] Excitement about "global electronic networks" had already been frothing around the more hidebound

---

to occur . . . . [A] stabilization of contract . . . law would provide immense advantages to the commercialization of cyberspace.").

160. *See* Alces, *supra* note 119, at 272–73 (criticizing the Article 2B draft's treatment of the warranty of merchantability as "inconsistent with the demands of product quality law"); David Nimmer et al., *The Metamorphosis of Contract into Expand*, 87 CALIF. L. REV. 17, 71 (1999) (criticizing the Article 2B draft for "*de facto* favor[ing] those with concentrated interests and large financial resources"); Rustad, *supra* note 157, at 555–60. *But see* AM. LAW INST. & NAT'L CONFERENCE COMM'RS, UNIFORM COMMERCIAL CODE ARTICLE 2B - LICENSES 14, 18 (1998) (circulated draft) (protesting that the public statements "made about the effect of Article 2B on consumer protection" are "political efforts to mislead").

161. 86 F.3d 1447 (7th Cir. 1996) (Easterbrook, J.).

162. *Id.*; *see also* Hill v. Gateway 2000, Inc., 105 F.3d 1147 (7th Cir. 1997). On April 7, 1999, the American Law Institute withdrew support for Article 2B, and the UCC recodifications efforts were abandoned. Unwilling to see a decade's work die in vain, NCCUSL resurrected those efforts as a freestanding uniform act—the Uniform Computer Information Transactions Act (UCITA), also led by Richard Nimmer—which suffered heavy criticism before being withdrawn finally in August 2003. *See generally* William H. Henning, *Amended Article 2: What Went Wrong*, 11 DUQ. BUS. L.J. 131 (2009).

163. *ProCD*, 86 F.3d at 1451–52 ("Only a minority of sales take place over the counter, where there are boxes to peruse . . . . Much software is ordered over the Internet by purchasers who have never seen a box . . . . On Zeidenberg's arguments, these unboxed sales are unfettered by terms—so the seller has made a broad warranty and must pay consequential damages for any shortfalls in performance, two 'promises' that if taken seriously would drive prices through the ceiling or return transactions to the horse-and-buggy age."); *id.* at 1453 ("Competition among vendors, not judicial revision of a package's contents, is how consumers are protected in a market economy . . . . As we stressed above, adjusting terms in buyers' favor might help Matthew Zeidenberg today . . . but would lead to a response, such as a higher price, that might make consumers as a whole worse off.").

164. *See* Lawrence Lessig, *The Path of Cyberlaw*, 104 YALE L.J. 1743, 1745 (1995); Jonathan Zittrain, *A History of Online Gatekeeping*, 19 HARV. J.L. & TECH. 253, 257 (2006).

discussions of boxed software.[165] But the arrival of cyberspace and user-generated content united the computing industry and consumer advocates together in common cause against efforts to regulate code quality.

The failed efforts to ban online pornography remain the most emblematic example of internet exceptionalism.[166] When Congress passed the Communications Decency Act of 1996 (CDA),[167] it attempted to impose minimum software standards that would make the internet "safe" for kids. It did so by criminalizing online distribution of sexually explicit content, unless appropriate age verification or other measures were used to screen access by minors.[168]

While free speech was the nominal headline of the online porn wars, much of the reasoning was framed in terms of potential economic harm. In striking down the CDA provisions as unconstitutional, the district court panel adopted a cost-benefit analysis that emphasized the exceptional qualities of the internet.[169] As one judge highlighted: "Internet communication is an abundant and growing resource" with "very low barriers to entry" for "both speakers and listeners," and the excessive "economic costs associated with compliance with the [CDA] will drive from the Internet speakers whose content falls within the zone of possible prosecution."[170] The panel found that the CDA effectively mandated use of adult verification services by all internet providers[171]—not just

165. Henry H. Perritt, Jr., *Tort Liability, the First Amendment, and Equal Access to Electronic Networks*, 5 HARV. J.L. & TECH. 65, 131–33 (1992) (worrying about the chilling effects of tort liability and arguing that "[a] network service provider that holds itself out as available to all comers should face commensurately less exposure to tort liability for the content carried"); I. Trotter Hardy, *The Proper Legal Regime for "Cyberspace"*, 55 U. PITT. L. REV. 993 (1994).

166. *See generally* Yochai Benkler, *Coase's Penguin, or, Linux and the Nature of the Firm*, 112 YALE L.J. 369 (2002); Bryan H. Choi, *The Anonymous Internet*, 72 MD. L. REV. 501, 516 (2013); Jonathan L. Zittrain, *The Generative Internet*, 119 HARV. L. REV. 1975 (2006).

167. Pub. L. No. 104-104, 110 Stat. 133 (1996).

168. *Id.* § 507, 110 Stat. 137.

169. ACLU v. Reno, 929 F. Supp. 824 (E.D. Pa 1996), *aff'd*, 521 U.S. 844 (1997).

170. *Id.* at 877–78 (Dalzell, J.); *see also id.* at 881 ("My examination of the special characteristics of Internet communication . . . lead me to conclude that the Internet deserves the broadest possible protection from government-imposed, content-based regulation.").

171. Initially, the CDA was ruled overbroad on textual grounds, given the inherent vagueness of interpreting the statutory terms "indecent" and "patently offensive," and the harsh, criminal consequences of guessing wrong. *Id.* at 854–55 (Sloviter, J.), 859–65 (Buckwalter, J.), 870–72 (Dalzell, J). Yet when Congress corrected the textual flaws, see Ashcroft v. ACLU, 535 U.S. 563 (2002), the courts subsequently clarified that it was the mandatory use of adult verification services that was in and of itself overbroad. *Id.* at 656. *But cf.* Digital Economy Act 2017 § 14, https://www.legislation.gov.uk/ukpga/2017/30/section/14/enacted [https://perma.cc/DZ5G-AVWX] (requiring anyone who "makes pornographic material available on the internet to persons in the

pornographers—and that this mandate was problematic because such services were "not technologically or economically feasible for most providers."[172] The Supreme Court echoed and incorporated these concerns in its affirmance of the panel's decision.[173]

In this climate of internet limerence,[174] it was only fitting that the CDA would come to be associated instead with its safe harbor. Section 230, originally drafted as a late amendment to shield internet intermediaries from CDA liability, was the only provision to survive the wreckage.[175] This *soi disant* "Good Samaritan" provision opened with a remarkable declaration: "It is the policy of the United States . . . to promote the continued development of the Internet and other interactive computer services . . . [and] to preserve the vibrant and competitive free market that presently exists for the Internet and other interactive computer services, unfettered by Federal or State regulation."[176] With that broad statement of purpose, courts interpreted section 230 as providing bright-line immunity to internet entities against tort liability for user-generated content.[177] The

---

United Kingdom on a commercial basis" to use age-verification means to block access by "persons under the age of 18").

172. *Reno*, 929 F. Supp. at 846–48, 854, 856 (Sloviter, J.); *id.* at 858 (Buckwalter, J.).

173. Reno v. ACLU, 521 U.S. 844, 881–82 (1997) ("Under the findings of the District Court, however, it is not economically feasible for most noncommercial speakers to employ such [adult] verification.").

174. *See, e.g.*, MICHAEL HAUBEN & RONDA HAUBEN, NETIZENS: ON THE HISTORY AND IMPACT OF USENET AND THE INTERNET (1997) (documenting the exhilaration, obsession, and devotion of early internet users); HOWARD RHEINGOLD, THE VIRTUAL COMMUNITY: HOMESTEADING ON THE ELECTRONIC FRONTIER (2000).

175. David S. Ardia, *Free Speech Savior or Shield for Scoundrels: An Empirical Study of Intermediary Immunity Under Section 230 of the Communications Decency Act*, 43 LOY. L.A. L. REV. 373, 409–12 (2010) (discussing legislative history and how it "upended a set of principles enshrined in common law doctrines that had been developed over decades, if not centuries, in cases involving *offline* intermediaries"); Zittrain, *supra* note 118, at 262. *But see* Fair Hous. Council v. Roommate.com, LLC, 521 F.3d 1157, 1164 n.15 (9th Cir. 2008) ("The Internet is no longer a fragile new means of communication that could easily be smothered in the cradle by overzealous enforcement of laws and regulations applicable to brick-and-mortar businesses."); Danielle Keats Citron & Benjamin Wittes, *The Internet Will Not Break: Denying Bad Samaritans § 230 Immunity*, 86 FORDHAM L. REV. 401, 404–11 (2017).

176. Pub. L. No. 104-104, § 509, 110 Stat. 133, 138 (1996) (codified as amended at 47 U.S.C. § 230(b)(1), (2) (2018)).

177. This broad-based immunity has endured despite serious pushback over the years. *See* Goldman, *supra* note 113; *cf.* Barrett v. Rosenthal, 146 P.3d 510 (Cal. 2006). Recent efforts to limit § 230 immunity have achieved more traction by focusing on sex trafficking and revenge porn. *See* Doe No. 14 v. Internet Brands, Inc., 824 F.3d 846 (9th Cir. 2016); Allow States and Victims to Fight Online Sex Trafficking Act of 2017, Pub. L. No. 115-164, 132 Stat. 1253 (2018); Mary Anne Franks, *"Revenge Porn" Reform: A View from the Front Lines*, 69 FLA. L. REV. 1251 (2017).

scope of this immunity is not infinite,[178] but it has been reliable enough to endow the "move fast and break things" attitude of the Silicon Valley era.[179]

A final episode, the Y2K bug, set the high-water mark of software protectionism.[180] As the year 2000 approached, the software industry discovered a basic error in the way calendar dates had been formatted. To conserve precious disk space and memory usage, dates were commonly saved in two-digit format. Where necessary, the "century" digits were hard-coded as "19" such that the year 2000 would be stored as "00" and read erroneously by the software as 1900.[181] These date-keeping errors were ubiquitous and (like any good doomsday prophecy) the severity of risk was unknowable *ex ante*.[182] Early signs, however, pointed to the possibility that software manufacturers might, for the first time, face open-ended liability for having cut corners in writing code.[183]

Congress blinked. On July 20, 1999, Congress enacted special restrictions on Y2K-related litigation, including a three-year moratorium

---

178. *See* 47 U.S.C. § 230(e); Mary Anne Franks, *The Lawless Internet? Myths and Misconceptions About CDA Section 230*, HUFFINGTON POST (Feb. 17, 2014, 5:35 PM), https://www.huffingtonpost.com/mary-anne-franks/section-230-the-lawless-internet_b_4455090.html [https://perma.cc/54TL-92UE].

179. *See* WITTES & BLUM, *supra* note 16, at 215–17; Carmine Giardino et al., *What Do We Know About Software Development in Startups?*, IEEE SOFTWARE, Sept.–Oct. 2014, at 28; Walter Isaacson, *Resistance Is Futile*, N.Y. TIMES: BOOK REVIEW, June 25, 2017, at 1.

180. Alces & Book, *supra* note 133, at 6–10; Andrew S. Crouch, Comment, *When the Millennium Bug Bites: Business Liability in the Wake of the Y2K Problem*, 22 HAMLINE L. REV. 797 (1999); *see also* Chris Taylor, *The History and the Hype*, TIME, Jan. 18, 1999, at 72–73.

181. The error was, of course, easily foreseeable. A famous, earlier incident occurred at the 1976 Olympics, when the first perfect 10.0 score in gymnastics was displayed as "1.0" because the electronic scoreboard lacked enough digits. Pritha Sarkar, *Nadia Still Turning Heads 40 Years on from Perfect 10*, REUTERS, July 17, 2016, https://reuters.com/article/us-olympics-gymnastics-comaneci/nadia-still-turning-heads-40-years-on-from-perfect-10-idUSKCN0ZY03X [https://perma.cc/ND33-S42Y].

182. Robert K. Hur, Note, *Passing the Y2K Buck: Examining Foundations of Economic Arguments for and Against Liability Limitation*, 11 STAN. L. & POL'Y REV. 193, 195 (1999) (noting that remediation cost estimates varied widely "from $20 billion, to between $300 billion to $600 billion, to $1.6 trillion").

183. Alces & Book, *supra* note 133, at 3 n.7; Crouch, *supra* note 180, at 804–07 (citing an individual settlement agreement of $250,000 and class action settlements of $565,000 and $46 million); Hur, *supra* note 182, at 195 ("Approximately 70 Y2K lawsuits were filed nationwide [before the Year 2000], producing a few large settlements, including one for $7.5 million. Many more legal disputes (as many as 800) proceeded to formal negotiations."). *But cf.* Jeffrey W. Stempel, *A Mixed Bag for Chicken Little: Analyzing Year 2000 Claims and Insurance Coverage*, 48 EMORY L.J. 169, 173 (1999) ("[T]he number of articles written about the Year 2000 matter dwarfs the handful of lawsuits actually filed . . . . [T]he Y2K problem would appear to be less catastrophic than many suggest.").

on claims, heightened pleading requirements, and limitations on recovery.[184] The preamble of the Year 2000 Responsibility and Readiness Act (Y2K Act) proclaimed it to be "in the national interest that producers and users of technology products concentrate their attention and resources in the time remaining before January 1, 2000" on fixing Y2K issues rather than on litigation defense "so as to minimize possible disruptions associated with computer failures."[185] Congress further admonished such litigation as causing "a range of undesirable effects" including first and foremost "waste [of] technical and financial resources that are better devoted to . . . ensuring that systems remain or become operational."[186] In sum, Congress sent a strong signal once again that software manufacturers would not be held to task for even the most trivial of errors, simply because the software industry was too important.

In the end, the Y2K hype amounted to very little in actual damages; instead, most of the costs accrued from the remediation efforts that came before the critical date.[187] To recover their remediation costs, some

184. Pub. L. No. 106-37 (1999) (codified at 15 U.S.C. §§ 6601–17 (2018)); *see also* Alces & Book, *supra* note 133, at 17–18; Christopher M. Fairman, *Heightened Pleading*, 81 TEX. L. REV. 551, 613 (2002) ("Despite the absence of factual proof of a [Y2K] litigation explosion or that it would be fueled by frivolous cases, Congress proceeded with regulation designed to thwart the impending tidal wave."); Martha A. Sabol & Beth Diebold, *Readiness and Responsibility in the Year 2000: A Look at Y2K Legislation*, 11 LOY. CONSUMER L. REV. 217 (1999); *cf.* Anthony J. Bellia Jr., *Federal Regulation of State Court Procedures*, 110 YALE L.J. 947, 953–55 (2001) (noting attacks by opposing senators that the bill was "an arrogant dismissal of the basic constitutional principle of federalism" and was "doing away" with the Tenth Amendment to the Constitution). The Y2K Act excluded claims for personal injury and wrongful death. *See* 15 U.S.C. § 6603(c); *cf.* USA PATRIOT Act, Pub. L. No. 107-56 § 814, 115 Stat. 272, 384 (2001) (codified as amended at 18 U.S.C. § 1030(g)) ("No [civil] action may be brought under [the Computer Fraud and Abuse Act] for the negligent design or manufacture of computer hardware, computer software, or firmware.").

185. 15 U.S.C. § 6601(a)(2).

186. *Id.* § 6601(a)(3)(B). Congress also disparaged such litigation as "insubstantial" and "frivolous," as well as "unnecessary, time-consuming, and costly." *Id.* § 6601(a)(6), (7), (8).

187. U.S. DEP'T OF COMMERCE, THE ECONOMICS OF Y2K AND THE IMPACT ON THE UNITED STATES 24 (1999) (estimating costs as having run "about $30 billion a year in 1998 and 1999 and a cumulative cost in the neighborhood of $100 billion for the period 1995 through 2001"); Fairman, *supra* note 184, at 616 ("Despite the deluge predictions, Y2K litigation has been a trickle."); Steve Lohr, *Computers Prevail in First Hours of '00*, N.Y. TIMES, Jan. 1, 2000; Tony Pyne, *The Exclusion of Y2K Related Losses from Aviation Insurance Policies: Practicalities, Politics, and Legalities*, 65 J. AIR L. & COM. 769, 771–77 (2000) (reporting a host of minor glitches from around the world, but that "[n]o widespread chaos or failure of systems occurred"); David Segal, *A Y2K Glitch for Lawyers: Few Lawsuits*, WASH. POST, Jan. 10, 2000, at A1; *cf.* Benjamin H. Barton, *Tort Reform, Innovation, and Playground Design*, 58 FLA. L. REV. 265, 282–83 (2006) (arguing that, counter to economic predictions, "[t]he expense of fixing the Y2K problem turned out to be a tremendous benefit for the economy instead of a detriment," because "Y2K gave companies an excuse to clean up their software and hardware underbrush").

customers sued software manufacturers claiming that the Y2K bug was a product defect. Those defect claims were summarily dismissed because "there is nothing inherently wrong with computer software that assumes a two-digit year entry means the Twentieth Century."[188] Other business customers—unable to obtain direct relief from software manufacturers—looked to insurance companies for recompense.[189] Yet in this too they were frustrated. Insurers resisted payment of Y2K remediation claims, and won decisively on the legal theory that the Y2K bug was an "inherent vice" or "latent defect," a standard exclusion from coverage in most insurance policies.[190] Courts explained that the Y2K bug was a latent defect because it was present at the time of creation, not introduced by external factors at a later date. Thus, the Y2K bug was simultaneously a defect and not a defect, as long as the end result was no financial penalties for software manufacturers.

While the economic significance of software is undeniable, the glaring omission from these discussions of cost-benefit balancing is that they rarely if ever attempt any analysis of the actual code design itself. This lacuna signals where the root cause for judicial avoidance may be found. If code designs are unevaluable and therefore indistinguishable at law, then every software liability claim threatens bet-the-industry litigation. In short, the argument that tort liability threatens the existence of the software industry necessarily rests on the tacit assumption that courts are somehow ill-equipped to define a standard of reasonable software quality.

---

188. Kaczmarek v. Microsoft Corp., 39 F. Supp. 2d 974, 977 (N.D. Ill. 1999); *accord* Against Gravity Apparel, Inc. v. Quarterdeck Corp., 699 N.Y.S.2d 368, 370 (N.Y. App. Div. 1999) (slip op.) ("Also without merit is plaintiff's claim, based on UCC 1–204, that the software's Y2K noncompliance is a latent defect . . . .").

189. Kenneth S. Abraham, *Peril and Fortuity in Property and Liability Insurance*, 36 TORT & INS. L.J. 777, 797 (2001) (observing that "a number of major policyholders" had filed insurance claims under "sue and labor" clauses); Jeffrey T. Piampiano, Comment, *Y2K Remediation: Who Should Bear the Cost?*, 4 J. SMALL & EMERGING BUS. L. 411 (2000) (describing three complaints filed by Nike, Port of Seattle, and Kmart against their respective insurers); *cf.* Stempel, *supra* note 183, at 174 ("To date, much of the Year 2000 discussion has simply, and probably incorrectly, assumed that big Y2K losses for business mean big insurance payments . . . . There has been a disturbing tendency for discussion of the Y2K problem to seemingly assume that Y2K losses are of a uniform type . . . .").

190. GTE Corp. v. Allendale Mut. Ins. Co., 372 F.3d 598, 609 (3d Cir. 2004) ("The problem in this case was not that a program or system malfunctioned, or some external threat caused damage to GTE's systems. Rather, the system performed in exactly the manner it was designed to operate—the problem is that the system as designed and specified did not permit recognition of dates in the 21st century."); State v. Allendale Mut. Ins. Co., 2007 MT 83, 154 P.3d 1233; Port of Seattle v. Lexington Ins. Co., 111 Wash. App. 901, 48 P.3d 334 (2002). An "inherent vice" is defined as "any existing defects, diseases, decay or the inherent nature of the commodity which will cause it to deteriorate with the lapse of time." *Id.* at 909–10, 48 P.3d at 338–39.

C. Software's Complexity Anomie

Intangibility and innovation capture the intuition, but do not explain, why software liability is a hard doctrinal problem. The true culprit is a very basic property of software—computational complexity—which defies conventional judicial methods of assessing reasonableness.[191]

Software strives to conceal or abstract away the "machine layer" as much as possible.[192] That abstraction provides unprecedented plasticity and reproducibility, but the flip side is that code offers few intrinsic constraints that govern "normal" behavior, the way physical objects must obey laws of motion and gravity.[193] Utilizing this freedom to its fullest advantage results in programs having so many possible permutations of machine-states that it is mathematically impossible to guarantee

---

191. *See Hearing on Deciphering the Debate over Encryption: Industry and Law Enforcement Perspectives Before the Subcomm. on Oversight and Investigations of the H. Comm. on Energy & Commerce*, 114th Cong. 2 (2016) (statement of Matt Blaze, Professor, University of Pennsylvania) ("[C]omputer science does not yet know how to build complex, large-scale software that has reliably correct behavior."), https://energycommerce.house.gov/sites/democrats.energycommerce.house.gov/files/Testimony-Blaze-OI-Encryption%20Hrg-2016-04-19.pdf [https://perma.cc/DD3E-NJ4T].

192. *See* David Chisnall, *C Is Not a Low-Level Language*, COMM. ACM, July 2018, at 44 (explaining that even the C language, which is considered "close to the metal," relies on substantial abstractions from the physical machine); Edward A. Lee, *Cyber Physical Systems: Design Challenges*, 11 IEEE SYMP. ON OBJECT & COMPONENT-ORIENTED REAL-TIME DISTRIBUTED COMPUTING 363, 364 (2008) (noting that digital circuit designers have "learned to harness intrinsically stochastic processes (the motions of electrons) to deliver a precision and reliability that is unprecedented in the history of human innovation"). *See generally* MAURICE J. BACH, THE DESIGN OF THE UNIX OPERATING SYSTEM (1986); ZITTRAIN, *supra* note 112, at 67–70; David D. Clark, *The Design Philosophy of the DARPA Internet Protocols*, 18 ACM SIGCOMM COMPUTER COMM. REV. 106 (1988); JEFF SHNEIDMAN ET AL., HARV. TECHNICAL REP. NO. TR-21-04, HOURGLASS: AN INFRASTRUCTURE FOR CONNECTING SENSOR NETWORKS AND APPLICATIONS (2004). *But cf.* Kevin Driscoll et al., *Byzantine Fault Tolerance, from Theory to Reality*, *in* 22 PROC. INT'L CONF. ON COMPUTER SAFETY, RELIABILITY & SECURITY 235, 239 (2003) (describing how a digital signal can get "stuck" at "1/2" that is neither a 0 nor a 1).

193. *See* Lee, *supra* note 192, at 364 ("The fact is that even the simplest C program is not predictable and reliable in the context of CPS [cyber-physical systems] because *the program does not express aspects of the behavior that are essential to the system*."); Ragunathan Rajkumar et al., *Cyber-Physical Systems: The Next Computing Revolution*, 47 PROC. DESIGN AUTOMATION CONF. 731, 735 (2010) (explaining one research challenge for CPS is the need for programming abstractions that can capture "[p]hysical properties such as the laws of physics and chemistry, safety, real-time and power constraints, . . . robustness, and security characteristics"); Lui Sha et al., *Cyber-Physical Systems: A New Frontier*, 2008 PROC. IEEE INT'L CONF. ON SENSOR NETWORKS, UBIQUITOUS & TRUSTWORTHY COMPUTING 1, 4 ("Existing hardware design and programming abstractions for computing are largely built on the premise that the principal task of a computer is data transformation. Yet cyber-physical systems are real-time systems. This requires a critical re-examination of existing hardware and software architectures that have been built over the last several decades.").

correctness through *ex post* testing.[194] Yet, imposing rigid controls *ex ante* for the sake of correctness makes it infuriatingly hard to write code that is actually useful.[195] Given this impossible tradeoff, almost all software manufacturers prioritize functionality and features over safety or validity. The consensus among the cybersecurity community is that one could throw infinite resources at development and quality assurance, yet still emerge with errors so basic a jury would be appalled.[196]

The platonic ideal would be to require all software be written in programming languages employing "formal methods," which rely on mathematical theory to enforce correctness of code as it is being written.[197] The avionics industry is the most successful example of this

---

194. *See* Joshua A. Kroll et al., *Accountable Algorithms*, 165 U. PA. L. REV. 633, 647–53 (2017) (describing various shortcomings with static testing and dynamic testing, as well as more fundamental limitations caused by the non-computability of certain "NP-hard" problems); Yegor Bugayenko, *Discovering Bugs, or Ensuring Success?*, COMM. ACM, Sept. 2018, at 12 (collecting commentary that quality assurance testing cannot guarantee that software is error-free); W. Richards Adrion et al., *Validation, Verification, and Testing of Computer Software*, COMPUTING SURVEYS, June 1982, at 159, 164–66.

195. *See* Brian Randell, *System Structure for Software Fault Tolerance*, SE-1 IEEE TRANSACTIONS ON SOFTWARE ENG'G 220, 220 (1975) ("The difference in complexity arises from the fact that the 'machines' that hardware designers produce have a relatively small number of distinctive internal states, whereas the designer of even a small software system has, by comparison, an enormous number of different states to consider—thus one can usually afford to treat hardware designs as being 'correct,' but often cannot do the same with software even after extensive validation efforts."); Harold "Bud" Lawson, *The March into the Black Hole of Complexity*, COMM. ACM, May 2018, at 43; *see also* Derek E. Bambauer, *Ghost in the Network*, 162 U. PENN. L. REV. 1011, 1020–25 (2014); Hurwitz, *supra* note 58, at 1501–04.

196. *See, e.g.*, Kroll et al., *supra* note 194, at 647 n.34 (describing how the Heartbleed episode "underscores how difficult it can be to find small and simple mistakes"); Kevin Poulsen, *Behind iPhone's Critical Security Bug, a Single Bad 'Goto'*, WIRED (Feb. 22, 2014, 11:27 AM), https://www.wired.com/2014/02/gotofail/ [https://perma.cc/N3X8-9N4D]; *Sunk by Windows NT*, WIRED (July 24, 1998, 4:35 PM), https://www.wired.com/1998/07/sunk-by-windows-nt/ [https://perma.cc/X5RN-VFH6] (divide-by-zero error caused Navy ship to lose control of its propulsion system for several hours); *see also* Diomidis Spinellis, *Modern Debugging: The Art of Finding a Needle in a Haystack*, COMM. ACM, Nov. 2018, at 124, 134 (describing modern best practices in debugging techniques, and exhorting that "[n]o bug can elude a programmer who perseveres"). *But see* Maggie Hamill & Katerina Goseva-Popstojanova, *Common Trends in Software Fault and Failure Data*, 35 IEEE TRANSACTIONS ON SOFTWARE ENG'G 484, 484 (2009) (empirical study finding that "individual failures are often caused by multiple faults spread throughout the system").

197. *See* Kroll et al., *supra* note 194, at 649, 662–65; Imran Quadri et al., *Modeling Methodologies for Cyber-Physical Systems: Research Field Study on Inherent and Future Challenges*, 36 ADA USER J. 246, 247 (2015); Gerwin Klein et al., *Formally Verified Software in the Real World*, COMM. ACM, Oct. 2018, at 68; Nuno P. Lopes et al., *Practical Verification of Peephole Optimizations with Alive*, COMM. ACM, Feb. 2018, at 84 (applying formal methods to compilers); Chris Newcombe et al., *How Amazon Web Services Uses Formal Methods*, COMM. ACM, Apr. 2015, at 66; *cf.* Baishakhi Ray et al., *A Large-Scale Study of Programming Languages and Code Quality in Github*, COMM. ACM, Oct.

top-down approach. In the 1970s, in order to streamline the number of programming languages in use, the Department of Defense (DoD) developed a new programming language—named Ada after Ada Lovelace, the first programmer—and in 1987 issued the "Ada mandate," a requirement that Ada "shall be the single, common, computer programming language for Defense computer resources."[198] Due to national security considerations, Ada was designed from the ground up using formal methods, and thus it enforces strong typing and other rigid constraints on code structure that reject programmer sloppiness and error at the outset. Those limitations also sealed Ada's unpopularity among the general software community, which in turn forced the DoD to abandon its Ada mandate in 1997.[199] Nevertheless, some safety-critical applications such as military aircraft have continued to rely on legacy Ada code, and the uniquely tight regulation of the U.S. aircraft industry has allowed the FAA to entrench requirements to use formal methods for avionics software.[200] Even in this high-stakes area, however, training and compliance remain spotty.[201] Cost increases of using a language like Ada

---

2017, at 91 (empirical comparison of error-proneness across programming languages). *But see* Peter Alvaro & Severine Tymon, *Abstracting the Geniuses Away from Failure Testing*, COMM. ACM, Jan. 2018, at 55 (describing the historical failure of formal methods and model checkers, because "[m]odern distributed systems are simply too large, too heterogeneous, and too dynamic for these classic approaches to software quality to take root").

198. U.S. DEP'T OF DEF., DIRECTIVE NO. 3405.1 § 4.3.1 (1987); *see also* Pub. L. No. 101-511, § 8092, 104 Stat. 1856, 1896 (1990) ("[A]ll Department of Defense software shall be written in the programming language Ada."); Ricky E. Sward, *The Rise, Fall and Persistence of Ada*, 2010 ACM ANN. INT'L CONF. ON ADA & RELATED TECH. 71, 71–74 (2010); Benjamin M. Brosgol, *Ada in the 21st Century*, J. DEF. SOFTWARE ENG'G, Mar. 2001, at 20.

199. COMPUT. SCI. & TELECOMM. BD., NAT'L RESEARCH COUNCIL, ADA AND BEYOND: SOFTWARE POLICIES FOR THE DEPARTMENT OF DEFENSE 7 (1997) ("Hopes for broad commercial adoption of Ada have not been realized, however. Its commercial use has been eclipsed by other languages, such as C, then C++, and, most recently, Java. DOD's inclusive approach in the development of the language, as well as its promotional campaigns in support of Ada, do not appear to have been successful in fostering adoption of the language beyond defense and other mission-critical applications."); *see also* Kroll et al., *supra* note 194, at 649 n.44 (observing that "developers often choose memory unsafe languages for performance and other reasons").

200. *See* RADIO TECHNICAL COMM'N FOR AERONAUTICS, INC., DO-178C: SOFTWARE CONSIDERATIONS IN AIRBORNE SYSTEMS AND EQUIPMENT CERTIFICATION (2011). DO-178C replaced the older standard, DO-178B. *See* FAA, U.S. DEP'T OF TRANSP., ADVISORY CIRCULAR 20-115C: AIRBORNE SOFTWARE ASSURANCE (2013); *see also* QI D. VAN EIKEMA HOMMES, NHTSA, ASSESSMENT OF SAFETY STANDARDS FOR AUTOMOTIVE ELECTRONIC CONTROL SYSTEMS (2016) (comparing DO-178C to other safety standards for electronic control systems). *See generally* Martin, *supra* note 74, at 488.

201. HUSSEIN YOUSSEF, SAE INT'L, VERIFICATION AND VALIDATION OF COMPLEX SYSTEMS 2 (2011) ("Formal methods however are not considered mainstream for large, complex software systems found in aerospace, except for developments at the component level."); Chong, *supra* note

are estimated at an additional 75% to 150% of total development costs.[202] Low usage rates also cause these languages to suffer from undercapitalization in upkeep and resources, and to forfeit positive spillovers that more popular languages receive.

In other safety-critical domains, movements to improve code quality have been substantially weaker.[203] In the automotive industry, the National Highway Transportation Safety Administration (NHTSA) has repeatedly deferred issuing guidance on software safety standards; instead, the industry operates largely by self-regulation.[204] In the early 1990s, the U.K. government provided limited seed funding for an initiative named the Motor Industry Software Reliability Association (MISRA).[205] An initial standard was published in 1994, which then

---

31 ("Yet not all life-critical systems—indeed, not even all aircraft—are required to comply with such baselines. Software for unmanned aerial vehicles (UAVs) need not meet the DO-178C standard."); Jean-Pierre Rosen, *Is Ada Education Important?*, 29 ADA USER J. 146, 208–09 (2008) ("[M]any people had few, if any, Ada education before they were assigned to an Ada project . . . . People just write the C program, 'translate' it (badly) into Ada, measure, and make the general conclusion: 'Ada is slower.'"); *see also* Kenneth Magel, *Revisiting the Impact of the Ada Programming Language*, COMPUTER, Sept. 2017, at 10 ("Outside of its use in safety-critical applications, Ada has declined in popularity in recent years. The 2016 IEEE Spectrum ranking of programming languages based on relative popularity placed Ada 40th among all languages it highlighted.").

202. Andreas Wölfl et al., *Generating Qualifiable Avionics Software: An Experience Report*, 30 IEEE/ACM INT'L CONF. ON AUTOMATED SOFTWARE ENG'G 726 (2015). *But cf.* Kroll et al., *supra* note 194, at 665 (predicting "the costs of building fully verified software will likely drop precipitously in the coming decades, leading to wide adoption in the software industry due to the benefits of reduced security exposure and the elimination of many types of software bugs").

203. *See, e.g.*, Manfred Broy, *Challenges in Automotive Safety Engineering*, 28 ACM PROC. INT'L CONF. ON SOFTWARE ENG'G 33 (2006); Alexander Pretschner et al., *Software Engineering for Automotive Systems: A Roadmap*, 2007 IEEE FUTURE SOFTWARE ENG'G 55 (2007); Corman & Woods, *supra* note 38, at 59–60 (analyzing automakers' failures to make progress on cybersecurity).

204. NHTSA, U.S. DEP'T OF TRANSP., AUTOMATED DRIVING SYSTEMS 2.0: A VISION FOR SAFETY 11 (2017), https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf [https://perma.cc/ZLA5-SVXH] ("Entities are encouraged to design their ADSs following established best practices for cyber vehicle physical systems. Entities are encouraged to consider and incorporate voluntary guidance, best practices, and design principles published by National Institute of Standards and Technology (NIST), NHTSA, SAE International, the Alliance of Automobile Manufacturers, the Association of Global Automakers, the Automotive Information Sharing and Analysis Center (Auto-ISAC), and other relevant organizations, as appropriate."); David Benjamin, *Toyota Underestimated 'Deadly' Risks*, EE TIMES (Apr. 1, 2014), https://www.eetimes.com/document.asp?doc_id=1321734 (last visited Jan. 31, 2019); *see also* NHTSA, U.S. DEP'T OF TRANSP., PRELIMINARY STATEMENT OF POLICY CONCERNING AUTOMATED VEHICLES 7–9 (2013), https://www.nhtsa.gov/staticfiles/rulemaking/pdf/Automated_Vehicles_Policy.pdf [https://perma.cc/AUV9-CD73].

205. *A Brief History of MISRA*, MISRA, https://www.misra.org.uk/MISRAHome/AbriefhistoryofMISRA/tabid/69/Default.aspx [https://perma.cc/QBN2-PX5S]; Les Hatton, *Safer Language Subsets: An Overview and a Case History, MISRA C*, 46 INFO. & SOFTWARE

evolved by 1998 into "MISRA C": a set of advisory guidelines that programmers using the C language should take into consideration when writing software for vehicle-embedded systems.[206] Since its development in the early 1970s, C has reigned as the most dominant general-purpose programming language because of its versatility, simplicity, and ease-of-use.[207] It does not impose rigorous checks on code quality, deferring that task instead to the programmer.[208] MISRA C was an effort to instill better hygiene practices among automotive programmers,[209] akin to a hospital policy that gently reminds doctors to wash their hands. Adoption of MISRA C has been voluntary, and low.[210] A competing standard initiated by continental European automotive manufacturers, AUTOSAR C++, appears to have fared better, but its primary objective is standardization of software architectures for cross-platform compatibility.[211] Safety improvements are treated as a secondary effect of standardization.[212] Other automotive trade groups vying for relevance include SAE International (blind-spot monitoring systems), Consumer Electronics for

---

TECH. 465, 469 (2004); Chris Tapp, *An Introduction to MISRA C++*, 1 SAE INT'L J. PASSENGER CAR – ELEC. & ELEC. SYS. 265 (2009).

206. *See* MISRA, GUIDELINES FOR THE USE OF THE C LANGUAGE IN VEHICLE BASED SOFTWARE (1998). Two subsequent editions were published in 2004 and 2012.

207. Dennis M. Ritchie, *The Development of the C Language*, *in* HISTORY OF PROGRAMMING LANGUAGES II 671, 685 (Thomas J. Bergin, Jr. et al. eds., 1993) ("C remains a simple and small language, translatable with simple and small compilers . . . . A parsimonious, pragmatic approach influenced the things that went into C: it covers the essential needs of many programmers, but does not try to supply too much."); Stephen Cass, *The 2017 Top Programming Languages*, IEEE SPECTRUM (July 18, 2017), https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages [https://perma.cc/VMS9-BFK5] (ranking C as the top language for non-Web applications, and second overall).

208. Ritchie, *supra* note 207 (noting the "tolerance of C compilers to errors in type" such as arrays and pointers).

209. Hatton, *supra* note 205, at 466.

210. *See In re* Toyota Motor Corp. Unintended Acceleration Litig., 978 F. Supp. 2d 1053, 1094 n.70 (C.D. Cal. 2013) ("Toyota . . . does not use MISRA coding standards used by other two other [sic] major auto manufacturers . . . . That Toyota has adopted its own coding standards rather than following the (voluntary) MISRA standards is uncontroverted, although the parties do not agree whether Toyota's internal coding standards incorporate MISRA standards or the equivalent.").

211. *History*, AUTOSAR, https://www.autosar.org/about/history/ [https://perma.cc/7SWD-DF46] (describing initial discussions between BMW, Bosch, Continental, DaimlerChrysler, and Volkswagen). AUTOSAR is short for AUTomotive Open System ARchitecture. *See* AUTOSAR, https://www.autosar.org [https://perma.cc/37KJ-2WQH]. In response, MISRA published its own version for C++ in 2008. MISRA, GUIDELINES FOR THE USE OF THE C++ LANGUAGE IN CRITICAL SYSTEMS (2008).

212. HOMMES, *supra* note 200, at 7–8.

Automotive (mobile device interfaces), and the Automotive Electronics Council.[213]

Because correct code is so difficult to write, standard practice among software engineering firms is to run extensive testing for quality assurance (QA) after the fact.[214] The main strategy of QA testing is to run the software through as many different scenarios as feasible, to make sure nothing obvious is amiss.[215] But from a mathematical theory perspective, this strategy is provably incomplete.[216] For any reasonably complex software, there are more possible permutations of machine-states than can be tested in finite time.[217] This inevitable blind zone explains why all but the simplest software is susceptible to "zero-day" exploits.[218] Nor does it work to break up the testing into smaller modules. The composition of two provably correct segments of code does not yield a whole that is provably correct, because the composition generates new unknown interactions between the modules.[219] In short, after-the-fact testing is useful but limited, and unable to offer safety guarantees of any kind.

---

213. Crane et al., *supra* note 64, at 281.

214. *See* IEEE COMPUTER SOC'Y, IEEE 730-2014—IEEE STANDARD FOR SOFTWARE QUALITY ASSURANCE PROCESSES (2014).

215. *See* Kroll et al., *supra* note 194, at 652–53 (explaining that "no testing regime can establish any property for all possible programs" but that testing can be useful "in specific cases, especially when those cases have been designed to facilitate testing"); *cf.* Aarian Marshall, *We've Been Talking About Self-Driving Car Safety All Wrong*, WIRED (Oct. 29, 2018, 8:00 AM), https://www.wired.com/story/self-driving-cars-safety-metrics-miles-disengagements/ [https://perma.cc/56EK-B2LT].

216. *See* Kroll et al., *supra* note 194, at 650 n.49 (citing H.G. Rice, *Classes of Recursively Enumerable Sets and Their Decision Problems*, 74 TRANSACTIONS AM. MATHEMATICAL SOC'Y 358 (1953)); *cf.* Bambauer, *supra* note 195 (discussing known unknowns and unknown unknowns).

217. *See* Kroll et al., *supra* note 194, at 650 nn.48–49 (explaining that "achieving complete coverage of a program's behavior by testing alone is considered impossible," due to the fundamental problem of "Combinatorial Explosion" that affects "all but the very simplest programs"); *id.* at 652 (describing Alan Turing's "Halting Problem" as the canonical example of a noncomputable problem).

218. *See* Mailyn Fidler, *Government Acquisition and Use of Zero-Day Software Vulnerabilities*, *in* CYBER INSECURITY, *supra* note 6, at 279–80.

219. Benjamin Beurdouche et al., *A Messy State of the Union: Taming the Composite State Machines of TLS*, 36 IEEE SYMP. ON SECURITY & PRIVACY 535, 535–36 (2015) (explaining how even systems that are well-understood in isolation can generate "disastrous misunderstandings" when combined into a composite state machine); Blaze, *supra* note 191, at 2 n.2 ("[A]dding new features to a system that makes it twice as large generally has the effect of making it far more than twice as vulnerable."); Jefrey Voas, *Composing Software Component "Ilities"*, IEEE SOFTWARE, July/Aug. 2001, at 16; *see also* J.L. Fiadeiro, *On the Emergence of Properties in Component-Based Systems*, 5 PROC. ALGEBRAIC METHODOLOGY & SOFTWARE TECH. 421 (1996); Khaled Md. Khan & Jun Han, *A Security Characterisation Framework for Trustworthy Component Based Software Systems*, 27 PROC. ANN. INT'L COMPUTER SOFTWARE & APPLICATIONS CONF. (2003) ("Repeated experiences suggest that just relying on the security claims made by the component developer such as 'secure

Even when errors are known and fixable, many more obstacles lurk. Patching a bug can easily introduce new errors.[220] This is true not only because the patch might be poorly written, but also because the composition problem generates new, unverifiable interactions.[221] For example, that very fear has been cited by medical device manufacturers who have refused to patch cybersecurity vulnerabilities because they do not want to lose their FDA clearances.[222] When security researchers reported basic vulnerabilities in infusion pumps that would allow remote hackers to inject fatal doses into patients, the expected response was that manufacturers would act immediately to issue security patches.[223] Instead, manufacturers objected that any alterations to the medical device software might cause it to fall out of compliance and forfeit FDA approval.[224] In

---

component' may not be very appealing to the software composers. In current practices, software composers are almost forced to compose systems with components for which they have partial or no knowledge about their underlying security properties.").

220. *See* Robert M. Lee, *Protecting Industrial Control Systems in Critical Infrastructure*, *in* CYBER INSECURITY, *supra* note 6, at 31, 34–36; Alvaro A. Cárdenas et al., Challenges for Securing Cyber-Physical Systems (July 18, 2009) (unpublished report), https://ptolemy.berkeley.edu/projects/chess/pubs/601/cps-security-challenges.pdf [https://perma.cc/8BZV-HT8J] (asserting that "software patching and frequent updates, are not well suited for control systems" and citing anecdotally the accidental shutdown of a nuclear power plant on March 7, 2008, because of a routine reboot of a monitoring device after a security update).

221. *See* Blaze, *supra* note 191, at 2 n.2 ("[E]ach new software component or feature operates not just in isolation, but potentially interacts with everything else in the system, sometimes in unexpected ways that can be exploited.").

222. FDA, GUIDANCE FOR INDUSTRY: CYBERSECURITY FOR NETWORKED MEDICAL DEVICES CONTAINING OFF-THE-SHELF (OTS) SOFTWARE 4 (2005), https://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm077823.pdf [https://perma.cc/H3DQ-FPG6] ("It is possible, but unlikely, that a software patch will need a new 510(k) submission."); *see also* FDA, DECIDING WHEN TO SUBMIT A 510(K) FOR A SOFTWARE CHANGE TO AN EXISTING DEVICE 11 (2017), https://www.fda.gov/downloads/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm514737.pdf [https://perma.cc/H24U-VNVM] ("In many cases, a change made solely to strengthen cybersecurity is not likely to require submission of a new 510(k).").

223. Chunxiao Li et al., *Hijacking an Insulin Pump: Security Attacks and Defenses for a Diabetes Therapy System*, 13 IEEE INT'L CONF. E-HEALTH NETWORKING, APPLICATIONS & SERVS. 150 (2011); Kevin Fu, Trustworthy Medical Device Software (prepublication draft 2011), https://spqr.eecs.umich.edu/papers/fu-trustworthy-medical-device-software-IOM11.pdf [https://perma.cc/K4KQ-XN8F]; Barry Meier, *More Oversight Due for Infusion Pumps*, N.Y. TIMES, Apr. 24, 2010, at B1; *see also* Daniel Halperin et al., *Security and Privacy for Implantable Medical Devices*, IEEE COMPUTER SOC'Y, Jan.–Mar. 2008, at 30.

224. *See* Laura Hagen, *Coding for Health: Cybersecurity in Medical Devices*, HEALTH LAW., June 2016, at 25 (infusion pump manufacturer was "not interested in verifying that other pumps are vulnerable"); Daniel B. Kramer & Kevin Fu, *Cybersecurity Concerns and Medical Devices*, 318 J. AM. MED. ASS'N 2077, 2078 (2017) (describing careful efforts by FDA to allay anxieties of medical device manufacturers when announcing a required firmware upgrade for pacemakers).

response, the FDA has issued multiple statements encouraging medical device manufacturers to provide security updates for known vulnerabilities.[225] Nevertheless, the FDA admits it cannot provide any guarantees, since changes to software could indeed alter functionality substantially enough that revocation of approval would be warranted.[226]

III. CRASHWORTHY CODE: A RULE OF EQUANIMITY

There are two main takeaways from the discussion above. The first is that code crashes will remain inevitable, even in safety-critical settings such as cyber-physical systems, because of fundamental attributes of software technology. Although careful design and testing are necessary components of software quality, a guarantee of error-free code is not possible. This axiom is so well-accepted among the software engineering community that cybersecurity experts have long advocated a strategic shift from prevention to mitigation.[227]

The second lesson is that conventional approaches to software liability law will remain stalled, even in safety-critical settings such as cyber-physical systems, because software errors defy easy legal categorization. No amount of testing can guarantee the absence of errors, yet more testing does make code more reliable, so it is not clear how much testing is

---

225. FDA, POSTMARKET MANAGEMENT OF CYBERSECURITY IN MEDICAL DEVICES 13 (2016), https://www.fda.gov/downloads/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm482022.pdf [https://perma.cc/JGX9-MC9S] ("Manufacturers should respond in a timely fashion to address identified vulnerabilities.").

226. *See* FDA, GENERAL PRINCIPLES OF SOFTWARE VALIDATION: FINAL GUIDANCE FOR INDUSTRY AND FDA STAFF 8 (2002), https://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm085371.pdf [https://perma.cc/6ZAC-RDX6] ("Seemingly insignificant changes in software code can create unexpected and very significant problems elsewhere in the software program."); *cf.* Shyamnath Gollakota et al., *They Can Hear Your Heartbeats: Non-Invasive Security for Implantable Medical Devices*, 2011 PROC. ACM SIGCOMM CONF. 2, https://dl.acm.org/citation.cfm?id=2018438 [https://perma.cc/CNK7-R7G2] ("Between 1999 and 2005, the number of recalls of software-based medical devices more than doubled; more than 11% of all medical-device recalls during this time period were attributed to software failures.").

227. NAT'L INST. STANDARDS & TECH., U.S. DEP'T OF COMMERCE, SYSTEMS SECURITY ENGINEERING: CYBER RESILIENCY CONSIDERATIONS FOR THE ENGINEERING OF TRUSTWORTHY SECURE SYSTEMS (Draft NIST Special Publication 800-160, vol. 2) (Mar. 2018), https://csrc.nist.gov/CSRC/media/Publications/sp/800-160/vol-2/draft/documents/sp800-160-vol2-draft.pdf [https://perma.cc/K5E9-3ULX]; Azad M. Madni & Scott Jackson, *Towards a Conceptual Framework for Resilience Engineering*, 3 IEEE SYS. J. 181 (2009); *see also* Bambauer, *supra* note 195, at 1016, 1029 ("Cybersecurity cannot prevent the ghost in the network; instead, it should seek to cabin its depredations. Mitigation—not prevention—is the key."); Cárdenas et al., *supra* note 220, at 4 ("Because we can never rule out successful attacks, security engineering has recognized the importance of detection and response to attacks.").

required to meet the threshold of reasonable code safety. Because even diligent testing can miss trivial errors, a factfinder cannot rely on intrinsic attributes of a software error to determine whether it was avoidable with due care. For the same reason, it is difficult to explain why a manufacturer should have used an alternative code design, without committing unfair hindsight bias. Software liability is stuck on crash prevention.

Tort law overcame a similar quandary in the late 1960s with respect to manual car accidents. For many decades, automakers maintained that they could not prevent crashes and that safety standards were futile.[228] In a line of cases culminating in *Evans v. General Motors Corp.*,[229] car manufacturers successfully defended against products liability claims by arguing that collisions were not an "intended purpose" of driving, and therefore manufacturers owed no duty to make cars "accident-proof or fool-proof."[230] At that time, car safety innovations focused only on crash prevention technologies such as brakes, windshield wipers, and turn signals.[231] The only crash mitigation offered was lap belts, which were optionally installed and rarely worn.[232]

---

228. *See* NADER, *supra* note 22, at 3–4; O'CONNELL & MYERS, *supra* note 24, at 20–21.

229. 359 F.2d 822 (7th Cir. 1966).

230. *Id.* at 824–25 ("The intended purpose of an automobile does not include its participation in collisions with other objects, despite the manufacturer's ability to foresee the possibility that such collisions may occur. As defendant argues, the defendant also knows that its automobiles may be driven into bodies of water, but it is not suggested that defendant has a duty to equip them with pontoons."); *id.* at 827 n.3 (Kiley, J., dissenting) ("General Motors has argued here that it owed no duty to plaintiff . . . . [Because] the automobile is intended for travel, not colliding with other vehicles or things."); *see also* Harvey M. Sklaw, *"Second Collision" Liability: The Need for Uniformity*, 4 SETON HALL L. REV. 499, 508–16 (1973) (explaining the "intended purpose" argument as "attractive in its simplicity" and showing how cases reiterated it). *But see* Ralph Nader & Joseph A. Page, *Automobile Design and the Judicial Process*, 55 CALIF. L. REV. 645, 655–56 (1967) (attacking the *Evans* decision for "set[ting] the development of the common law of auto design back thirty years"); Recent Cases, *Torts—Liability of Maker of Chattel—Manufacturer Is Not Liable for Failure to Design "Crashworthy" Automobile*, 80 HARV. L. REV. 688, 689 (1967) (criticizing the *Evans* court's "excessively narrow assumption that the purpose of an automobile is solely to provide a means of transportation"); *cf.* NADER, *supra* note 22, at 129–31 (detailing manufacturers' knowledge of "obvious" structural weaknesses of the X-frame construction).

231. *See* LEMOV, *supra* note 23, at 6 ("'Collision avoidance' was the predominant safety principle during the first sixty years of the century." (citing JOHN D. GRAHAM, AUTO SAFETY: ASSESSING AMERICA'S PERFORMANCE 17 (1989))); Window-cleaning device, U.S. Patent No. 743,801 (filed June 18, 1903) (issued Nov. 10, 1903).

232. *See* LEMOV, *supra* note 23, at 60–63; MASHAW & HARFST, *supra* note 26; *infra* notes 281–282.

The crashworthy doctrine broke the impasse.[233] In 1968, the Eighth Circuit charted a bold new course in the watershed case, *Larsen v. General Motors Corp.*,[234] involving a head-on collision that caused the steering column and wheel to be thrust like a spear into the driver's skull.[235] Though it was clear the driver was at fault for causing the accident (the "first collision"), the court held the automaker responsible for injuries caused or enhanced by the steering column and wheel during the "second collision."[236] The *Larsen* court cited statistical data on the annual rate of accidents—which in 1966 had risen to 52,500 deaths and 1.9 million disabling injuries—and also that "[b]etween one-fourth and two-thirds of all vehicles manufactured are at sometime during their subsequent use involved in the tragedy of human injury and death."[237] Given the "statistically inevitable" nature of such crashes, the court held that car manufacturers owed a duty to minimize the injurious effects of such eventualities.[238] The intended purpose of a car was not merely to provide transportation, but to provide *reasonably safe* transportation consonant with the state of the art.[239] Manufacturers were not wholesale insurers, but neither were they wholly immune.

Change was not immediate,[240] but astonishingly quick for common law.[241] A mere decade later, the "intended purpose" reasoning of *Evans*

---

233. *See generally* NADER, *supra* note 22. Congress responded by enacting the National Traffic and Motor Vehicle Safety Act of 1966 and the Highway Safety Act of 1966. Christopher Jensen, *50 Years Ago, 'Unsafe at Any Speed' Shook the Auto World*, N.Y. TIMES, Nov. 27, 2015, at B3.

234. 391 F.2d 495 (8th Cir. 1968).

235. *Id.* at 497 & n.2.

236. *Id.* at 502; *see also* NADER, *supra* note 22, at 90 ("The most flagrant instrument of trauma . . . is the steering assembly. It caused approximately twenty per cent of the injuries in the data sample taken during the past decade. As would be expected, it is the driver who is most often injured by the steering assembly, either by being thrown forward into it or by being impaled on a ramming steering column.").

237. *Larsen*, 391 F.2d at 502 n.4, 505 n.8.

238. *Id*. at 502.

239. *Id*. at 503 ("The manufacturers are not insurers but should be held to a standard of reasonable care in design to provide a reasonably safe vehicle in which to travel."); *accord* Volkswagen of Am. v. Young, 321 A.2d 737 (Md. 1974).

240. Knippen v. Ford Motor Co., 546 F.2d 993, 997–98 (D.C. Cir. 1976) ("Commentators have been critical of the reasoning of *Evans*, but it nonetheless has its judicial adherents . . . . The modern trend of the case law and increasingly the weight of authority favors *Larsen's* extended scope of liability."); Frericks v. Gen. Motors Corp., 317 A.2d 494, 534–38 (Md. Ct. Spec. App. 1974) (tallying nine states and the District of Columbia that had chosen to follow *Larsen*, and ten states that had chosen to follow *Evans*); *see also* Yetter v. Rajeski, 364 F. Supp. 105 (D.N.J. 1973); McClung v. Ford Motor Co., 333 F. Supp. 17 (S.D. W. Va. 1971), *aff'd*, 472 F.2d 240 (4th Cir. 1973).

241. *See* Huff v. White Motor Corp., 565 F.2d 104, 110 (7th Cir. 1977) (overruling *Evans*).

was no longer being cited with approval.[242] *Larsen* became the unanimous rule across the nation.[243] The crashworthy doctrine encouraged quicker adoption of and further innovation in automobile safety technologies.[244] To be sure, some academics minimized the significance of the doctrine, crediting statements from automakers that each case was too one-of-a-kind and low-impact to affect broader car design trends.[245] But the reduced

---

242. *See* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 16, Reporters' Note to cmt. a (AM. LAW INST. 2016) ("In the early years of tort liability for defective product design, some courts refused to recognize a duty on the part of an auto manufacturer to design a reasonably crashworthy vehicle. The overwhelming majority, however, followed the view of Larsen v. General Motors Corp., which held that collisions are foreseeable and that manufacturers must design cars so that they are reasonably crashworthy. The *Larsen* rule appears now to be the unanimous position of American courts." (citations omitted)).

243. *See* Barry Levenstam & Daryl J. Lapp, *Plaintiff's Burden of Proving Enhanced Injury in Crashworthiness Cases: A Clash Worthy of Analysis*, 38 DE PAUL L. REV. 55, 61 n.33 (1989) (collecting early cases showing thirty-five states adopting *Larsen*). Eleven of the remaining fifteen states have since adopted the doctrine: Alabama—Gen. Motors Corp. v. Edwards, 482 So.2d 1176 (Ala. 1985); Alaska—Gen. Motors Corp. v. Farnsworth (Alaska 1998); Arizona—Cota v. Harley Davidson, Inc., 684 P.2d 888 (Ariz. Ct. App. 1984); Delaware—Gen. Motors Corp. v. Wolhar, 686 A.2d 170 (Del. 1996); Meekins v. Ford Motor Co., 699 A.2d 339 (Del. Sup. Ct. 1997); Hawaii—Holliday v. Bell Helicopters Textron, Inc., 747 F. Supp. 1396 (D. Haw. 1990); Mississippi—Tolive v. Gen. Motors Corp., 482 So.2d 213 (Miss. 1985); Nevada—Andrews v. Harley Davidson, Inc., 796 P.2d 1092 (Nev. 1990); New Hampshire—Trull v. Volkswagen of Am., Inc., 761 A.2d 477 (N.H. 2000); North Carolina—Warren v. Colombo, 377 S.E.2d 249 (N.C. Ct. App. 1989); Utah—Egbert v. Nissan North Am., Inc., 167 P.3d 1058 (Utah 2007); West Virginia—Blankenship v. Gen. Motors Corp., 406 S.E.2d 781 (W. Va. 1991). Three states have broached the question but remained noncommittal: Arkansas—Bishop v. Tariq, Inc., 384 S.W.3d 659 (Ark. Ct. App. 2011); Connecticut—Giannini v. Ford Motor Co., 616 F. Supp. 2d 219, 222 (D. Conn. 2007); Maine—Taylor v. Ford Motor Co., No. 06-69-BW, 2006 WL 2228973 (D. Me. Aug. 3, 2006). No decisions on point were found from Vermont.

244. *See* Johnson, *supra* note 56, at 685 ("The 'second collision' auto cases show the value of products liability litigation in improving industry customs.").

245. *See* MASHAW & HARFST, *supra* note 26, at 240–41 ("It is difficult to imagine that the products liability system is a major influence on the safe design of automobiles. The messages from the liability system to the manufacturers are both weak and full of static."); John D. Graham, *Product Liability and Motor Vehicle Safety*, *in* LIABILITY MAZE, *supra* note 74, 120, 183–84 (concluding that there is "little evidence that expanded product liability risk was necessary to achieve the safety improvements that have been made," because other effects including "consumer demand, regulation, and professional responsibility would have been sufficient to achieve improved safety," though acknowledging that "liability seemed to cause safety improvements to occur more quickly than they would have occurred in the absence of liability"); Peter W. Huber & Robert E. Litan, *Overview*, *in* LIABILITY MAZE, *supra* note 74, at 1, 5 ("[W]hile Eads and Reuter find from their interviews of corporate managers that product liability exerts a strong 'pro-safety' effect on product design, they also confess that current liability law sends an 'extremely vague signal,' since it does not indicate '*how* to be careful, or more important, *how* careful to be.'"). *But see* Peter L. Kahn, *Regulation and Simple Arithmetic: Shifting the Perspective on Tort Reform*, 72 N.C. L. REV. 1129, 1176–78 (1994) (arguing that the development of crashworthiness litigation had a greater impact on automobile safety than scholars gave it credit).

rate of repeat cases was a remarkable change from prior practices, when design flaws persisted lazily across multiple model years, and it suggests that lessons from crashworthy cases were learned more attentively than not.[246] The certitude of stare decisis helped steer automakers away from dashboard knobs, flimsy door latches, and rear-mounted engines, and toward padded interiors, sounder components, and collapsible steering columns and bumpers.[247] The annual rate of traffic deaths plummeted asymptotically from 5.5 per 100 million vehicle miles traveled in 1966, to 2.76 in 1982, 1.58 in 1998, and 1.18 in 2016.[248]

In short, the crashworthy doctrine *worked*.[249] It gave courts a dynamic framework that ratcheted incentives to reduce injuries and promote

---

246. *See* O'CONNELL & MYERS, *supra* note 24, at 160–61, 173–84 (observing that the automobile industry has become "panicked over the lawsuits filed against it" and that it has resulted in substantial reductions in "lead time for a completely changed car" from three years to two years or less); Johnson, *supra* note 56, at 677 ("Court decisions in these suits have played an active role in . . . providing incentives for manufacturers to improve products and thereby avert future litigable injuries."). There were exceptions to the rule. *See* Carl T. Bogus, *War on the Common Law: The Struggle at the Center of Products Liability*, 60 MO. L. REV. 1, 77–82 (1995) (narrating the multi-year sagas of the Ford Pinto and the GM side-saddle fuel tanks, in which automakers apparently determined the costs of litigation were worth enduring); *cf.* Graham, *supra* note 245, at 128–37 (offering a more sympathetic account that the Ford Pinto's performance was comparable to that of other subcompact and compact cars).

247. *See, e.g.*, Hancock v. Paccar, Inc., 283 N.W.2d 25 (Neb. 1979) (bumper); Huddell v. Levin, 537 F.2d 726 (3d Cir. 1976) (headrest); Jeng v. Witters, 452 F. Supp. 1349 (M.D. Pa. 1978) (car door); Schwartz, *supra* note 5 (discussing public fallout to *Grimshaw v. Ford Motor Co.*, 174 Cal. Rptr. 348 (Ct. App. 1981) (rear-mounted engine)); *see also* Federal Standard No. 515—Standard Safety Devices for Automotive Vehicles, 30 Fed. Reg. 8,319 (June 30, 1965) (complementing judicial doctrine with regulatory rulemaking).

248. NHTSA, U.S. DEP'T OF TRANSP., MOTOR VEHICLE TRAFFIC FATALITIES AND FATALITY RATES 1899–2016 (2018), https://cdan.nhtsa.gov/tsftables/Fatalities%20and%20Fatality%20Rates.pdf [https://perma.cc/TFQ4-XNRK]; Injury rates fell from 169 per 100 million vehicle miles traveled in 1988, to 79 in 2015. *Traffic Safety Facts Annual Report Tables*, NHTSA, https://cdan.nhtsa.gov/tsftables/tsfar.htm# [https://perma.cc/FBN6-FA9G] (click "Trends," then click "Trends: General," then follow "Table 2" hyperlink). *But see* EASTMAN, *supra* note 24, at 155 (noting that the switch from a fatality ratio based on the number of automobiles registered to one based on deaths per 100 million miles driven per year was done to present a more pleasing picture); NADER *supra* note 22, at 265 (criticizing "any claim of a reduced death rate per vehicle miles traveled" as "giv[ing] an illusion of progress which is definitely misleading"); NIDHI KALRA & SUSAN M. PADDOCK, RAND, DRIVING TO SAFETY: HOW MANY MILES OF DRIVING WOULD IT TAKE TO DEMONSTRATE AUTONOMOUS VEHICLE RELIABILITY? (2016); *supra* note 35 (questioning the credibility of traffic safety statistics).

249. To be sure, Ralph Nader and his allies worried greatly that the pace and scope of automotive safety improvements failed to meet expectations. RALPH NADER, UNSAFE AT ANY SPEED xxvii–lxxxvii (2d ed. 1972) [hereinafter NADER (2d ed.)]; Ralph Nader & Joseph A. Page, *Automobile-Design Liability and Compliance with Federal Standards*, 64 GEO. WASH. L. REV. 415 (1996); *see also* Marc Galanter, *Why the 'Haves' Come Out Ahead: Speculations on the Limits of Legal Change*, 9 LAW & SOC'Y REV. 95, 98–104 (1974) (introducing the theory of repeat player advantage in

innovation, without demanding perfection or bankrupting manufacturers.[250] Crashworthiness was less than strict enterprise liability, because it excused "unreasonable" measures, and offered remedy only when harms were preventable given the state of the art. Yet it also was more than pure negligence law, because it offered a legal solution to the problem of statistically inevitable injuries, not just proximately foreseeable ones.

Though the crashworthy doctrine remains a potent force, its domain has been limited to vehicles such as cars, motorcycles, boats, and aircraft, and equivalent specialized machinery such as farm tractors, grain harvesters, lawnmowers, and snowmobiles.[251] This Article argues crashworthiness should extend to code.

A. On the Origin of Crashworthiness

The crashworthy doctrine was invented as a response to judicial deadlock.[252] For decades, the Big Four automakers—General Motors, Ford, Chrysler, and American Motors—successfully argued that preventing car accidents was beyond their control. Drivers bore the brunt of the blame: they were speed demons, drunk drivers, young hot-rodders,

---

litigation). Nonetheless, what "worked" is that the crashworthy doctrine revealed to courts a new path forward they were willing to travel. *Cf.* Ronald Dworkin, *Hard Cases*, 88 HARV. L. REV. 1057, 1093–94, 1097–1101 (1975) (expounding on the "gravitational force" of common law precedent that constrains judicial decisions to arguments of principle rather than arguments of policy).

250. *Compare* Victor E. Schwartz & Leah Lorber, *The General Aviation Revitalization Act: How Rational Civil Justice Reform Revitalized an Industry*, 67 J. AIR L. & COM. 1269 (2002), *with* Nathan J. Rice, *The General Aviation Revitalization Act of 1994: A Ten-Year Retrospective*, 2004 WIS. L. REV. 945, 951 (2004).

251. *See* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 16, Reporters' Note to cmt. a (AM. LAW INST. 1998) (collecting cases). *But see* Scott G. Lindvall, *Aircraft Crashworthiness: Should the Courts Set the Standards?*, 27 WM. & MARY L. REV. 371 (1986) (finding courts reluctant to apply crashworthiness in cases involving aircraft). Some commentators advocated an expanded conception of "enhanced damages" to include any form of enhancement, but those calls do not appear to have been heeded. *See* Thomas V. Harris, *Enhanced Injury Theory: An Analytic Framework*, 62 N.C. L. REV. 643, 647–50 (1984).

252. *See* Peter L. Kahn, *Regulation and Simple Arithmetic: Shifting the Perspective on Tort Reform*, 72 N.C. L. REV. 1129, 1168–71 (1994) (describing the rise of the crashworthy doctrine and noting that "as of 1966, no plaintiff had yet prevailed on a claim that an automobile was defectively designed" (quoting Wood v. Gen. Motors Corp., 865 F.2d 395, 405 (1st Cir. 1988)); Nader & Page, *supra* note 230, at 645–46 ("On appeals, plaintiffs are batting zero. The appellate courts have yet to reverse a judgment for a manufacturer or affirm a judgment for a plaintiff in a case involving a traffic accident allegedly caused by the unsafe design of an American passenger car.").

old codgers, or too inept, undereducated, or otherwise careless.[253] Car manufacturers complained further that it was not their fault if passengers failed to purchase or use seat belts and other "extra" safety features.[254] When drivers were not at fault, road and weather conditions were the villain.[255] Automakers persuaded Congress to appropriate millions of dollars for the improvement of national highways.[256] Expensive proposals were developed to embed sensors, transmitters, and lights directly into the millions of miles of roadways, as well as to remove trees and all other obstacles within close striking distance from the road.[257]

At the same time, automakers resisted calls to impose any upfront restrictions on car design. The popular mantra then was that "safety

253. EASTMAN, *supra* note 24, at chs. 5 & 6; O'CONNELL & MYERS, *supra* note 24, at 67–87; LEMOV, *supra* note 23, at 59 ("Americans accepted the automobile industry and the safety establishment's repeated assertion: 'Cars are safe. Drivers cause accidents.' One could view it as a highly effective public brainwashing."); NADER, *supra* note 22, at 235–39 ("Today almost every program is aimed at the driver—at educating him, exhorting him, watching him, judging him, punishing him, compiling records about his driving violations, and organizing him in citizen support activities . . . . The reasoning behind this philosophy of safety can be summarized in this way: Most accidents are in the class of driver fault; driver fault is in the class of violated traffic laws; therefore, observance of traffic laws by drivers would eliminate most accidents."); Graham, *supra* note 34, at 1260 ("[T]here exists a tendency, in early accidents that involve a novel device, to focus on the behavior of its consumers . . . [and] regard[] early adopters as taking their chances with a technology."); Sam Peltzman, *The Regulation of Automobile Safety*, *in* AUTO SAFETY REGULATION: THE CURE OR THE PROBLEM? 1 (Henry L. Manne & Roger L. Miller eds., 1976); *see also* Schemel v. Gen. Motors Corp., 384 F.2d 802 (7th Cir. 1967); Schumard v. Gen. Motors Corp., 270 F. Supp. 311 (S.D. Ohio 1967). *But see* Frericks v. Gen. Motors Corp., 317 A.2d 494, 541 (Md. Ct. Spec. App. 1974) (Lowe, J., dissenting) ("More to the point, we think it is the vestige of an 'anachronism' based upon an era when motor cars were luxuries.").

254. *See infra* note 281; LEMOV, *supra* note 23, at 52; NADER (2d ed.), *supra* note 249, at xiii ("Ford officials right up to Henry Ford II perpetuated the myth that motorists would reject safer cars and that sales strategies and safety don't mix."); O'CONNELL & MYERS, *supra* note 24, at 155, 160.

255. COMMERCE CLEARING HOUSE, INC., SUPPLEMENT TO FED. CARRIERS REP. NO. 456, MOTOR VEHICLE AND HIGHWAY SAFETY ACTS OF 1966 WITH EXPLANATION 15 (1966) ("Poor roads, it was felt, imposed upon the driver demands of judgment, decision, and reaction that he could not possibly meet adequately in the few seconds he usually has in which to meet them."); EASTMAN, *supra* note 24, at 147–48; LEMOV, *supra* note 23, at 6–7; NADER, *supra* note 22, at 233; O'CONNELL & MYERS, *supra* note 24, at 6 ("The real answer, he went on to tell us, is to strip away trees for one hundred feet on both sides of the highway. That'll take care of the tree question.").

256. *See* Federal-Aid Highway Act of 1956, Pub. L. No. 84-627, 70 Stat. 374; Post Office Appropriations Act of 1922, Pub. L. No. 67-244 § 4, 42 Stat. 652, 660; Federal Highway Act of 1921, Pub. L. No. 67-87, 42 Stat. 212; Federal Aid Road Act of 1916, Pub. L. No. 64-156, 39 Stat. 355.

257. *See* O'CONNELL & MYERS, *supra* note 24, at 91–99. This faith in road-improvement efforts has carried into the present day. *See* Dean Narciso, *Smart Corridor Will Allow Cars to Talk to One Another*, COLUMBUS DISPATCH (June 16, 2018, 6:07 PM), https://www.dispatch.com/news/20180616/smart-corridor-will-allow-cars-to-talk-to-one-another [https://perma.cc/LV7L-2LWJ] (describing test efforts to build out vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technologies).

doesn't sell," a catchy slogan that united a diverse set of ideologies. One was a story about what consumers wanted: raw speed and horsepower, without sissy frills that cost extra.[258] The 1956 model year was often invoked as a cautionary tale, when Ford attempted an ill-fated campaign that featured safety as a prime selling point.[259] Thereafter automakers pointedly turned their back on safety to prove the point, launching a concerted push in the late 1950s and early 1960s to sell stripped-down subcompact cars that promised luxury at a discount.[260] Second, car sellers expressed fears that any mention of "safety" would discourage buyers by reminding them of the unpleasant hazards of driving. Accordingly, manufacturers played down the need for safety engineering, and instead employed stylists to play up the thrill and sex appeal of the road.[261] A third version lamented the exorbitant costs of developing and testing proper safety mechanisms, with one executive famously declaring that "it is completely unrealistic even to talk about a foolproof and crashproof car."[262] Rushing ahead with unproven technologies did nobody any favors,[263] least of all overeager drivers misled into a false sense of

---

258. *See* EASTMAN, *supra* note 24, at ch.4; GRAHAM, *supra* note 231, at 204–06 (describing "[t]he showroom reality" of slow sales of optional air bags despite marketing studies and opinion polls indicating substantial consumer support); *cf.* Gina M. DeDominicis, *No Duty at Any Speed?: Determining the Responsibility of the Automobile Manufacturer in Speed-Related Accidents*, 14 HOFSTRA L. REV. 403 (1986) (discussing at length *Schemel v. General Motors Corp.*, 261 F. Supp. 134 (S.D. Ind. 1966), *aff'd*, 384 F.2d 802 (7th Cir. 1967), which rejected a claim that automakers should not manufacture cars capable of attaining speeds in excess of 100 miles per hour).

259. GRAHAM, *supra* note 231, at 123. *But see* LEMOV, *supra* note 23, at 61; NADER (2d ed.), *supra* note 249, at ix.

260. LEMOV, *supra* note 23, at 179; MASHAW & HARFST, *supra* note 26, at 105 ("As Lee Iacocca later put it, the American people wanted an economy car, no matter what it cost."). *But see* GRAHAM, *supra* note 231, at 126–28 (crediting arguments by Iacocca that American automakers needed subcompacts to combat competition from the Volkswagen Beetle and other imports).

261. NADER, *supra* note 22, at 210–31; O'CONNELL & MYERS, *supra* note 24, at 145–57; John Sibley, *State Study Says Safety Car Would Cut Injuries*, N.Y. TIMES, Feb. 1, 1966, at 24 ("The [New York] State Department of Motor Vehicles added its voice today to the chorus of automobile industry critics who charge that Detroit is more concerned with styling than safety.").

262. NADER, *supra* note 22, at 3–4 (quoting John F. Gordon, the president of General Motors); *cf.* Larsen v. Gen. Motor Corp., 391 F.2d 493, 497–500 (collecting prior case law finding no manufacturer duty to design automobiles to be "'accident-proof' or 'fool-proof'" (quoting Evans v. Gen. Motors Corp., 359 F.2d 822 (7th Cir. 1966))).

263. *See Larsen*, 391 F.2d at 504 n.7 (quoting industry protestations to Congress that "it is always relatively easy to come up with a new design of an old part, or the design of a new feature or part, but until we are able to adequately test this part and have a pretty clear picture of what it will do under the circumstances to which it is subjected, we are exposing ourselves, the users of our products, and frequently others on the highways to risks"); Jensen, *supra* note 233, at B3 (describing criticism by some who "thought that Mr. Nader did not understand the complexity and trade-offs of automotive engineering and that [his] book encouraged people to sue the auto industry").

complacency.[264] Yet as accident rates grew steadily worse in the 1960s, the automakers squandered their enormous reservoir of public trust.[265]

The theory of "second collision" or "crashworthiness" grew out of concerted efforts by plaintiffs' attorneys and consumer advocates to point the finger back at car manufacturers and force them to make reasonable design accommodations for safety. Regardless of who or what caused the first collision, the causes of second collision were by definition within the aegis of the car designer. The concept of the "first collision" cleverly enfolded all possible external causes of the crash, including driver fault, obstacles, and environmental conditions. Defining the "second collision" as the impact between passengers and the interior of the car then isolated those factors that lay within automakers' control.[266] Nader and his allies worked methodically in court and out of court to prove that automakers had extensive knowledge of the prevalence of second-collision injuries, possessed readily available safety solutions that could minimize such injuries, and had conspired to withhold and suppress such safety measures.[267] These efforts culminated in landmark federal legislation establishing a new regulatory agency with authority to issue national safety standards.[268]

---

264. *See* Murray Mackay, *Liability, Safety and Innovation in the Automotive Industry*, *in* LIABILITY MAZE, *supra* note 74, at 191, 214–17 (describing "technological uncertainties" and the "threat of product liability" as prime reasons contributing to the industry's opposition to air bags); *cf.* Peltzman, *supra* note 253, at 29 (arguing that any benefits of auto safety regulation were offset by an increase in driver willingness to take risks); David Shepardson, *Fatal Tesla Autopilot Crash Driver Had Hands Off Wheel: U.S. Agency*, REUTERS, June 7, 2018, https://www.reuters.com/article/us-tesla-crash/fatal-tesla-autopilot-crash-driver-had-hands-off-wheel-us-agency-idUSKCN1J31VP [https://perma.cc/J65E-CBML].

265. *See* O'CONNELL & MYERS, *supra* note 24, at 163 (quoting Dan Cordtz, *Auto Executives Hurt Own Cause*, WALL ST. J., July 20, 1965, at 14).

266. *See* LEMOV, *supra* note 23, at 111–12 (describing efforts by Dr. William Haddon, the first traffic safety administrator, to "move the emphasis on primary causation away from Box A (the driver) and towards Box B (the vehicle)" where "the most substantial payoff was").

267. In particular, they pointed to the 1956 model year when Ford had touted safety measures as a selling point, and which had been received with great interest by consumers, before being bullied by General Motors to recant and fall back in line with the mantra that "safety doesn't sell." *See* NADER (2d ed.), *supra* note 249, at xi–xvi; EASTMAN, *supra* note 24, at 228–32.

268. *See generally* MASHAW & HARFST, *supra* note 26. President Johnson signed the National Traffic and Motor Vehicle Safety Act, Pub. L. No. 89-563, 80 Stat. 718 (1966), and the Highway Safety Act, Pub. L. No. 89-564, 80 Stat. 731 (1966), on September 9, 1966, shortly after *Evans* and shortly before *Larsen*. *See* Walter Rugaber, *Safety Council Sees Auto Law Saving 10,000 Lives*, N.Y. TIMES, Sept. 11, 1966, at 80. Crashworthiness dominated the legislative hearings. *See, e.g.*, 112 Cong. Rec. 14,221 (1966) ("The committee heard compelling testimony that passenger cars can be designed and constructed so as to afford substantial protection against the 'second collision' for both driver and passenger; further, that some of these design changes can be achieved at little or no additional manufacturing cost."). NHTSA was formed in follow-on legislation enacted in 1970. *See* Highway

More significantly, the "second collision" theory broke open a longstanding logjam in judicial decisionmaking, providing courts a forceful rebuttal against the claim that safety was too hard to engineer.[269] Building a crashproof car was impossible, but installing a collapsible steering column was not. Reframing the liability problem in this narrower manner proved remarkably robust against resistance from automakers.

In the decades since *Larsen*, the body of critical commentary has remained quite modest, converging on only two main points of contention. One set of issues addresses apportionment of damages based on the comparative fault of drivers and other third parties.[270] The other set of issues concerns how to designate the "state of the art" of safety technologies.

The apportionment discussion has centered primarily on who should have to bear the burden of proof. Initially, in a line of cases beginning with *Huddell v. Levin*,[271] courts held that the usual rule in litigation is that plaintiffs bear the burden to prove their cases, and that this rule should extend to proving whether one's injuries are attributable to the second collision as opposed to other factors.[272] In a concurring opinion, Judge Rosenn criticized this aspect of the decision as doing a "gross injustice to an innocent plaintiff" particularly where it is "impossible to apportion

---

Safety Act of 1970, Pub. L. No. 91-605, 84 Stat. 1713; Crane et al., *supra* note 64, at 302 (outlining NHTSA's regulatory authority).

269. *See, e.g.*, James A. Henderson, Jr., *Judicial Review of Manufacturers' Conscious Design Choices: The Limits of Adjudication*, 73 COLUM. L. REV. 1531 (1973) (questioning the institutional competence of courts to assess design decisions that are highly "polycentric"); Alden D. Holford, *Limits of Strict Liability for Product Design and Manufacture*, 52 TEX. L. REV. 81, 85, 91–92 (1973); Comment, *Automobile Design Liability:* Larsen v. General Motors *and Its Aftermath*, 118 U. PA. L. REV. 299, 303 (1969) ("Judicial hesitancy to hold automobile manufacturers liable for negligent design is attributable in part to misgivings about the jury's judgment on the issues of damage apportionment and the expert's standard of care.").

270. *See* James B. Sales, *Contribution and Indemnity Between Negligent and Strictly Liable Tortfeasors*, 12 ST. MARY'S L.J. 323 (1980); Thomas V. Harris, *Enhanced Injury Theory: An Analytic Framework*, 62 N.C. L. REV. 643 (1984).

271. 537 F.2d 726 (3d Cir. 1976).

272. *Id.* at 738 ("[T]he automobile manufacturer is liable only for the enhanced injuries attributable to the defective product. This being the essence of the liability, we cannot agree that the burden of proof on that issue can properly be placed on the defendant manufacturer."); *see also* Heather Fox Vickles & Michael E. Oldham, *Enhanced Injury Should Not Equal Enhanced Liability*, 36 S. TEX. L. REV. 417, 429–30, 430 n.75 (1995); Michael Hoenig, *Resolution of "Crashworthiness" Design Claims*, 55 ST. JOHN'S L. REV. 633, 699–706 (1981); Robert A. McConnell, S*urvey of Utah Strict Products Liability Law: From* Hahn *to the Present and Beyond*, 1992 B.Y.U. L. REV. 1173, 1196–1207 (1992).

damages among concurrent tortfeasors."[273] Judge Rosenn invoked the counter-principle that innocent victims should receive preference over concurrent wrongdoers.[274] The *Huddell* concurrence has become the majority rule after adoption by two influential opinions—*Fox v. Ford Motor Co.* [275] and *Mitchell v. Volkswagenwerk A.G.* [276]—as well as the Restatement.[277] Only a minority of states continue to follow *Huddell*.[278]

The apportionment discussion also seeks to restrict what kinds of plaintiff fault can be used to offset manufacturer fault for second collisions.[279] Typically, courts will allow evidence of intoxication or other censured behavior.[280] But blaming victims for failure to wear seatbelts has been viewed differently. Historically, seatbelt use was not common practice.[281] Automakers had stubbornly opposed regulatory efforts to

---

273. *Huddell*, 537 F.2d at 746 (Rosenn, J., concurring).

274. *Id.* (citing Summers v. Tice, 199 P.2d 1 (Cal. 1948)). The majority opinion disagreed with the characterization of defendants as concurrent tortfeasors. *Id.* at 738–39.

275. Fox v. Ford Motor Co., 575 F.2d 774 (10th Cir. 1978).

276. Mitchell v. Volkswagenwerk A.G., 669 F.2d 1199 (8th Cir. 1982).

277. *Id.*; *Fox*, 575 F.2d 774; RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 16 Reporters' Note to cmt. d (AM. LAW INST. 1998) (tallying twenty-three states favoring the *Fox-Mitchell* approach); *see also* Stanton Phillip Beck, *Enhanced Injury: A Direction for Washington*, 61 WASH. L. REV. 571 (1986); Karen L. Chadwick, *"Causing" Enhanced Injuries in Crashworthiness Cases*, 48 SYRACUSE L. REV. 1223 (1998); Gerald F. Tietz et al., *Crashworthiness and Erie: Determining State Law Regarding the Burden of Proving and Apportioning Damages*, 62 TEMPLE L. REV. 587, 619 n.270 (1989); Aaron D. Twerski, *Inside the Restatement*, 24 PEPP. L. REV. 839, 848–49 (1997).

278. *Compare* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 16, Reporters' Note to cmt. d (AM. LAW INST. 1998) ("Only six states are clearly in the *Huddell* camp: Michigan, New Mexico, New York, Pennsylvania, South Carolina, and Virginia."), *with* Levenstam & Lapp, *supra* note 243, at 66 n.61 (collecting cases from seven states: Iowa, New Jersey, New Mexico, New York, North Carolina, Pennsylvania, and South Carolina), *and* Vickles & Oldham, *supra* note 272, at 429 n.74, 442–43 (criticizing the Restatement's count and collecting cases from eighteen states: California, Colorado, Georgia, Indiana, Iowa, Kentucky, Louisiana, Missouri, New Jersey, New Mexico, New York, Nevada, North Carolina, Pennsylvania, South Carolina, Oregon, Virginia, and Washington).

279. *See* Ryan P. Harkins, *Holding Tortfeasors Accountable: Apportionment of Enhanced Injuries Under Washington's Comparative Fault Scheme*, 76 WASH. L. REV. 1185 (2001).

280. *See, e.g.*, West v. Bell Helicopter Textron, Inc., 967 F. Supp. 2d 479 (D.N.H. 2013) (cell phone use); Giannini v. Ford Motor Co., 616 F. Supp. 2d 219 (D. Conn. 2007) (alcohol consumption); Ellen M. Bublick, *The Tort-Proof Plaintiff: The Drunk in the Automobile, Crashworthiness Claims, and the Restatement (Third) of Torts*, 74 BROOK. L. REV. 707, 719–21 (2009); *see also* RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 16, Reporters' Note to cmt. f (AM. LAW INST. 1998) (finding courts "sharply split" with a majority "allow[ing] the introduction of plaintiff's conduct as comparative fault in a crashworthiness context"); Twerski, *supra* note 277, at 851–52 (explaining the Restatement members' vote to reverse the Reporters on this issue).

281. MASHAW & HARFST, *supra* note 26, at 85 (citing government studies finding seatbelt usage rates at 25% to 30%); Brian T. Bagley, *The Seat Belt Defense in Texas*, 35 ST. MARY'S L.J. 707, 716

require seatbelt installation in vehicles, which they characterized as an unnecessary expense that consumers did not want.[282] But as mandatory seatbelt laws suddenly swept across the nation in the late-1980s,[283] automakers switched tack and began arguing in court that those who failed to buckle up were lawbreakers who shared fault in enhancing their own injuries.[284] Many states took umbrage at this about-face and immediately banned use of this "seatbelt defense."[285] Accordingly, not all types of plaintiff fault are treated as equally culpable.

---

n.34 (2004); Robert F. Cochran, Jr., *New Seat Belt Defense Issues: The Impact of Air Bags and Mandatory Seat Belt Use Statutes on the Seat Belt Defense, and the Basis of Damage Reduction Under the Seat Belt Defense*, 73 MINN. L. REV. 1369, 1387–88 (1989) (below 15%).

282. EASTMAN, *supra* note 24, at 186 (noting general consensus among sales departments that "the presence of safety belts would imply that the automobile was dangerous"); *id.* at 226–28, 231 ("The demand for seatbelts caught [Ford] by surprise"); NADER, *supra* note 22, at 112–28; O'CONNELL & MYERS, *supra* note 24, at 193–98 (documenting New York state senator Edward Speno's fight to have seat belts installed in new cars, which was "bitterly opposed by Detroit"); *see also* Thomas F. Powell, II, *Products Liability and Optional Safety Equipment—Who Knows More?*, 73 NEB. L. REV. 843 (1994).

283. *See* GRAHAM, *supra* note 231, at 174–91, 222–24; MASHAW & HARFST, *supra* note 26, at 211 (explaining the background involving NHTSA's Federal Motor Vehicle Safety Standard No. 208 that led to sudden enactment of mandatory seatbelt use laws across the country); Cochran, *supra* note 281, at 1378 & n.31 ("Within the last few years, twenty-nine states and the District of Columbia have enacted statutes requiring seat belt use."); *State Seat Belt Law Takes Effect Today*, N.Y. TIMES, Dec. 1, 1984, at 26 (reporting New York's law as "the first in the nation").

284. *See* Kelly Carbetta-Scandy, *Litigating Enhanced Injury Cases: Complex Issues, Empty Precedents, and Unpredictable Results*, 54 U. CIN. L. REV. 1257, 1283–84 (1986) (noting that "Dean Prosser and the Restatement (Second) of Torts both approve of apportioning damages for a claimant's prior negligence in a crashworthiness case" and that "[r]ecent cases illustrate the trend toward allowing a 'mitigation rule' in crashworthiness cases to reduce a claimant's award proportionately by that amount attributable to the claimant's failure to use available safety restraints"); David A. Westenberg, *Buckle Up or Pay: The Emerging Safety Belt Defense*, 20 SUFFOLK U. L. REV. 867, 880–81 (1986) ("Since 1984, seven jurisdictions have adopted the safety belt defense in personal injury cases through judicial action and an additional four states have enacted the defense through legislative action, while only one new appellate court has rejected it.").

285. RESTATEMENT (THIRD) OF TORTS: PRODUCTS LIABILITY § 16, Reporters' Note to cmt. f (AM. LAW INST. 1998) (counting thirty-one states and the District of Columbia as having enacted statutory bars against the seatbelt defense); *id.* Reporters' Note to cmt. d (counting seven states as allowing the seatbelt defense in full, six states as permitting it with a cap on reduction in damages, and two states that leave the question to common law); Bagley, *supra* note 281, at 722–24 (noting that "a majority of states still reject a general admission of seat belt evidence"); Cochran, *supra* note 281, at 1387–88, 1401–04; Michael B. Gallub, *A Compromise Between Mitigation and Comparative Fault?: A Critical Assessment of the Seat Belt Controversy and a Proposal for Reform*, 14 HOFSTRA L. REV. 319, 334–38 (1986); Juli Spector, *The Continuing Controversy of the Seatbelt Defense*, 27 HOUSTON L. REV. 179, 180–81 (1990); Westenberg, *supra* note 284, at 904 (finding support in ten states); *see also* Bagley, *supra* note 281, at 719–20 (pointing also to the role of contributory negligence schemes: "Entirely denying compensation to an auto accident victim because he or she failed to use a seat belt was simply too harsh.").

The second, more challenging set of issues raised in the crashworthiness literature goes to the reasonableness of demanding safety performance that exceeds the state of the art. Manufacturers raise a valid objection that they cannot be held responsible for the impossible.[286] This concern is especially heightened for safety features, where consumers are less forgiving of errors.[287] For example, the introduction of collapsible steering columns undoubtedly saved many lives, but also invited many lawsuits for not being more perfect.[288] At the same time, the sordid history of auto safety regulation strongly suggests that some form of technology-forcing mechanism is needed when the financial incentives to delay and cheat on safety are too great.[289]

Here, the judicial response has been equivocal. On one hand, courts have consistently rejected efforts to rigidly define the "state-of-the-art" in narrow terms such as industry consensus[290] or regulatory compliance.[291]

---

286. *See* James A. Henderson, Jr., *Judicial Review of Manufacturers' Conscious Design Choices: The Limits of Adjudication*, 73 COLUM. L. REV. 1531 (1973) (repurposing Lon Fuller's concept of "polycentricity" to critique judicial second-guessing of safety design decisions); Aaron D. Twerski, *Seizing the Middle Ground Between Rules and Standards in Design Defect Litigation: Advancing Directed Verdict Practice in the Law of Torts*, 57 N.Y.U. L. REV. 521, 556–61 (1982) (describing the "state of the art" objection as encompassing three related concerns: (1) practical feasibility, (2) after-arising technology, and (3) shifts in societal norms).

287. *See* Jonathan J. Koehler & Andrew D. Gershoff, *Betrayal Aversion: When Agents of Protection Become Agents of Harm*, 90 ORG. BEHAV. & HUM. DECISION PROCESSES 245, 251 (2003); Timothy Wilton, *Federalism Issues in "No Airbag" Tort Claims: Preemption and Reciprocal Comity*, 61 NOTRE DAME L. REV. 1, 6 (1986) (noting "surprising" findings from a 1984 NHTSA study that "airbags by themselves are effective only in frontal collisions" and will not be activated in rear or side impact collisions or rollovers).

288. *See, e.g.*, Fouche v. Chrysler Motors Corp., 692 P.2d 345 (Idaho 1984) (insufficient collapse); Durett v. Baxter Chrysler-Plymouth, Inc., 253 N.W.2d 37 (Neb. 1977) (breach of warranty); Gen. Motors Corp. v. Howard, 244 So.2d 726 (Miss. 1971) (failure to telescope); *see also* John D. Morris, *Despite Progress in Auto Safety, Future Effectiveness of Federal Program Is in Doubt*, N.Y. TIMES, July 13, 1970, at 16 (explaining that the collapsible steering column was the only innovative safety standard the industry had put into effect, and that it "had been voluntarily incorporated on some General Motors models before being required under the 1966 [federal] safety standard").

289. *See* Nader & Page, *supra* note 249, at 457–58 (critiquing the regulatory compliance approach for being (1) primarily reliant on industry-supplied data, (2) subject to political interference, and (3) vulnerable to lethargic evolution or freezing of existing standards); Comment, *Automobile Design Liability:* Larsen v. General Motors *and Its Aftermath*, 118 U. PA. L. REV. 299, 310–11 (1969).

290. Hancock v. Paccar, Inc., 283 N.W.2d 25, 35 (Neb. 1979) ("Obviously, the inaction of all the manufacturers in an area should not be the standard by which the state of the art should be determined . . . . The question therefore is not whether anyone else was doing more, although that may be considered, but whether the evidence disclosed that anything more could reasonably and economically be done." (citing The T.J. Hooper, 60 F.2d 737 (2d Cir. 1932) (Learned Hand, J.))); Johnson, *supra* note 56, at 680–84.

291. Steven L. Holley, *The Relationship Between Federal Standards and Litigation in the Control of Automobile Design*, 57 N.Y.U. L. REV. 804, 818–25 (1982) (canvassing multiple arguments in

Such evidence can be relevant but never determinative—a sensible approach given that the "state-of-the-art" inquiry is ultimately one of technological fact, not of business judgment or legal decree. On the other hand, the bitter battle over passive airbags offers a cautionary tale where technological consensus is sharply divided.[292] After decades of back-and-forth wrangling led to a fragile truce between automakers and federal regulators, courts refused to disturb the peace.[293] Instead they dismissed "no airbag" lawsuits as preempted by federal law—a ruling ultimately ratified by the Supreme Court.[294] That judicial reticence illustrates the challenge of evaluating the adequacy and readiness of technology that has not yet been commercialized.[295] Since airbags became mandatory in 1997—more than four decades after their invention in 1953—deployment has been marred by sweeping recalls of tens of millions of devices,

---

support of the venerable principle that "compliance with government standards is not a complete defense, but only some evidence of due care"); Johnson, *supra* note 56, at 687–89; *see also* Mark A. Geistfeld, *Tort Law in the Age of Statutes*, 99 IOWA L. REV. 957 (2014) (exploring the interplay between negligence per se, the regulatory compliance defense, and statutory preemption); Catherine M. Sharkey, *Inside Agency Preemption*, 110 MICH. L. REV. 521, 532–45 (2012). *But see* Dana P. Babb, Note, *The Deployment of Car Manufacturers into a Sea of Product Liability? Recharacterizing Preemption as a Federal Regulatory Compliance Defense in Airbag Litigation*, 75 WASH. U. L.Q. 1677 (1997). Others have made the point that highly complex technologies such as aviation may be better suited for the regulatory compliance defense. *See* Scott G. Lindvall, *Aircraft Crashworthiness: Should the Courts Set the Standards?*, 27 WM. & MARY L. REV. 371 (1986); Patrick J. Shea, *Solving America's General Aviation Crisis: The Advantages of Federal Preemption over Tort Reform*, 80 CORNELL L. REV. 747 (1995).

292. *See* LEMOV, *supra* note 23, at 153 (noting opposition from automakers who called claims of airbag reliability and safety "preposterous"); MASHAW & HARFST, *supra* note 26, at 208–10, 213 (explaining that "the [U.S. Supreme] Court was baffled by NHTSA's apparent but unexplained abandonment of airbags . . . , devices that the agency had maintained for over a decade were technologically available and cost-beneficial").

293. *See* MASHAW & HARFST, *supra* note 26, at 184–87, 205–23; Nader & Page, *supra* note 249, at 434–52 & n.161.

294. *Compare* Geier v. Am. Honda Motor Co., 529 U.S. 861, 866 (2000) (finding NHTSA's Federal Motor Vehicle Safety Standard (FMVSS) 208 preempts "no airbag" claim), *with* Motor Vehicle Mfgrs. Ass'n v. State Farm Mut. Auto. Ins. Co., 463 U.S. 29, 46 (1983) (finding NHTSA's proposed rescission of FMVSS 208 arbitrary and capricious). *But cf.* Williamson v. Mazda Motor of Am., Inc., 562 U.S. 323 (2011) (finding FMVSS 208 does not preempt "no rear lap-and-shoulder seatbelt" claim).

295. *See* JOHN D. GRAHAM, AUTO SAFETY: ASSESSING AMERICA'S PERFORMANCE (1989) (arguing that market forces are faster and more effective than technology-forcing regulations). *Compare* Ted Sichelman, *Commercializing Patents*, 62 STAN. L. REV. 341 (2010) (describing the substantial challenges that lie between describing an invention and commercializing it), *with* NADER, *supra* note 22, at 75–77 (using patent filings to establish knowledge and deliberate disregard of safety risks).

sparked by just a few scattered reports of eye injuries, burns, and fatalities.[296]

All that said, the crashworthy doctrine was an ingenious innovation that has promoted remarkable progress in an area long mired in inaction. Breaking down the collision event into smaller sub-components refocused the cost-optimization function from crash prevention to crash mitigation, which has made all the difference. For consumer advocates, crashworthiness provided a long-sought way to push manufacturers off the perch of inaction. For manufacturers, crashworthiness offered an alternative to total enterprise liability for all car accidents. Courts' willingness to draw limits based on apportionment of fault and technological state-of-the-art showed that the doctrine is a rule of reason, not one of per se liability. Importantly, crashworthiness does not demand that every crash must result in no harm.

B. Software Fault Tolerance: Translation from Cars to Code

The basic lesson of the crashworthy doctrine is simple: "inevitable" crashes that could not have been prevented in their entirety can nonetheless be played back in slow-motion and partitioned into smaller chunks more amenable to legal treatment. In the automotive context, that process of subdivision has yielded a "first collision" and a "second collision," with automakers bearing legal responsibility for one but not the other. The "first collision" collects all the causes leading up to the accident and sets those aside; the "second collision" trains judicial attention instead on the window of opportunity following the moment of impact, where substandard design decisions enhance (or fail to soften) the severity of the crash.

As this Article argues, that same process of slow-motion diffraction can—and should—be extended to cyber-physical systems. Where a software error leads to physical injury or death, the initial activation of that software error can be partitioned from the subsequent software failure—which in turn can be partitioned from the physical crash that follows after that. To be sure, crashworthiness in the *cyber* sense is different from crashworthiness in the *physical* sense. No literal collisions are involved in code crashes, so the physics of cause-and-effect are

296. *See* Myron Levin, *Air Bag Lawsuits Blame Nissan for Eye Injuries*, L.A. TIMES, Nov. 18, 2002, at C4 (reporting 215 deaths since 1990, and 1.2 million vehicles recalled just in 2002, because of airbag defects); Hiroko Tabuchi, *The Quest to Save a Few Dollars Per Airbag Led to a Deadly Crisis*, N.Y. TIMES, Aug. 27, 2016, at A1 (reporting fourteen deaths and more than 100 injuries due to defective ammonium nitrate inflators, leading to recalls of 64 million airbags).

determined not by Newtonian momentum and inertia but by code and data. The provenance and progression of code crashes shares little in common with the straightforward wham-bam sequence of automotive crashes. Nonetheless, there are analogous opportunities for cyber-physical manufacturers to use safer designs that can mitigate the effects of a software error between the onset and the end of a code crash event.[297]

Within the computer science literature, there are two basic approaches to software dependability: fault avoidance and fault tolerance.[298] (To be clear, use of the term "fault" in the computer science sense is not a *legal* assignment of liability but merely a *factual* proposition that an error exists in the system.) This bifurcation between avoidance and tolerance parallels the division between first and second collisions: in both contexts, the split represents the conceptual pivot point between a manufacturer's duty to design a system that runs safely *before* a crash event and its duty to design a system that runs safely *after* a crash event has begun.

Fault avoidance seeks to make software "foolproof or crashproof" by averting errors from the outset, at the design and build stages. Lay discussions of software liability often begin and end with fault avoidance. As recited earlier, however, perfect fault avoidance is effectively unachievable.[299] The most rigorous strategies—such as formal methods and model checking—can validate small, limited modules of code. But because the extra overhead required for this level of perfection is impracticable for commercial development, these formal constraints are often relaxed, used sparingly, or omitted entirely, even in safety-critical

297. *See* Jean-Claude Laprie, *Dependable Computing and Fault Tolerance: Concepts and Terminology*, 15 PROC. IEEE INT'L SYMP. ON FAULT-TOLERANT COMPUTING 2 (1985) (distinguishing between faults that create latent errors, and system failures resulting from activation of those latent errors); Ang Chen et al., *Dispersing Asymmetric DDoS Attacks with SplitStack*, 15 PROC. ACM WORKSHOP ON HOT TOPICS IN NETWORKS 197 (2016) (observing that existing responses to distributed-denial-of-service attacks "primarily focus on stopping the attack traffic as early as possible," either at the source, in the network, or at the end hosts).

298. PETER ALAN LEE & THOMAS ANDERSON, FAULT TOLERANCE: PRINCIPLES AND PRACTICE 4–8 (Springer-Verlag/Wien 1990) (1981) ("What is surprising is that, until recently, tolerance for software faults has not been advocated and that almost all software research has been applied to chasing the elusive goal of producing perfect software."); PULLUM, *supra* note 30, at 7–13; Laprie, *supra* note 297, at 3; *see also* WILFREDO TORRES-POMALES, SOFTWARE FAULT TOLERANCE: A TUTORIAL, NASA 6–7 (2000) (describing two methods of dealing with software faults on the front end (fault prevention and fault tolerance) as well as two methods on the back end (fault removal and workarounds)).

299. *See supra* Section III.C.

systems.[300] Compromising on *ex ante* correctness, of course, implies that some errors will slip through despite best efforts. To compensate, software developers lean heavily on *ex post* error removal strategies: testing the code as much as is economically feasible, and fixing any errors discovered thereby. But a well-known truism of software assurance is that testing cannot prove the absence of errors, only their presence.[301] Thus it is inevitable that all commercial software is shipped with latent errors.

Fault tolerance picks up where fault avoidance leaves off and attempts to minimize the likelihood that latent errors will lead to system failures. Within this literature, "faults" and "errors" are defined as distinct from "failures," to emphasize the point that faults need not lead immediately or inevitably to failures.[302] Ordinarily, a fault might generate an error which in turn might lead to failure;[303] but that fault is tolerated when the error is detected in time and failure is thereby averted or minimized.

A vivid illustration, borrowed from the physical world, is the concept of the "flight envelope."[304] When an airplane loses its engines, it does not immediately fall out of the sky. Instead, momentum continues to carry the machine forward without noticeable change for some period of time. This margin of error (the "flight envelope") can be computed using the altitude and speed at which the airplane is traveling. If the engines recover while the plane remains inside the flight envelope, the flight can continue undisturbed. Alternatively, if engine recovery is not an option, the pilot

300. *See* PULLUM, *supra* note 30, at 8–9 (observing that "formal methods have not been generally used on large projects" due to difficulty and overhead, but suggesting that formal methods might be usable on a specific part of a system to handle risk mitigation if that component were "small enough").

301. *See id.* at 11 ("Testing has its problems, too, and these should be kept in mind: it is not currently possible to exhaustively test a large, complex system; testing can show the presence, but not the absence of faults; it may be impossible to test under realistic conditions; and specification errors may not be visible until the system is used under operational conditions.").

302. *See id.* at 3–4 ("A *fault* is the identified or hypothesized cause of an error, sometimes called a 'bug.' . . . An *error* is part of the system state that is liable to lead to a failure. It can be unrecognized as an error (i.e., latent) or detected . . . . A *failure* occurs when the service delivered by the system deviates from the specified service, otherwise termed an incorrect result.").

303. *See* Laprie, *supra* note 297, at 4 ("[A] programmer's mistake is a *fault*: the consequence is a (*latent*) *error* in the written software (erroneous instruction or piece of data); upon *activation* . . . the error becomes *effective*; when this effective error produces erroneous data (in value or in the timing of their delivery) which affect the delivered service, a *failure* occurs.").

304. Ang Chen et al., *Fault Tolerance and the Five-Second Rule*, *in* 15 PROC. USENIX CONF. ON HOT TOPICS IN OPERATING SYS. 11 (2015) (using the flight envelope concept to argue that "allowing small mistakes could also be a useful approach to fault tolerance in distributed systems").

has some time to plan a more graceful failure mode, such as a water landing or a cockpit ejection.[305]

To be clear, this "crashworthy code" doctrine would constitute a new, additional theory of liability; it would not preempt any existing tort rules. For example, suppressing knowledge of an available bug fix could result in separate charges for failure of the duty to warn, even if the code satisfies the duty of crashworthiness.[306] Likewise, an autonomous car would still be subject to the same safety standards as a manual car, including negligence, products liability, and conventional crashworthiness—but the autonomous car would be expected further to meet a standard of code crashworthiness. That duty could be fulfilled by building in an adequate level of software fault tolerance.

1. Redundancy

Software fault tolerance consists of three key elements: (1) redundancies; (2) adjudication methods; and (3) recovery modes. Redundancy is the basic building block of any fault-tolerant design.[307] In the physical world, redundant design is ubiquitous and readily identifiable. Eighteen-wheeler trucks are better equipped to handle a flat tire than two-wheel motorcycles; twin-engine planes are safer than single-engine planes; server farms are less likely to suffer data loss than personal home computers. Redundancy need not be identical; devices such as seatbelts and airbags add redundancy by offering complementary forms of crash protection. Today, many cyber-physical manufacturers already tout hardware redundancies—such as diverse arrays of sensors, spare batteries, and backup engines—as markers of their commitment to safety.

---

305. *See* Robert D. McFadden, *All 155 Aboard Safe as Crippled Jet Crash-Lands in Hudson*, N.Y. TIMES, Jan. 16, 2009, at A1; Christine Negroni, *A 1956 Version of Landing an Airplane on Water*, N.Y. TIMES, Nov. 9, 2017, at A18; *cf.* RICHARD H. GRAHAM, FLYING THE SR-71 BLACKBIRD 43–48 (2008) (narrating the survival of test pilot Bill Weaver); NADER, *supra* note 22, at 81–86 (narrating the survival of air cadet Hugh De Haven).

306. *See* Andrea M. Matwyshyn, *Hidden Engines of Destruction: The Reasonable Expectation of Code Safety and the Duty to Warn in Digital Products*, 62 FLA. L. REV. 109 (2010).

307. *See* Brian Randell, *System Structure for Software Fault Tolerance*, SE-l IEEE TRANSACTIONS ON SOFTWARE ENG'G, at 2 (1975) ("All fault tolerance must be based on the provision of useful redundancy, both for error detection and error recovery. In software the redundancy required is not simple replication of programs but redundancy of design.").

Unlike hardware redundancies, software redundancies remain uncommon.[308] The main hurdle is that true redundancy of code is costly to build, because it requires more than simply duplicating extra instances of the same code.[309] Hardware components can be doubled up because wear-and-tear occurs at variable rates, but each copy of software is exactly identical.[310] Extra copies of software will replicate the same errors and fail in precisely the same way given the same inputs.[311] This identity is a hard-won feature of software architecture, which strives to conceal or abstract away the "machine" layer as much as possible in order to ensure that random quirks across different machines do not corrupt software execution.[312] This abstraction is useful because it allows software engineers to build highly reproducible systems. It also means that effective software redundancy depends on injecting artificial diversity back into the system—in a carefully planned manner.[313]

Two accepted ways to introduce diversity into a software system are design diversity and data diversity. The first involves implementing the same task multiple times in multiple ways—using different algorithms, different programming languages, or different computing environments.[314] Ideally, each implementation is sufficiently independent

---

308. *See* Ellis F. Hitt & Dennis Mulcare, *Fault-Tolerant Avionics*, *in* AVIONICS DEVELOPMENT IMPLEMENTATION 8-1, 8-11 (Cary R. Spitzer ed., 2d ed. 2007) (noting in the avionics context that "[i]n general, much of this redundancy resides in additional hardware components").

309. *See* Jie Xu et al., *Dynamic Adjustment of Dependability and Efficiency in Fault-Tolerant Software*, *in* PREDICTABLY DEPENDABLE COMPUTING SYSTEMS 155 (Brian Randell et al. eds., 1995) (summarizing space and time overheads of software fault tolerance techniques); PULLUM, *supra* note 30, at 73 (collecting experimental studies finding that "the cost of threefold diversity . . . is not three times that of a single development (it is less) and the cost of twofold diversity is less than twice that of a single development").

310. *See* LEE & ANDERSON, *supra* note 298, at 62–63 (observing that physical fault tolerance assumes that "failures will occur independently in independent replicated components" but that "software systems do not wear out").

311. PULLUM, *supra* note 30, at 18–19 ("If the same software is copied and a failure occurs in one of the software replicas, that failure will also occur in the other replicas and there will be no way to detect the problem. (This assumes the same inputs are provided to each copy.)").

312. *See, e.g.*, Lee, *supra* note 192, at 2 (noting that digital circuit designers have "learned to harness intrinsically stochastic processes (the motions of electrons) to deliver a precision and reliability that is unprecedented in the history of human innovation"); Chisnall, *supra* note 192, at 44 (explaining that even the C language, which is considered "close to the metal," relies on substantial abstractions from the physical machine); *see also* Clark, *supra* note 192, at 109; SHNEIDMAN ET AL., *supra* note 192. *See generally* BACH, *supra* note 192.

313. PULLUM, *supra* note 30, at 25; Randell, *supra* note 307.

314. The two original design-diverse schemes are "recovery blocks" and "N-version programming." *See* J.J. Horning et al., *A Program Structure for Error Detection and Recovery*, *in* 16 LECTURE NOTES IN COMPUTER SCIENCE 171 (G. Goos & J. Hartmanis eds., 1974); L. Chen &

to minimize the likelihood of identical error causes. A prominent example is the Airbus A320 flight control system, introduced in the late-1980s, which uses two versions of the same software running simultaneously on independent computers manufactured by separate companies and having distinct functional specifications.[315] Because each variant is expected to produce the same behavior, any discrepancy indicates a fault has been detected.

Data diversity operates on a similar principle, except that the variants are generated by altering the input data rather than the program code.[316] Again, each data variant must be non-identical yet logically equivalent, such that a discrepancy in result signals the presence of a fault, rather than a valid difference. Sensor "fusion" offers a real-world application where data can be collected and combined from multiple sensor devices to provide useful diversity for fault tolerance purposes.[317] Distributed computing systems offer another important use case for cyber-physical manufacturers, where each node can be running identical code but receiving sufficiently proximate input to generate robust data redundancy.[318] In these examples, the method works when the data is generated independently, yet is expected to be reasonably consistent; it does not work if the diverse data sources are uncorrelated.

Algirdas Avizienis, *N-Version Programming: A Fault-Tolerance Approach to Reliability of Software Operation*, 8 PROC. ANN. IEEE INT'L CONF. ON FAULT-TOLERANT COMPUTING 3 (1978). Subsequent variations explored within the literature include distributed recovery blocks, consensus recovery blocks, N self-checking programming, and acceptance voting. See PULLUM, *supra* note 30, at 106, 132–72.

315. *See* Dominique Brière & Pascal Traverse, *AIRBUS A320/A330/A340 Electrical Flight Controls: A Family of Fault-Tolerant Systems*, 23 PROC. IEEE INT'L SYMP. ON FAULT-TOLERANT COMPUTING 616 (1993); Hitt & Mulcare, *supra* note 308, at 8-8 to 8-9 (describing design diversity in the flight control software for various aircraft manufactured by Airbus, Boeing, Lockheed, and McDonnell Douglas).

316. *See* P.E. Ammann & J.C. Knight, *Data Diversity: An Approach to Software Fault Tolerance*, 37 IEEE TRANSACTIONS ON COMPUTERS 418 (1988); P.E. Ammann, *Data Redundancy for the Detection and Tolerance of Software Faults*, 22 PROC. SYMP. ON THE INTERFACE 43 (1990). Temporal diversity, which alters time as the data input, is sometimes treated as its own category of software diversity. *See* D.J. Martin, *Dissimilar Software in High Integrity Applications in Flight Control*, *in* SOFTWARE FOR AVIONICS, AGARD CONFERENCE PROCEEDINGS, NATO (1982).

317. *See* Radoslav Ivanov et al., *Attack-Resilient Sensor Fusion for Safety-Critical Cyber-Physical Systems*, 15 ACM TRANSACTIONS ON EMBEDDED COMPUTING SYS., Feb. 2016, at 21-1, 21-2.

318. *See* Jiaxing Zhang et al., *SIROM3 – A Scalable Intelligent Roaming Multi-Modal Multi-Sensor Framework*, 38 IEEE ANN. COMPUTER SOFTWARE & APPLICATIONS CONF. 446 (2014).

2. Adjudication

Once a discrepancy has been detected, the system must decide which variants are true and which are false. Two general categories of adjudicatory methods exist: acceptance tests and voting algorithms. Acceptance tests offer a sanity check on whether an output is within an appropriate range or is otherwise reasonable. For example, if a flight control system computes an airspeed value that is impossible given the structural capabilities of the aircraft, then it is immediately evident that something must be wrong with the sensor, the computer, or the aircraft.[319] Likewise, in a networked environment, if a request does not receive any response within a preset time period, returning a "time out" error is a very common use of an acceptance test to avoid an undesirable wait. Thus, acceptance tests offer a quick way to check and weed out certain kinds of invalid results. But they are ineffective at adjudicating between equally plausible results.

When the discrepancy is less easily resolved, a voting mechanism is needed to move forward. A broad selection of choices among election protocols offers different advantages and tradeoffs.[320] One important consideration involves how to define the passing threshold. The simplest scheme is a majority count.[321] More nuanced variations have experimented with plurality voting, weighted average voting, predictive voting, as well as hybrid voting schemes, in order to optimize the likelihood of achieving a correct result depending on starting assumptions about how a system could be attacked or compromised.[322]

---

319. *See* H. Hecht & M. Hecht, *Fault-Tolerant Software*, *in* FAULT-TOLERANT COMPUTING: THEORY AND TECHNIQUES 658 (D.K. Pradhan ed., 1986).

320. *See* G. Latif-Shabgahi et al., *A Taxonomy for Software Voting Algorithms Used in Safety-Critical Systems*, 53 IEEE TRANSACTIONS ON RELIABILITY 319 (2004); Paul R. Lorczak et al., *A Theoretical Investigation of Generalized Voters for Redundant Systems*, 19 INT'L SYMP. ON FAULT-TOLERANT COMPUTING 444 (1989); Behrooz Parhami, *Voting Algorithms*, 43 IEEE TRANSACTIONS ON RELIABILITY 617 (1994); Behrooz Parhami, *A Taxonomy of Voting Schemes for Data Fusion and Dependable Computation*, 52 RELIABLE ENG'G & SYS. SAFETY 139 (1996); *cf.* Richard H. Pildes & Elizabeth S. Anderson, *Slinging Arrows at Democracy: Social Choice Theory, Value Pluralism, and Democratic Politics*, 90 COLUM. L. REV. 2121 (1990) (applying Arrow's Impossibility Theorem to argue that there is no perfect voting system).

321. *See* R. B. Broen, *New Voters for Redundant Systems*, 97 J. DYNAMIC SYS. MEASUREMENT & CONTROL 41 (1975).

322. *See* Latif-Shabgahi, *supra* note 320, at 322–25; *cf.* Lani Guinier, *No Two Seats: The Elusive Quest for Political Equality*, 77 VA. L. REV. 1413 (1991) (exploring the use of pooling to overcome the problem of minority vote dilution in winner-take-all voting systems).

A second set of issues concerns which voters are eligible to be counted. Just as vote fraud and vote suppression are persistent fears in political elections, cyber-physical elections also raise similar concerns. For example, a common problem for distributed systems is how to authenticate messages transmitted remotely over the network. When a system is expecting a response but does not receive one, it could have a benign explanation such as network latency or temporary glitch, or it could be due to a malignant cause such as component failure or hostile attack. Conversely, a response that is properly received could be a spoofed message that should be distrusted. At one extreme, a perfectly naïve environment will always allow all possible voters. At the opposite end, a perfectly paranoid model will be quick to exclude voters, and even examine voting patterns for evidence of collusion to deceive the system.[323] In most real-world systems, regular attacks can be expected so some method of quarantine is warranted, but finding the right calibration is tricky because a system that is too quick to disable itself is unusable.

Third, any adjudication scheme in a safety-critical system must grapple with the element of time.[324] Communication is often unreliable in networked environments, which means responses may not arrive in the expected sequence (if at all).[325] Moreover, cyber-physical systems operate in real-time, so any crash protection must activate within a useful window of time.[326] Some delay is unavoidable, but taking too long to detect a software error—like waiting too long to deploy an airbag—could be fatal.

---

323. *See* Leslie Lamport et al., *The Byzantine Generals Problem*, 4 ACM TRANSACTIONS ON PROGRAMMING LANGUAGES & SYS. 382 (1982); Driscoll et al., *supra* note 192, at 237–38. Current research has offered algorithms that will tolerate up to one-third of the nodes being compromised. *See* Miguel Castro & Barbara Liskov, *Practical Byzantine Fault Tolerance*, 3 PROC. SYMP. ON OPERATING SYS. DESIGN & IMPLEMENTATION 173 (1999).

324. *See* Lee, *supra* note 192, at 365–66 (criticizing the basic design choice to hide timing properties from higher software layers); Sha et al., *supra* note 193, at 3–4; Rajkumar et al., *supra* note 193, at 735.

325. *See* Lee, *supra* note 192, at 4 ("Concurrent software often has timing-dependent behavior in which small changes in timing have big consequences."); Linh Thi Xuan Phan, *Towards a Safe Compositional Real-Time Scheduling Theory for Cyber-Physical Systems*, 4 ANALYTIC VIRTUAL INTEGRATION CYBER-PHYSICAL SYS. WORKSHOP 21, 22 (2013) (explaining that "even small discrepancies [in timing] can cause scheduling anomalies and thus 'snowball' into large anomalies"); *see also* Driscoll, *supra* note 192, at 241 (stating that the more common problems for Byzantine fault propagation are in the time domain).

326. *See* Linh Thi Xuan Phan, *supra* note 325.

3. Recovery

Once the system has detected an error, it must select an appropriate recovery mode. The solution space consists of backward recovery and forward recovery.[327] Backward recovery techniques attempt to restore or "roll back" the system to a prior state where the error had not yet occurred. These methods are especially well-developed in database technologies where they are critical to guaranteeing the reliability of financial ledgers and other transactions that demand high fidelity but not real-time availability.[328]

For cyber-physical systems, where timing considerations are inherently vital, forward recovery techniques are the more optimal choice.[329] The essential thrust is to neutralize the detected error by switching to a new state, rather than by reverting to a prior state. By physical analogy, if backward recovery is like rewinding a tape, then forward recovery is like swapping out the tape—or even the entire tape deck.

The gold standard of forward recovery is for the system to self-correct or mask faults without skipping a beat. That seamlessness is best achieved by executing redundant software processes in parallel, and cycling out faulty components upon detection.[330] Given the difficulties of fault detection, cruder models cycle components on a proactive basis even where no fault has been detected.[331] More elegant alternatives to brute-force redundancy include "roll forward" estimators that attempt to

327. *See* LEE & ANDERSON, *supra* note 298, at 144; PULLUM, *supra* note 30, at 13–17.

328. *See* Michael Treaster, *A Survey of Fault-Tolerance and Fault-Recovery Techniques in Parallel Systems* (Jan. 1, 2005) (pre-print draft), https://arxiv.org/abs/cs/0501002.

329. *But cf.* Maarja Kruusmaa et al., *Don't Do Things You Can't Undo: Reversibility Models for Generating Safe Behavior*, 2007 IEEE INT'L CONF. ON ROBOTICS & AUTOMATION 1134.

330. *See* Hitt & Mulcare, *supra* note 308, at 8-12 to 8-13 (describing fault masking and fault containment techniques in avionics); Shu-Yi Yu & Edward J. McCluskey, *On-line Testing and Recovery in TMR Systems for Real-Time Applications*, 2001 PROC. IEEE INT'L TEST CONF. 240.

331. *See* Jialei Liu et al., *Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability*, 6 IEEE TRANSACTIONS ON CLOUD COMPUTING 1192 (2016); Miguel Castro & Barbara Liskov, *Practical Byzantine Fault Tolerance and Proactive Recovery*, 20 ACM TRANSACTIONS ON COMPUTER SYS. 398, 400 (2002) (describing "a proactive recovery mechanism" that "recovers replicas periodically even if there is no reason to suspect that they are faulty"); Airworthiness Directive, 81 Fed. Reg. 86,912 (Dec. 2, 2016) (ordering power to be periodically reset on Boeing 787 aircraft to avoid "simultaneous reset [of all three flight control modules] if continuously powered on for 22 days"); *cf.* George Candea & Armando Fox, *Crash-Only Software*, 9 PROC. WORKSHOP ON HOT TOPICS IN OPERATING SYS. 67 (2003) (arguing that rebooting from a crash can be faster than and preferable to performing a clean shutdown and reinitialization).

forecast what the next actions would have been had the error not occurred.[332]

Where fault masking is infeasible or imprudent, the safer choice may be to transition to graceful degradation or shutdown.[333] These types of reconfiguration or adaptive recovery have long been considered the most challenging to construct, because they require intimate knowledge of both the system and the surrounding environment.[334] For example, one natural option for an autonomous vehicle might be to pull over to the breakdown lane, but where that is and whether the car can get there depends on a host of variables specific to the vehicle and the road it is on. In the medical setting, the appropriate response for a heated catheter might be to shut off, but for a ventilator the safe mode might be to stay on.[335] This problem of heterogeneity has received little concentrated attention from the research community, but it is arguably the most important because it represents the catch-all condition when things go wrong. The default workaround is to trigger a manual override and outsource any improvisational tasks to

332. *See* Václav Mikolásek & Hermann Kopetz, *Roll-Forward Recovery with State Estimation*, 14 PROC. IEEE INT'L SYMP. ON OBJECT/COMPONENT/SERV.-ORIENTED REAL-TIME DISTRIBUTED COMPUTING 179 (2011).

333. *See, e.g.*, Ehsan Dehghan-Azad et al., *Sensorless Control of IM for Limp-Home Mode EV Applications*, 32 IEEE TRANSACTIONS ON POWER ELECTRONICS 7140 (2017); Oscar González et al., *Adaptive Fault Tolerance and Graceful Degradation Under Dynamic Hard Real-Time Scheduling*, 18 PROC. IEEE REAL-TIME SYS. SYMP. 79 (1997); Linh T.X. Phan & Insup Lee, *Towards a Compositional Multi-Modal Framework for Adaptive Cyber-Physical Systems*, 17 PROC. IEEE INT'L CONF. ON EMBEDDED & REAL-TIME COMPUTING SYS. & APPLICATIONS 67 (2011); *see also* Algirdas Avizienis, *Toward Systematic Design of Fault-Tolerant Systems*, 30 COMPUTER 51, 53 (1997) (describing the need to define "the acceptability of different modes of service (full, reduced, degraded, emergency, safe shutdown, and so on) for each phase and establish each mode's required service level").

334. *See* LEE & ANDERSON, *supra* note 298, at 146–47 (explaining the shortcomings of forward error recovery as including that it is "designed specifically for a particular system" and "inappropriate as a means of recovery from unanticipated faults"); PULLUM, *supra* note 30, at 17 (forward recovery is "application-specific," "can only remove predictable errors," and "requires knowledge of the error").

335. Thanks to Jane Chong for suggesting this example.

human intelligence.[336] But any remote backdoor necessarily introduces new vulnerabilities and failure points to the cyber-physical system.[337]

C. The Reasonable Fault-Tolerant System

The crashworthy code framework is a rule of reason. As such, it offers a nuanced alternative to the bright-line schemes favored by consumer protectionism and technology protectionism. At the same time, it is a different measure of reasonableness than orthodox applications of negligence and products liability law, because it redirects judicial scrutiny away from the initial code failure and instead toward the subsequent mitigation response.

To illustrate how this rule might work in practice, consider again the case described at the opening of this Article, *Singh v. Edwards Lifesciences Corp*.[338] In the actual case, the software error in the heart monitor was exacerbated by the manufacturer's knowledge and callous disregard of the risk, which greatly simplified the jury's decision to hold the manufacturer culpable. But suppose the manufacturer had no forewarning. A conventional negligence or products liability analysis might attempt to weigh the reasonableness of the manufacturer's software development process by considering whether the manufacturer had followed industry norms for code review and validation testing. But absent rare circumstances, this inquiry would be a dead end.

---

336. *See* Alex Davies, *Self-Driving Cars Have a Secret Weapon: Remote Control*, WIRED (Feb. 1, 2018, 7:00 AM), https://www.wired.com/2017/01/human-problem-blocking-path-self-driving-cars/ [https://perma.cc/C97F-7JKJ]; Mary L. Gray & Siddharth Suri, *The Humans Working Behind the AI Curtain*, HARV. BUS. REV. (Jan. 9, 2017), https://hbr.org/2017/01/the-humans-working-behind-the-ai-curtain [https://perma.cc/35YK-DJL7]; Gunar Schirner et al., *The Future of Human-in-the-Loop Cyber-Physical Systems*, 46 COMPUTER 36 (2013); Olivia Solon, *The Rise of 'Pseudo-AI': How Tech Firms Quietly Use Humans to Do Bots' Work*, GUARDIAN (July 6, 2018), https://www.theguardian.com/technology/2018/jul/06/artificial-intelligence-ai-humans-bots-tech-companies [https://perma.cc/88MG-3R8X]; Press Release, Ca. Dep't of Motor Vehicles, Driverless Testing and Public Use Rules for Autonomous Vehicles Approved (Feb. 26, 2018), https://www.dmv.ca.gov/portal/dmv/detail/pubs/newsrel/2018/2018_17 [https://perma.cc/GPH9-NAJ8] (announcing revisions to California DMV rules approving fully autonomous vehicles on condition that such vehicles have a "communication link between the vehicle and a remote operator"). *But see* Josiah Dykstra & Eugene H. Spafford, *The Case for Disappearing Cyber Security*, COMM. ACM, July 2018, at 40.

337. *See* Abdulmalik Humayed et al., *Cyber-Physical Systems Security—A Survey*, 4 IEEE INTERNET THINGS J. 1802, 1809–10 (2017) (explaining that cyber-physical systems have traditionally relied on an assumption of isolation or "security by obscurity," and that adding more connectivity increases the number of attack vectors); A. Michael Froomkin, *The Metaphor Is the Key: Cryptography, the Clipper Chip, and the Constitution*, 143 U. PA. L. REV. 709 (1995).

338. *See supra* text accompanying note 1.

By contrast, a court conducting a crashworthy code analysis would be able to scrutinize directly the code design itself, for the narrow purpose of inquiring whether it used appropriate software fault tolerance techniques. A total absence of such techniques within a safety-critical system would constitute a per se violation. Likewise, if some efforts had been made but were found cursory or insufficient, then the manufacturer would not escape liability. But as long as the manufacturer's efforts were reasonably adequate, then recovery would be barred under this claim even if Mr. Singh suffered the same horrific injuries as described in the actual case.

Two ensuing questions are *why* the shift to crashworthy code offers a better reasonableness framework, and *how much* crashworthiness is enough to be sufficient.

The first question can be addressed in three parts. From an engineering perspective, requiring software fault tolerance is less disruptive to the day-to-day practices of software engineering than second-guessing the correctness of each and every line of code. Adding software redundancy requires more programmer hours, but not qualitatively different ones. By relinquishing the assumption of fault-free code, most software engineers can continue to write and test code as they do today, without an abrupt overhaul of programming norms or culture.[339] Reframing cyber-physical liability in terms of crashworthiness or risk mitigation also has the virtue of aligning well with broader trends in cybersecurity practice.[340]

From the judicial perspective, a crashworthy code doctrine offers an easier analytical framework because it shifts the locus of tort scrutiny from whether any arbitrary segment of code is "unreasonable" or "defective," to whether a specific, smaller subset of code provides adequate failsafe functionality. Instead of second-guessing each and every software design decision, the judicial inquiry is limited to reviewing the reasonableness of only the fault tolerance aspects. In particular, adjudication modules need to be handled with heightened care, given how crucial their role is as the nerve center of fault detection. Luckily, the limited size and scope of those modules may make it cost-efficient to require use of formal methods as well as other emerging techniques such

339. Whether programming ethics should change is a normative question reserved for future work. *See also* Don Gotterbarn et al., *ACM Code of Ethics: A Guide for Positive Action*, COMM. ACM, Jan. 2018, at 121.

340. See *supra* note 227 and accompanying text.

as protected memory space on the processor chip.[341] More challenging will be the task of pushing companies to develop robust fault recovery techniques. In the near term, courts will likely require only modest improvements such as better emergency warnings and human handoffs.[342] More avant-garde techniques will need to be proven in the field before prevailing at trial.[343]

For consumers, a crashworthy code doctrine offers a more intuitive way to describe whether a cyber-physical injury is unreasonable, without having to understand the technical reasons for the code crash that caused it. That explainability generates a more effective cause of action at trial, which in turn engenders more consumer trust that code quality can be vetted in court. It also has the potential to lower market prices by reducing development and testing times, cutting down on pass-through costs such as insurance premiums, and decreasing barriers to entry.[344] Hopefully it can save some lives, too.

The second issue—how much fault tolerance is "good enough"—is the proverbial devil in the details. Every case involving cyber-physical injury will necessarily feature an instance where fault tolerance has failed (or is absent). But crashworthiness does not mean code must prevent all faults, nor does it mean code must never cause physical harm. In code crashes—just as in automotive crashes—safety measures will sometimes fail to prevent death, bodily injury, or property damage. Not all such cases should trigger liability. How courts choose to meter liability will be the determinative factor that distinguishes this doctrine from absolute liability and absolute immunity.

Of the two common-law limitations developed in the automotive context, fault apportionment and state of the art, the former is likely to play a diminished role in the cyber-physical setting. That is not to say end users can never be held at fault. Those with direct physical access to the

---

341. *See* Victor Costan & Srinivas Devadas, *Intel SGX Explained*, *in* IACR CRYPTOLOGY EPRINT ARCHIVE 1, 2 (2016), (describing an Intel processor architecture that "protects the integrity and confidentiality of the computation inside an enclave by isolating the enclave's code and data from the outside environment"). *But see* Johannes Götzfried et al., *Cache Attacks on Intel SGX*, 10 PROC. EUR. WORKSHOP ON SYS. SECURITY (2017) (describing side-channel vulnerabilities in the SGX method).

342. *But see* Geistfeld, *supra* note 13, at 1626–29; Tracy Hresko Pearl, *Fast and Furious: The Misregulation of Driverless Cars*, 73 N.Y.U. ANN. SURV. AM. L. 19, 31–34 (2017) (collecting skepticism that "a quick handoff from machine to human is feasible").

343. *See supra* notes 292–296 and accompanying text.

344. *But cf.* Martha Chamallas, *The Disappearing Consumer, Cognitive Bias and Tort Law*, 6 ROGER WILLIAMS U. L. REV. 9 (2000) (describing the focus on efficient pricing in business and products liability law as being in opposition to the moral values of the consumer law movement in the United States).

system will be able to "jailbreak," "mod," or otherwise misuse the system in unauthorized ways.[345] Apportionment may be apt as well when a system is designed to hand off control to the end user in case of emergency, and the end user is negligent in not taking the helm.[346] As for third parties who obtain remote access to hijack or disable cyber-physical systems from afar,[347] courts should refuse apportionment. Even though these instances involve a clear intervening actor, anyone designing a software fault tolerance scheme should be expected to foresee and respond to cyberattacks.[348]

As apportionment claims fade in relevance, courts can expect to see a concomitant rise in disputes over the state of the art, as manufacturers seek to limit their damages. Courts do not expect manufacturers to achieve the impossible,[349] but even within the realm of possible, there is plenty of room for minds to differ as to what software engineering is presently capable of bringing to market. One principal fault line will be to screen which persons are competent to speak as experts in the field. For complex technologies such as software engineering, courts defer to technical experts but play a crucial gatekeeper function in disallowing non-credible witnesses.[350] The American experience with automotive engineering

345. *See* Trace H. Jackson, *Can Jailbreaking Put You in Jail, Broke?*, 68 FLA. L. REV. 631 (2016); Pamela Samuelson, *New Exemptions to Anti-Circumvention Rules*, COMM. ACM, Mar. 2016, at 24.

346. *See, e.g.*, Neal E. Boudette, *Tesla's Self-Driving Tech Cleared in Crash Inquiry*, N.Y. TIMES, Jan. 20, 2017, at B1; Tom Krisher & Jacques Billeaud, *Police: Backup Driver in Fatal Uber Crash Was Distracted*, ASSOCIATED PRESS, June 22, 2018; Press Release, Nat'l Transp. Safety Bd., Preliminary Report Issued for Investigation of Fatal, Mountain View, California, Tesla Crash (June 7, 2018), https://www.ntsb.gov/news/press-releases/pages/nr20180607.aspx [https://perma.cc/VN2X-VBKY]. *But cf.* John R. Quain, *The Autonomous Car vs. Human Nature*, N.Y. TIMES, July 8, 2016, at B4.

347. *See* Cahen v. Toyota Motor Corp., 147 F. Supp. 3d 955 (N.D. Cal. 2015).

348. *Compare In re* Sept. 11 Litig., 280 F. Supp. 2d 279 (S.D.N.Y. 2003) (aircraft manufacturers owe a duty to install unbreachable cockpit doors that locks out hijackers), *and* Nash v. Port Auth. of N.Y. & N.J., 856 N.Y.S.2d 583 (App. Div. 2008), *with* Port Auth. of N.Y. & N.J. v. Arcadian Corp., 189 F.3d 305 (3d Cir. 1999), *and* Stahlecker v. Ford Motor Co., 667 N.W.2d 244 (Neb. 2003). For a longer discussion of the role of foreseeability in duty to guard against acts of terrorism, see JOHN C.P. GOLDBERG & BENJAMIN C. ZIPURSKY, OXFORD INTRODUCTION TO TORTS 171–76 (2010).

349. *See, e.g.*, Anderson v. Owens-Corning Fiberglas Corp., 810 P.2d 549, 556 (Cal. 1991) ("[I]f a manufacturer could not count on limiting its liability to risks that were known or knowable at the time of manufacture or distribution, it would be discouraged from developing new and improved products for fear that later significant advances in scientific knowledge would increase its liability."); Henderson v. Ford Motor Corp., 519 S.W.2d 87 (Tex. 1974) ("[T]he manufacturer is not charged by the law nor expected by the purchasing public to design every part to be the best that science can produce or to guarantee that no harm will befall the user.").

350. *See* Eli Siems & Kathy Strandburg, Trade Secrets and Markets for Evidential Forensic Technology 22–37 (unpublished manuscript) (draft on file with author) (describing the rise of the

shows that the establishment of industrial research laboratories, trade groups, and governmental advisory panels can have a powerful influence on who gets a seat on the witness stand.[351] The software engineering community is far less cohesive than the Big Four automakers were in their heyday, and it has long prided itself on its antiestablishment orientation and hobbyist culture.[352] This freedom to code has been an asset in attracting talent in the early decades, and should continue to be embraced in most software development settings. For safety-critical applications, though, formal organization as a professional discipline would help generate an expert consensus that would have powerful influence over legal determinations of the state of the art.

Turning to the merits of the state-of-the-art defense, the closest question will be how to draw lines between research in the lab that is too speculative to trigger liability, versus commercially ready technology that is unavailable for reasons other than viability. By analogy, automakers neglected to install side mirrors, door latches, and other straightforward safety measures long after they were feasible, because they were viewed as unnecessary frills.[353] Automakers' deliberate disregard of these simple, life-saving measures fueled an angry backlash against the industry. Arguably, within the field of software fault tolerance, *redundancy* and *adjudication* techniques are sufficiently well-developed and generalizable to make the case that companies should already be availing themselves of those safety measures. Less persuasive is the readiness and availability of *recovery* techniques, because the requirements are so heterogeneous, the solution set so open-ended, and the legal pressures nonexistent, that research efforts have been sparse.

Where the current state of the art falls short of the desired mark, lawmakers may wish to explore interim measures to incentivize advancement of the art. For instance, cyber-physical manufacturers could

---

Daubert standard, which positions judges as gatekeepers, over the Frye standard, which defers entirely to the scientific community); Sarah Jeong, *The Judge's Code*, VERGE (Oct. 19, 2017, 10:57 AM), https://www.theverge.com/2017/10/19/16503076/oracle-vs-google-judge-william-alsup-interview-waymo-uber [https://perma.cc/MYE5-CN29] (profile on Judge William H. Alsup, who received media attention for being the rare judge claiming to possess some knowledge of software programming).

351. *See* NADER, *supra* note 22, at ch.7.

352. A compelling analogy can be drawn to the early days of auto manufacturing, when the field was littered with hundreds of independent inventors and small entrepreneurs scattered across the country, before the industry consolidated in the 1920s. *See generally* BEVERLY RAE KIMES, PIONEERS, ENGINEERS AND SCOUNDRELS: THE DAWN OF THE AUTOMOBILE IN AMERICA (2004).

353. *Cf.* O'CONNELL & MYERS, *supra* note 24, at 20 (quoting a tire official: "Detroit is probably the only place in the world where a ten-cent saving per car looks like $3.5 million.").

be required to generate and update an auditable Crashworthy Code Plan that justifies how their system detects and recovers from code crash events. Other cybersecurity regulations require broadly termed Privacy and Security Plans,[354] which have been criticized as lacking substantive remedies and being too deferential to weak industry practices.[355] Specifying a narrow, defined goal such as software fault tolerance may prove more effective at raising the bar. Tying the documentation requirement to the liability standard is important; early iterations of the plan may be weak, but the repeated exercise of having to defend how one's software fault tolerance compares to the state of the art would drive manufacturers to inspect and adopt new techniques at a faster clip. In turn, adoption by some manufacturers would influence and advance peer perceptions of the state of the art, which in turn elevates the standard of reasonable fault tolerance.

Extending the concept of crashworthiness beyond its original motor vehicle context will likely raise new complications. Heterogeneous cyber-physical systems may demand new affirmative defenses. What is excusable for a self-driving school bus may be different from what is excusable for an insulin pump.[356] The precise contours will need to be sorted out as such systems are developed and deployed. While much of that future discussion will invariably focus on establishing an upper bound on the doctrine, it is worth closing with a reminder that the crashworthy code doctrine is foremost a mechanism for raising the minimum floor of software safety.

CONCLUSION: MEMENTO MORI

Code is not perfect, but it can be safer. By accepting that code crashes are statistically inevitable, courts can skip ahead to how those crashes could be mitigated, rather than getting mired in the fool's gold of crash

---

354. *See, e.g.*, Federal Information Security Management Act, 44 U.S.C. §§ 3541–3549 (2018); HIPAA Security Rule, 45 C.F.R. §§ 160, 164 (2013); Gramm-Leach-Bliley Act Safeguards Rule, 16 C.F.R. § 314 (2018).

355. *See* Rustad & Koenig, *supra* note 6, at 1594–98; Paul N. Otto, Note, *Reasonableness Meets Requirements: Regulating Security and Privacy in Software*, 59 DUKE L.J. 309, 325–29 (2009). *Contra* Daniel J. Solove & Woodrow Hartzog, *The FTC and the New Common Law of Privacy*, 114 COLUM. L. REV. 583, 655–56 (2014) (praising the HIPAA Security Rule as "one of the most specific data security laws" and arguing that industry customs are still better than no standards).

356. *See* MASHAW & HARFST, *supra* note 26, at 141–46; David Shepardson, *U.S. Regulator Orders Halt to Self-Driving School Bus Test in Florida*, REUTERS, Oct. 22, 2018, https://www.reuters.com/article/us-usa-selfdriving/u-s-regulator-orders-halt-to-self-driving-school-bus-test-in-florida-idUSKCN1MW2SG [https://perma.cc/G3VQ-ZPAV].

prevention. Like the crashworthy doctrine for automotive vehicles, a crashworthy doctrine for code requires engineers to design for safer crashes, not just for safer intended uses. In particular, the computer science literature on software fault tolerance provides a mature toolkit that could be mandated for all safety-critical cyber-physical systems.

The crashworthy doctrine is a common law judicial doctrine, but its lessons could be embraced and amplified by other regulatory bodies.[357] For example, federal regulators at NHTSA and the FDA have hesitated to issue firm guidance on cyber-physical safety, reflecting collective, shared apprehension about how best to regulate software.[358] Working in mutual conversation with the judiciary to establish and expound a crashworthy code standard may be more robust than each attempting to venture forth alone.[359]

Another area for future work is extension to artificial intelligence techniques such as deep learning where scholarly concern has focused more on data errors than on code errors. As these learning algorithms have won acclaim for startlingly impressive demonstrations, they have also come under heavy criticism for their inability to explain those results.[360] Researchers have questioned the trustworthiness of those algorithms, showing that bias in the data leads to bias in the results,[361] leading to calls

---

357. *Cf.* Mark A. Geistfeld, *The Regulatory Sweet Spot for Autonomous Vehicles*, 53 WAKE FOREST L. REV. 337 (2018). *See generally* Shavell, *supra* note 91 (exploring the economic tradeoffs of controlling risk through regulation versus through liability).

358. MASHAW & HARFST, *supra* note 26, at 224–54 (expressing the challenges of "technology-forcing" regulation); GRAHAM, *supra* note 231, at 37, 55–56 (same).

359. The U.S. Supreme Court's pending argument in Kisor v. Shulkin, 869 F.3d 1360 (Fed. Cir. 2017), *cert. granted sub nom.*, Kisor v. Wilkie, 202 L. Ed. 2d 491 (U.S. Dec. 10, 2018) (No. 18-15), appears poised to transfer power from federal agencies to courts, which could revitalize common law over regulatory rulemaking. *See also* Justin (Gus) Hurwitz, *Data Security and the FTC's UnCommon Law*, 101 IOWA L. REV. 955 (2016) (criticizing the legitimacy of legal jurisprudence developed through the administrative process).

360. Kiel Brennan-Marquez, *"Plausible Cause": Explanatory Standards in the Age of Powerful Machines*, 70 VAND. L. REV. 1249 (2017); Andrew D. Selbst & Solon Barocas, *The Intuitive Appeal of Explainable Machines*, 87 FORDHAM L. REV. 1085 (2018). *But cf.* David Lehr & Paul Ohm, *Playing with the Data: What Legal Scholars Should Learn About Machine Learning*, 51 U.C. DAVIS L. REV. 653 (2017); Natalie Wolchover, *New Theory Cracks Open the Black Box of Deep Learning*, QUANTA MAG. (Sept. 21, 2017), https://www.quantamagazine.org/new-theory-cracks-open-the-black-box-of-deep-learning-20170921/ [https://perma.cc/WQ88-3CJC] (describing "information bottleneck" theory of deep neural networks); *see also* Will Knight, *The Dark Secret at the Heart of AI*, MIT TECH. REV., May/June 2017, at 54, https://www.technologyreview.com/s/604087/the-dark-secret-at-the-heart-of-ai/ [https://perma.cc/2DHF-NV65].

361. *See* Julia Angwin et al., *Machine Bias*, PROPUBLICA (May 23, 2016), https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing [https://perma.cc/HA2H-CB7H]; Solon Barocas & Andrew D. Selbst, *Big Data's Disparate Impact*,

for fairness, transparency, and due process in how those results are applied to citizens.[362] If a misclassification could be reframed instead as a kind of algorithmic "crash," then a logical follow-up would be to ask whether a crashworthy design could achieve better performance than one that is blind to its own fallibility.[363]

Adopting a crashworthy code doctrine would not be an overnight fix. Manufacturers would need to adjust budget allocations and alter code design practices. Courts would need to address important questions such as how much redundancy is reasonable; what recovery modes should be legally required; and whether code written for fault tolerance modules can be held to a heightened standard of reliability. Litigating these questions could take years if not decades. But by comparison, courts allowed half a century to go by before applying a crashworthy standard to the automotive industry. Crashworthy code is the looked-for lynchpin of the coming cyber-physical era.

---

104 CALIF. L. REV. 671 (2016); Batya Friedman & Helen Nissenbaum, *Bias in Computer Systems*, 14 ACM TRANSACTIONS ON INFO. SYS. 330 (1996).

362. *See* Sam Corbett-Davies, *Algorithmic Decision-Making and the Cost of Fairness*, 23 ACM PROC. INT'L CONF. ON KNOWLEDGE DISCOVERY & DATA MINING 797 (2017) (quantifying the tension between improving public safety and satisfying prevailing notions of algorithmic fairness); Muhammad Bilal Zafar, *Fairness Constraints: Mechanisms for Fair Classification*, 54 PROC. MACHINE LEARNING RES. 962 (2017); Shira Mitchell et al., Prediction-Based Decisions and Fairness: A Catalogue of Choices, Assumptions, and Definitions (Nov. 20, 2018) (unpublished manuscript), https://arxiv.org/pdf/1811.07867.pdf; Kroll et al., *supra* note 194; Wexler, *supra* note 10.

363. The layering of multiple AI techniques is well-established in the art, but typically is not used for redundancy. *See* David Silver et al., *Mastering the Game of Go with Deep Neural Networks and Tree Search*, 529 NATURE 484 (2016) (describing AlphaGo's combined use of two neural networks performing separate functions).