# Chapter Six

## WON'T SOMEBODY PLEASE THINK OF THE INNOVATION?

I've now spent five weeks spelling out the follies of fintech, top to bottom and side to side. I've provided examples of the limits on what individual technologies (particularly blockchain and generative AI) can achieve, and I've highlighted the elements of some broad structural problems (like financial inclusion) that Silicon Valley will never come close to solving. Now, I want to shift to talking about why, when fintech's limitations seem so obvious when you spell it all out, belief in fintech's promise is so pervasive. Part of it is that Silicon Valley makes a lot of money from fintech hype, and so has vested interests in perpetuating it (particularly in weaponizing that hype to mold the law to its whims). We'll get to all of that in coming chapters. But tech hype isn't completely cynical. There is often genuine – if unjustified – optimism among founders and followers about the ability to use technology to innovate our way past any obstacle. That's what I want to start with here.

There's no one single cause of, or explanation for, this kind of techno-solutionism. It might come from an almost religious belief in the power of technological innovation (a belief often encouraged by the media). Or it could be prompted by an ideological aversion to government solutions – an aversion so strong that even the most unrealistic promises from the private sector seem appealing by comparison. Or it could spring from what we might call an "extreme engineering" view of the world that sees everything as a technological puzzle waiting to be solved. At a more fundamental level, our brains sometimes conspire against us to naively embrace technological solutions that don't actually make a whole lot of sense.

The unfortunate result is that our society tends to revere technological innovation as unqualifiedly positive, as a process that should be permitted at all costs, even if the goal is an unrealistic pipe dream. As I said in the introduction to this book, we do need optimists and their different ways of looking at the world. However, that optimism (translated into a faith in technological innovation's potential to cut, quickly and meaningfully, through the Gordian Knot of social problems) is given far too much airtime and credibility in today's society.

## Innovation worship

There's an episode of the sitcom *30 Rock*, where Steve Martin plays an extremely wealthy agoraphobic man holed up in a Connecticut mansion. As it turns out, though, [spoiler alert] his character is not actually agoraphobic but instead under house arrest for tax fraud, embezzlement, and racketeering ("what is racketeering? No one knows"). All his crimes have been committed in connection with his company Sunstream, which [spoiler alert #2] turns out to have been a fake company all along. The scene cuts away to a Sunstream commercial that shows suns rising and cars driving and eagles flying, and then flashes the three magic words: “innovation,” “tomorrow” and “America.” In 21st century America, “innovation” is a sacred cow that few would dare to question. That’s why Steve Martin’s character hid his fraudulent business behind it.

American Studies professor David Nye has described the United States as a nation held together in part by its shared sense of the “technological sublime,” its affection for spectacular technologies. In recent decades, we have come to see innovation as the source of that technological sublime, but humans haven’t always worshipped at the altar of innovation. Prior to the 19th Century, the word “innovator” was often associated with deviance and immorality, used to describe heretics or dissidents who rebelled against the order established by a monarch or by a religion. Starting in the 19th century, though, the perception of innovation began to gradually change, and the 180-degree turn was complete by the time the twenty-first century dawned. Disruptive change that had once upon a time been condemned as “innovation” is now praised and venerated as “innovation,” particularly in technological contexts. Apple founder Steve Jobs, probably the seminal tech innovator of our time, was eulogized as a “secular prophet,” and the modern-day heresy (one I’m encouraging you all to commit!) is to express skepticism about tech innovation’s ability to improve anything and everything.

Innovation is regularly described as an inexorable force that we couldn’t stop if we tried. We’re also told that the benefits of innovation are so valuable that we should never take any action that might threaten innovation (we’re supposed to somehow embrace the paradox that any attempt to stomp out bad innovation would be futile, and also that stomping out bad innovation is dangerous because it will stomp out good innovation). But what exactly does innovation mean? Etymology isn't always helpful, but I'll be damned if I won't take advantage of this opportunity to use my high school Latin. The word derives from the Latin "innovare," made up of the building blocks "in" (meaning "into") and "novare" (meaning "new"). So now you know! We're making something new! But there are lots of ways to do that, and in Silicon Valley, there's a particular brand of innovation that holds court: disruptive innovation.

This version of innovation draws heavily on Austrian economist Joseph Schumpeter's work on "creative destruction." Schumpeter believed that creative destruction was the primary engine of capitalism and economic growth – here's the money quote, from his book *Capitalism, Socialism, and Democracy*, first published in 1942:

> *The fundamental impulse that sets and keeps the capitalist engine in motion comes from the new consumers' goods, the new methods of production or transportation, the new markets, the new forms of industrial organization that capitalist enterprise creates...[a process] that incessantly revolutionizes the economic structure from within, incessantly destroying the old one, incessantly creating a new one. This process of Creative Destruction is the essential fact about capitalism.*

In his 1997 book *The Innovator's Dilemma*, Harvard Business School Professor Clayton Christensen embraced and embroidered upon this idea, concluding that "disruptive technology" is what makes Schumpeter's creative destruction possible. Christensen's goal with *The Innovator's Dilemma* was to put forward a theory that could predict when businesses would succeed and when they would fail as a result of their approach to innovation. We know from Chapter 4, though, that what constitutes a success or a failure will be in the eye of the beholder, and historian Jill Lepore has critiqued Christensen's cherry-picking of case studies, his disregard for the impact of other historical forces on companies' fortunes, and his arbitrary choices of time-frames for assessing success or failure. I might add that Christensen's theory doesn't reckon with the venture capital funding that disruptors can rely upon to subsidize an inferior product until they've put incumbents out of business. But the cultural significance of the concept of "disruptive innovation" can't be denied, so let's unpack it a little more.

In *The Innovator's Dilemma*, Christensen theorized that good companies who invest in innovations that respond to the needs of their best customers may be inadvertently sowing the seeds of their own demise. This is the eponymous dilemma that incumbent businesses face: following this kind of common-sense management style leaves them vulnerable to being outcompeted by a business that starts out exploiting a niche market for innovations that don't work as well and aren't as profitable, but can eventually scale up by taking advantage of technological progress to outcompete the incumbents. The first kind of innovation, innovation that improves on the core business model to make existing customers happier, Christensen calls "sustaining;" the latter is "disruptive" and relies on technologies that are "typically cheaper, simpler, smaller and, frequently, more convenient to use."

You don't hear many people in Silicon Valley talking about sustaining innovation – it's all disruption, disruption, disruption. Lepore describes the term "disruptive innovation" as having taken on messianic proportions, “holding out the hope of salvation against the very damnation it describes: disrupt, and you will be saved.” But as it has grown in popular usage, many of the contours and nuances of Christensen’s definition of disruptive innovation have been lost. Uber, for example, is widely held up as disruptive innovation, even though Christensen was reported as saying it didn’t fit within his theory (he viewed Uber as a sustaining innovation, because it was just replicating and improving taxi services, something that already existed). Lepore also recounts a Business Week interview in 2007 where Christensen said that the iPhone wasn’t a disruptive innovation because it was too high-end (he also said his theory predicted that the iPhone would not be a success for Apple). But a quick Google search “was the iPhone disruptive” yields the AI overview response “Yes, the iPhone is considered a notable example of disruptive innovation” (Google’s AI results are not always reliable, but if you’re trying to get a sense of what the internet thinks, it can be a pretty good approximation).

It seems safe to say that the common usage of “disruptive innovation” has morphed into a more all-purpose description for “Silicon Valley technology being deployed in new domains to take market share from incumbents.” But despite this evolution, the messianic connotations that Jill Lepore identified remain: this kind of disruptive innovation is still seen as a panacea that can fix just about anything. Unfortunately, Silicon Valley disruptors often don’t know much about the domains they propose to disrupt, and so they may not understand (or even care) why a particular industry evolved a particular way, or why its pain points are what they are. Silicon Valley’s outside perspective can be helpful to a degree: outsiders are well positioned to break out of the groupthink box, to come up with new and creative ideas about how to tackle problems that incumbents have simply come to accept as their lot. But outside-the-box thinking only gets you so far when you don't understand the basics. As one incisive journalist put it, when we're talking about disruptors, "have we just constructed a sexy new language to talk about novices?"

Acquiring unfamiliar knowledge and researching the nuances of the problem at hand is often hard and expensive work, and the first impulse for any profit-driven enterprise is to find an opportunity to exploit quickly and cheaply. After all, that's the type of opportunity Christensen identified as ripe for exploitation by disruptors. Unfortunately, disruptive innovators haven't always confined their ambitions to opportunities that are well-suited to a quick and dirty reboot. When we've reached the point that someone like Elizabeth Holmes, who had no biomedical expertise and didn't care to listen to anyone who did, can be feted for her vision for Theranos' disruptive blood testing innovations – well, it's clear that innovation worship has jumped the shark. The first requirement for disruptive innovation is an enabling technology that, you know, works, but those who want to see the receipts are often accused of being "anti-innovation."

## Weaponizing innovation worship

More than a decade ago, before I started researching technology's impacts on our society, I was asked to join an academic workshop to talk about innovation in financial services. I made what I thought was an innocuous and non-controversial statement: "not all innovation is beneficial." I had in mind some of the more byzantine financial products that had helped fuel the 2008 crisis – things with esoteric names like synthetic CDO squareds. Post-2008, most people accepted that these kinds of financial innovations had not turned out to be good for society (although they certainly made a lot of money for big financial institutions pre-2008). I hadn't really engaged with a lot of innovation worship at that point in my career, and I certainly wasn't expecting anyone to get all red and discombobulated in response to my comment and seethe back at me "but all innovation is by definition an improvement!" But that's what happened.

At the time, I was really quite taken aback. That other professor had invoked the idea of "innovation" to forestall any conversation about the downsides of whatever it was we were talking about (I think it was online crowdfunding). While it shocked me at the time, this is now something I deal with all the time in conversations about fintech. There's a classic Simpsons episode where Helen Lovejoy, the Reverend's wife, implores "won't somebody please think of the children?"; this hackneyed phrase has become a meme used to call out anyone who tries to tug on emotional heartstrings to justify whatever it is they want to do. In a similar vein, "won't somebody please think of the innovation?" pleads with us not to do anything that might mess with our feelgood sense of innovation and the seemingly inevitable improvements that come with it. But a question I've posed again and again in this book is, whose values decide the matter? When it comes to innovation, who gets to decide whether it is, in fact, an improvement?

Do we judge an innovation by the fact that it has attracted a lot of investment? (in the sense of that immortal line from the Simpsons monorail song, "Sorry Marge, the mob has spoken!" – I'm clearly on a Simpsons kick right now). I've certainly been told that the amount of money invested in bitcoin proves it's a good innovation – and I've also quietly wondered whether, by the same logic, Bernie Madoff's Ponzi scheme should also feature in the innovation hall of fame. Do we judge an innovation by whether it cornered the market? In that case, the Sacklers innovated an excellent way of delivering opioids to the American people: Oxycontin has been described as a "commercial triumph, public health tragedy."

I would humbly suggest that, in light of these examples, and all the examples of fintech exploitation we covered earlier in the book, we need to start asking what other public tragedies are being perpetuated under the guise of innovation.

Conduct that we would otherwise find very problematic can be imbued with positive connotations and legitimized just by labeling it as "innovation:" in their book *The Innovation Delusion*, Lee Vinsel and Andrew Russell talk a lot about the weaponization of "innovation speak," which they describe as a "sales pitch about a future that doesn't yet exist" that is "built on the hidden, often false premise that innovation is inherently good." They argue that although this kind of rhetoric "is often cast in terms of optimism, talking of opportunity and creativity and a boundless future, it is in fact the rhetoric of fear. It plays on our worry that we will be left behind." This innovation speak can be deployed to attract investment, juice adoption, and to discourage regulators from intervening, even when a technology can't deliver on its hype. As tech columnist Charlie Warzel put it, "the greatest trick of a faith-based industry is that it effortlessly and constantly moves the goal posts, resisting evaluation and sidestepping criticism. The promise of something glorious, just out of reach, continues to string unwitting people along. All while half-baked visions promise salvation that may never come."

Faith in technological innovation's promise can be enough to lock a particular tech business model into our lives, making it very difficult to dislodge (along with all the harms and distractions that accompany it) once evidence emerges that it is not, in fact, the path to salvation. For example, as economists Daron Acemoglu and Simon Johnson note in their book *Power and Progress*, "if everybody becomes convinced that artificial intelligence technologies are needed, then businesses will invest in artificial intelligence, even when there are alternative ways of organizing production that could be more beneficial." Weaponized innovation worship is directed particularly keenly at regulators (we innovators alone can save the world, so don't you bureaucratic fuddy-duddies get in our way!), and it can make regulators' already difficult job of protecting the public inestimably harder. This interplay between innovation worship and regulation – something we'll really get into in Chapter 8 – highlights something that often flies beneath the radar: that innovation worship can have a decidedly ideological dimension.

## The politics of innovation worship

In June of 2023, a submersible operated by the OceanGate expeditions company imploded during a dive to see the Titanic wreck, killing all five passengers on board (including OceanGate's CEO Stockton Rush). It was a tragic event, but also a vivid illustration of the perils of innovation worship – and its political leanings. Numerous experts had raised concerns that the submersible was unsafe before the ill-fated trip, but Rush told Smithsonian Magazine in 2019 that "well-meaning" passenger safety regulations "needlessly prioritized passenger safety over commercial innovation." He also told the magazine that "there hasn't been an injury in the commercial sub industry in over 35 years. It's obscenely safe, because they have all these regulations. But it also hasn't innovated or grown—because they have all these regulations." Stockton's innovate-at-all-cost mindset and antipathy for regulation ended up costing five people their lives. As writer Nathan J. Robinson put it, "in industry standards and regulations, [Rush] does not see the accumulated wisdom of many generations of engineers, but a lot of pointless paperwork…I've heard variations on this story over and over…and it's a core part of the libertarian story of the world."

Silicon Valley has long been associated with libertarianism, and if your goal is to show that government is useless, then it is very useful if people believe that private sector innovation will always provide a better solution than democratically elected governments. The relationship between libertarianism and innovation worship works the other way as well: if someone firmly believes that technology *is* magic, that with enough money, data, and compute that anything is possible, then an explanation will be needed if it turns out the technology can't ultimately deliver. Admitting the fallibility or limitations of the technology would require that person to rethink their ideological priors, and we humans *hate* doing that. An easier path is to find another reason why the technology has not been able to live up to its full potential – a reason like, say, innovation-killing government regulation.

When author and computer programmer Ellen Ullman was working on a network for San Francisco service providers during the AIDS crisis, she sometimes found herself embarrassed to tell others in the tech industry what she was working on. Not because she had any qualms about helping those suffering from AIDS, but because of the stigma of working for the government:

> *But actually working on a project for end users? Where my client is a government agency? In the libertarian world of computing, where "creating wealth" is all, I am worse than uncool: I am aiding and abetting the bureaucracy, I am a net consumer of federal taxes – I'm what's wrong with this country.*

In her 1997 book *Close to the Machine: Technophilia and its Discontents*, Ullman also describes a romantic dalliance with a younger man named Brian whom she describes as "too smart and too isolated for his own good." But for the book being written a few decades too early, Brian might serve as the archetype of a crypto bro: he identifies as an anarchocapitalist who wants markets to operate outside the structures maintained by the law. He sees his mission in life as creating an entirely anonymous global banking system, to "arbitrage existing law to set up a banking system without being a bank." Brian also seemed to have the same kind of dreams that inspired Mark Zukerberg to waste billions on the Metaverse, voicing his aspirations to be the "net landlord" that takes a little cut every time someone clicks on content. May I remind you that Ullman's book was published in 1997? There is nothing particularly new (nor dare I say it, innovative) about these techno-libertarian fantasies.

For a more recent iteration of these fantasies, I watched the 2024 movie *God Bless Bitcoin* so that you don't have to. Towards the end of the movie, figures from different religious faiths reconcile bitcoin with their holy teachings, really putting the "worship" in "innovation worship." We hear that bitcoin is the most Islamic, Jewish, *and* Christian form of money ever – as well as aligned with Hindu and Buddhist principles. In case that wasn't impressive enough for you, we're told that "bitcoin represents an evolution of consciousness beyond anything we've seen in thousands of years." But I find this movie most interesting for how heavily it leans into libertarian notions of government as the root of all evil. The overwhelming message is that government is the problem, and that bitcoin can cut governments down to size.

The movie starts with the line "Remember the Brady Bunch? Back before our money was broken, you could have one parent working, while supporting a family with six kids, and a live-in maid. What happened?" So, a bit of barely-concealed panic about childless cat ladies failing to procreate to kick us off, before we segue into interviews with bitcoin luminaries (including early Elizabeth Holmes-backer Tim Draper, and billionaire Mark Cuban). The first interviewee, though, is United States Secretary for Health and Human Services and roadkill enthusiast RFK Jr, and right off the bat, he's blaming government for most of the world's ills. As the movie progresses, we hear that many of the problems in our society – genuinely troubling problems like economic inequality, the military-industrial complex, the worst excesses of capitalism – can be solved by the silver bullet techno-solution of bitcoin, because all those aforementioned problems stem from the root cause of what the movie calls "broken money."

According to the movie, "broken money" is money controlled by central banks, and it is apparently broken because government elites apparently use inflation to steal from working people. This is referred to as "centralized control of the economy," as a copy of the Communist Manifesto is held up and waved around. We're told that bitcoin, on the other hand, can obviate the role of democratically elected governments and the central bankers appointed by them, and in so doing "fix everything" including war and greed! How, you may ask? Well, there's a lot of refried gold bug talking points about bitcoin's limited supply limiting inflation (and zero reckoning with the volatility and deflationary consequences of limited supply that we discussed in Chapter 2). There's also a lot of focus on bitcoin's absence of intermediaries, and therefore absence of censorship and fees – claims that we have already thoroughly debunked in this book. El Salvador's bitcoin adoption is even held up as a success in the movie, rather than the abject and abandoned failure it really was (again, see Chapter 2).

But I guess if you genuinely believe that central banks and governments are the greatest threats we face, then you'll be more motivated to ignore or forgive the evidence of bitcoin's deficiencies – and some survey evidence suggests that crypto investors do indeed tend to skew libertarian. As Bloomberg columnist Matt Levine put it, crypto "take[s] the problems of traditional finance and make them, *worse*, sure, but…subject to unbridled free markets." Back in Chapter 4, I mentioned David Golumbia's book *The Politics of Bitcoin: Software as Right-Wing Extremism*, where he concludes that "Bitcoin and the blockchain technology on which it rests satisfy needs that make sense only in the context of right-wing politics." In 2024, the president of a conservative Super PAC went on the record with her agreement, stating that "ideological strands unite the crypto industry and founders with the [Republican] party itself. Which is, we support pro-freedom, pro-liberty, we support a max amount of choice to use your dollars, to have independence and freedom."

The rampant regulatory arbitrage associated with blockchain that we documented earlier in the book can only be justified if you believe that whatever bad things the crypto industry does beyond the reach of the law are far preferable to what a democratically elected government or central bank might do. As Ullman's young lover Brian put it, a financial system built on regulatory arbitrage would "incidentally, make the world safe for crooks, thieves, money launderers, and any average citizen who should just not feel like paying his taxes" but that is "a side effect of freedom he said, the price of liberty, can't be helped."

## Of overengineering and enshittification

Different people are predisposed to and influenced by innovation worship to different degrees and for different reasons, not all of them political or ideological. In his book on techno-solutionism, Evgeny Morozov talks about people who simply have an engineering mindset, and therefore like to see the world as a series of technological optimization problems to be fixed. Ellen Ullman offers excellent insight into this kind of perspective in *Close to the Machine*: it's really worth reading her whole book (which flows like poetry and has the added virtue of being short) but I'll share with you a couple of illuminating passages on the flow of the software engineering mindset. She writes:

> *Soon the programmer has no choice but to retreat into some private interior space, closer to the machine, where things can be accomplished. The machine begins to seem friendlier than the analysts, the users, the managers. The real-world reflection of the program – who cares anymore? Guide an X-ray machine or target a missile;*

> *print a budget or a dossier; run a city subway or a disk-drive read/write arm: it all begins to blur. The system has crossed the membrane – the great filter of logic, instruction by instruction – where it has been cleansed of its linkages to actual human life. The goal now is not whatever the analysts first set out to do; the goal becomes the creation of the system itself. Any ethics or morals or second thoughts, any questions or muddles or exceptions, all dissolve into a junky Nike-mind: Just do it. If I just sit here and code, you think, I can make something run…Talk all you want, but this thing here: it works.*

> And:

> *In another part of my being – later, perhaps when we emerge from this room full of computers – I will care very much why and for whom and for what purpose I am writing software. But just now: no. I have passed through a membrane where the real world and its uses no longer matter. I am a software engineer.*

People aren't necessarily born this way, though. An extreme engineering mindset can be built or encouraged through interactions with like-minded individuals, and within academic environments.

For years now, there's been a lot of emphasis on Science, Technology, Engineering, and Math (aka "STEM") education, often at the expense of humanities courses that consider the social context in which STEM output will be deployed and teach skills like communication and critical thinking. We were told that STEM was where all the jobs would be – advice that seems a little short-sighted in 2025, as tech companies are laying off employees and scientific research grants are being decimated – but back in 2016, STEM education was ascendant. That's when an interdisciplinary group of scholars came together to think about how education reform could help undo the "dominant view of engineering as one of detached technological quests apart from society," and instead "integrate the social sciences into engineering practice and research." The conference resulted in a book titled *Engineering a Better Future* that is filled with suggestions for better integration, but what I want to focus on here is their initial diagnosis: how does an engineering education help encourage the embrace of superficial technological solutions?

One workshop participant felt that there is often little opportunity for engineering students to reflect on "the power and the limits of their professional expertise," with students only being trained on engineering knowledge and skills and not in thinking critically about how and when to use those knowledge and skills for the common good. Another participant expressed concern that an engineering education often narrowed students' focus to solving small technical problems assigned by others; that by graduation, many students had accepted their lot as mere "cogs in the wheel." An earlier report cited in the introduction to the book found that engineering students typically don't learn much about "technological history and the role of social forces in the history of the development of technologies." If engineers have few opportunities to learn about the limits of their own knowledge, then techno-solutionist solutions are perhaps to be expected.

To be clear, I'm not suggesting that merely possessing an engineering degree condemns you to a life of techno-solutionism

(I also want to be clear that there are plenty of prominent techno-solutionists out there with no engineering chops to speak of – what's their excuse?). I'm married to an engineer who's a born optimizer, but I wouldn't call him a techno-solutionist because he is keenly aware of the limits of what he can optimize. Many of his fellow optimizers are also very aware that their technical expertise only goes so far. Many of them also focus their work on maintenance – driven to fix what is obviously broken with tools they know can do the job, rather than eternally seeking out new problems to fix with shiny technological toys. But Silicon Valley's relentless focus on growth at all costs can make it hard for software engineers to take an "if it ain't broke, don't fix it" approach.

> As an aside, there can be money in "if it ain't broke, don't fix it." Apple's Snow Leopard, for example, is arguably the most popular Mac operating system update of all time. Explicitly marketed as having "no new features," Snow Leopard simply fixed a bunch of existing bugs in the Mac operating system – and people still wax nostalgic about it. Sadly, Snow Leopard is an outlier.

If the status quo is seen as something that must always be improved upon, engineers will always be encouraged to keep tinkering, even if the result is worse than the status quo ante. To illustrate the dangers of this kind of overengineering, some engineering and business schools teach the cautionary tale of the 17th century Swedish warship, the Vasa. The Vasa was an extremely expensive warship considered critical to Sweden's war effort against Poland – and it capsized and sank about twenty minutes into its maiden voyage after encountering a light gust of wind.

Why was a light gust of wind enough to knock it over? Well, the ship was unstable, a fact that had been discovered by the Vasa's Captain and the Swedish Admiral, who had seen it perform abysmally during a "lurch test" (thirty men ran from side to side on the ship to see what would happen – it lurched so much that they stopped the test). This information was not communicated to the shipbuilder or to Swedish King Gustav, though, and it was decided to launch anyway because time was of the essence (there was a war going on), no one involved wanted to abandon the ship, and there was no way to fix it. An example of the *sunk* cost fallacy if ever there was one (see what I did there?).

What is interesting from our perspective is how the Vasa got "heavier above than below." The original design had been for a much smaller ship, but multiple changes to that design were made to accommodate the latest innovations in warship building – in particular, King Gustav had heard that the Danes were building a warship with two gun decks as opposed to the usual one, and he didn't want to be out*gunned* (ok, ok, I'll stop). And so the Vasa was modified to accommodate a second gun deck after the keel had already been laid, which required many other design modifications that also hadn't been tried before. The result was a ship with limited room for ballast and a high center of gravity – a center of gravity made higher as the decks were packed with more guns than had originally been envisioned, and as the ship was decorated with heavy ornate oak carvings meant to communicate to all who saw her that no expense had been spared.

In addition to demonstrating a level of hubris that rivals the Titanic, the story of the Vasa illustrates that unbridled pursuit of the latest innovations can lead to overengineering that undermines the primary function of something – in this case, a ship's ability to stay afloat. Messing with any existing system to accommodate new and unfamiliar technologies will inevitably increase the complexity of that system, and increased complexity tends to create unanticipated fragilities. Often, pressures to overengineer don't come from the engineers themselves, but from their bosses (like King Gustav), who have a specific vision and don't want to hear about the fragilities overengineering is creating. Those bosses can also set arbitrary deadlines that can rush a project, limiting time for carefully thinking through and testing for resulting fragilities.

An important lesson that Silicon Valley should (but almost certainly won't) draw from the Vasa is that overengineering because of FOMO will often be counterproductive. So, for that matter, will overengineering because of an abstract fear that tech companies must keep innovating or else die like a shark that stops swimming.

To be sure, increased complexity and fragility will sometimes be worthwhile if innovation is done in service of solving an important problem. But "worthwhile" is, as always, in the eye of the beholder. So much organizational and management research has flowed from the work of Schumpeter and Christensen, focusing on identifying the conditions in which firms become more innovative, and the environments within which the spread of innovation is more conducive. Recipes for things like "agile workflows" abound, but they don't tell us much about the best way in which to identify, generate, and spread the kind of innovation that is most likely to solve the problems that plague society. In fact, recipes for things like "agile workflows" might give Silicon Valley types the upper hand in disseminating their preferred kinds of solutions, at the expense of potentially superior ideas coming from outside the innovation-industrial complex.

One kind of overengineering that benefits tech platforms, but not the rest of us, results in the enshittification of technology products and services. "Enshittification" is a fantastic new word coined by Cory Doctorow – a deserving winner of the American Dialect Society's 2023 Word of the Year. It can be tempting to use enshittification to describe anything that's overengineered into getting crappier (mea culpa: I have definitely done this), but Doctorow had a more specific scenario in mind. He uses the word to describe how the usefulness of technology (particularly platforms) can decay as a result of profit-driven overengineering that negatively impacts users.

The first phase of building a technology platform requires attracting users – this is when it is engineered to be most pleasing to use (and use may also be subsidized by venture capital funding…coming up in the next chapter). Google's search engine, for example, started out as a delight to use, and remained so for many years – so much so that the word "google" became synonymous with online searching. The problem is, though, that once users are more or less locked into a platform and in the absence of other constraining factors like competition and regulation, the platform may give in to economic incentives to "innovate" for the benefit of the business customers (like advertisers) who actually provide the platforms' revenue – even if that makes the experience worse for its users.

Over the course of 2019 and 2020, amidst concerns about slowing growth, changes were rolled out to how ads were displayed on Google, resulting in customer complaints that "Google's ads look just like search results now." Even when they're not disguised as ads, many people (myself included) have noticed that Google's search results have gotten worse in recent years, prompting suggestions that "poor organic search results actually benefit Google's bottom line in two ways: they make paid advertisements more valuable to users seeking accurate information, and they force users to refine their searches multiple times, exposing them to more advertising in the process."

A degraded platform can really suck for its users, but enshittification-style tinkering won't necessarily stop there. If the platform's business customers become dependent enough on the platform, then *their* experience can also be made worse without too much risk of them going elsewhere – and so innovation can be directed at extracting as much profit as possible from business customers *and* end users. Cory Doctorow argues that Amazon has entered this phase of enshittification: it charges its sellers more and more to have their products placed before customer eyeballs; Facebook too has sought more and more free content from media publishers in exchange for promoting it on the platform. But hey, it's all innovation, right?

## The perils of underengineering

Not only can our collective innovation fetish excuse harmful overengineering, it can also result in harmful *under*engineering. Without consistent maintenance engineering efforts, software will eventually develop security vulnerabilities and fall into disrepair, resulting in glitches and outages. In *The*

*Innovation Delusion*, Vinsel and Russell argue that this critically important maintenance work is being devalued and delayed because of our societal fixation on new innovation. Because maintenance can never lay claim to being the sexy new thing, it is often neglected; when promises of future innovation are dangled as a solution to existing technology problems, maintenance is particularly likely to be ignored until underlying problems have metastasized into an emergency.

To give just one anecdotal example, I have a software engineer friend who works on a maintenance team at a tech company. There are plenty of software problems they need to fix, and they know how to fix them. What's stopping them is lack of resources – not enough people hours to keep up with all the work that needs to be done. So you can imagine my friend's chagrin when a large chunk of their team was poached internally to join an AI team that essentially roams the company asking "is there perhaps a problem here that we might solve with AI?"

Now, if you've read Farhad Manjoo's New York Times column *It's the End of Computer Programming as We Know It (And I Feel Fine)* or one of the other breathless reports about AI's ability to code, you might be asking yourself, can't AI help my beleaguered engineer friend? Well, AI can certainly produce a lot of lines of code quickly, but they're not always of good quality and they don't necessarily reflect an understanding about how those lines of code fit into the broader system – something that shouldn't really surprise us, given what we know from Chapter 5 about hallucinations in AI-generated text. Bill Harding, the lead on a report about AI's impact on code quality, put it this way: "hastily added code is caustic to the teams expected to maintain it afterward." So basically, my friend's maintenance job is going to get harder rather than easier as more software engineers embrace this AI-assisted "vibe coding" approach. I sent my friend a copy of Harding's report, and promised them a stiff drink.

I later read something that made me think a stiff drink wasn't going to be enough. Just as lawyers can get into trouble when AI hallucinates citations to non-existent court decisions, software engineers can get into trouble when AI hallucinates references to code libraries that don't exist. There will certainly be problems if code is directed to a non-existent library, but in an even worse case scenario, WIRED reports that enterprising hackers are "identifying nonexistent packages that are repeatedly hallucinated. The attackers would then publish malware using those names and wait for them to be accessed by large numbers of developers." In other words, hackers are turning hallucinated fake code libraries into real malicious code libraries in a type of cyberattack. Maybe – like the many Silicon Valley folks who enjoy a wee bit of ketamine – my friend will need something a little stronger to help dissociate from this reality…

Maintaining open-source software often comes with an extra degree of difficulty. If you're not familiar with open-source code, it's code written and maintained by volunteer software developers who make it available to the public for free. Many for-profit companies and government bodies use and customize open-source software for their own purposes, and (without many of us realizing it) we are all utterly dependent upon this kind of software in our day-to-day lives. As legal scholar and computer programmer Chinmayi Sharma puts it "Google, iPhones, the national power grid, surgical operating rooms, baby monitors, surveillance technology, and wastewater management systems all run on open-source software" (one industry study conducted in

2022 concluded that about three-quarters of *all* lines of code in use at that time were open source). Open-source code has therefore been compared to other kinds of critical public infrastructure, like roads and bridges, that allow the economy to happen.

The open-source software movement has been credited with being a force multiplier for technological innovation: it means that not everyone has to build their own code from scratch, and people familiar with a particular type of open-source code can transfer that knowledge from one project to the next. However, our obsession with constant innovation can undermine support for maintenance of open-source code. A particular open-source software language or code library might be wildly popular for a few years before people move on to the next new shiny thing – but the old version may still be incorporated as critical digital infrastructure into lots of things that people depend on. Where's the incentive to maintain it then, after it's lost its new car smell?

To some degree, businesses dependent on open-source code have incentives to ensure that it is maintained (or at least to maintain their own version of it). They will, after all, be deemed responsible for problems their users experience as a result of security breaches and outages arising from the open source code they've incorporated into their systems.

> Please indulge me in a mini-rant: when I criticize blockchains (like I did in Chapter 4), I'm often accused of hating on open-source software. But what I'm actually hating on is the lack of accountability associated with blockchain operations, because this lack of accountability is incompatible with something as high stakes as financial

> market infrastructure. Businesses that build on open-source code face reputational and potentially legal consequences if their operations are compromised by a problem in that foundational code. In other words, someone can ultimately be held accountable for poor vetting, deployment, or maintenance of the open-source code – not so with public permissionless blockchains, where we're told that no one can be held responsible for blockchain security or preventing outages because operations are [waves hands] decentralized. That's just an unequivocally terrible base structure for a financial system. Rant over, thank you for your indulgence.

Some businesses (particularly large tech firms) will pay bug bounties to hackers who identify vulnerabilities in open-source code that can then be patched; some businesses will donate employee time to maintaining open-source code. Some open-source projects are supported by well-heeled foundations – like the Linux Foundation, which maintains the open-source Linux operating system that is the foundation for Android smartphones and kids' Chromebooks.

But "some" is the operative word in all of this. Not all open-source projects get as much love. Many rely on just a few volunteer developers for maintenance (which isn't always fun – it often entails a never-ending stream of GitHub notifications from people with questions, problems, requests to add new features, and not all of those people are nice about waiting patiently for a response). In 2013, for example, it took several days for a security flaw in RubyGems (a library management system for Ruby, a huge software programming language at the time) to be fixed after it was discovered, because it was maintained entirely by volunteers. The server was hacked in the interim and had to be rebuilt from scratch, a process that was prolonged because, again, it depended on the work of volunteers with limited time. Some open-source projects are abandoned entirely as their developers get burned out or move on to shinier pastures. If so, there will be no one to respond to reports of bugs, or to vet changes to the code that may have been proposed by someone with bad intentions or bad programming skills.

Awareness is slowly increasing about the cybersecurity risks associated with inadequately maintained open-source software; the Log4shell vulnerability discovered in 2021 particularly freaked people out. Log4j is popular open-source code used to audit and debug applications written using the programming language Java, and the Log4shell vulnerability could be exploited to mess with Log4j code. As Chinmayi Sharma explains, once the vulnerability was publicized, "[c]ompanies like Apple, Amazon, Cloudflare, IBM, Microsoft, and Twitter began experiencing a barrage of attacks and many had no choice but to shut down systems until the vulnerability could be resolved. The Belgian and Canadian governments had to do the same." This is a book about fintech, so it would be remiss of me not to note that banks were reportedly targeted through the Log4shell vulnerability as well. There were particular concerns that it would be used to introduce malware allowing bad actors to steal bank customer login details. As the American Banker reported, "after stealing login data, attackers can send fraudulent automated clearing house and wire transfers, open fraudulent accounts and potentially hijack victim accounts for other scams involving business email compromise or money-mule activity." This kind of malware is particularly insidious because the bad guys might not take advantage of it immediately – it can lie dormant for a very long time, meaning banks needed to increase surveillance of customer accounts until they could figure out all the places where Log4j was being used and deploy the security patch (which was developed, in this case, by a foundation).

## Are our brains (and the tech media) conspiring against us?

In the end, innovation speak will only get you so far: cybersecurity and other types of maintenance work need to be prioritized if we want software to keep working. Even if a particular technology doesn't solve the problems it's supposed to, it needs to at least *function* or it will eventually tell on you. But what about technologies that do sort of function, but don't live up to the hype – why do we give so much credence to innovation speak in these instances? For example, why do people still give the blockchain the benefit of the doubt after more than fifteen years of lackluster results? Why does Elon Musk have any credibility with regard to Tesla robo-taxis, when he's been promising "they'll be on the road next year" every year since at least 2020? Why does anyone listen to Sam Altman when he tells us that AI will make "fixing the climate, establishing a space colony, and the discovery of all of physics" commonplace efforts?

The answer lies partly in our brains, which evolved for a simpler world than the one we live in today. Some of our brains' evolutionary hangovers can cause problems in the modern world: that's the high-level insight yielded by decades of work by cognitive psychologists Daniel Kahneman and Amos Tversky, summarized in Kahneman's best-selling book *Thinking, Fast and Slow*. Over their decades working together, Kahneman and Tversky catalogued an entire menu of "cognitive biases" that come very naturally to us, but can cause us to act in ways that are not in our best interests given the complex realities of the world we live in. In fact, these cognitive biases come so naturally that it can sometimes feel unnatural (and exhausting) to fight them – and several of them help predispose us to techno-solutionism.

Take the availability bias, which causes us to judge the likelihood of something occurring (for example, the likelihood of a technology successfully solving a problem) by the ease with which we can think of examples of similar things happening. Our lives are filled with mostly enjoyable interactions with technologies that *do* work – like most people, I'm usually inseparable from my smartphone (although I'm getting better at leaving it at home now and then). Most of us have far less exposure to tech failures for the obvious reason that we never get our hands on them. Examples of technology successes are therefore much more "available" to us than examples of failure. Even science fiction stories can prime us to think that technological possibilities are limitless notwithstanding that science fiction is, you know, fictional.

If we try to gauge the probability of any given tech solution succeeding by dividing an exaggerated numerator of success stories by a denominator that doesn't accurately capture the number of technology failures, then we're going to overestimate the probability of a technology's chance of success. The availability heuristic can also lead us to underestimate the harms associated with a technology that we believe might succeed: Kahneman explains that in one experiment, "people who had received a message extolling the benefits of a technology also changed their beliefs about its risks. Although they had received no relevant evidence, the technology they now like more than before was also perceived as less risky.”

The media plays a particularly important role in perpetuating this techno-solutionism through its breathless and often uncritical coverage of supposed tech breakthroughs – some journalists go as far as simply publishing lightly-edited industry press releases. How many headlines have you seen about the impending AI revolution, for example? Now how many of those stories mentioned basic facts about how costly AI is to run, its inaccuracy problems, or environmental damage? If AI downsides *were* mentioned, were they about the potential for human extinction or mass job losses – the kind of criti-hype that makes AI seem more powerful than it is? Constant repetition of this kind of coverage can build a (fake) sense of truth that a particular technology is indeed a sliver bullet that can decimate any problem it confronts. Unfortunately, once people have formed a bullish view of a technology’s capabilities based on effusive early reporting, subsequent evidence to the contrary (like the evidence of AI’s limitations that we covered in Chapter 5) will often seem less compelling to them. After all, it’s often more pleasant to think of things succeeding than it is to think of them failing.

Effusive tech coverage also conditions us to expect more of the same “genius” from Silicon Valley in the future. Kahneman and Tversky came up with the term “hot hand fallacy” to describe our tendency to incorrectly interpret past success as predictive of future success. We have seen enormous strides in tech innovation in the last few decades, and so we assume that Silicon Valley’s growth will always continue apace – even though it’s entirely possible that Silicon Valley, at least in its current modus operandi, has already solved most of the problems it is well-suited to solving.

Fawning media coverage often extends beyond the tech solutions themselves to the tech founders who peddle them. When media coverage describes these people as exceptional, it then makes it harder for others to question tech solutions associated with such exceptional people (something that Kahneman calls the "halo effect"). We rarely see this orthodoxy challenged, to the point that it feels downright shocking to see tech critic Ed Zitron write the sentence "there is nothing special about Elon Musk, Sam Altman, or Mark Zuckerberg." Why does it feel shocking? Well, as Kahneman and Tversky demonstrated with their work, our brains don't really like to embrace randomness. Instead, we find it much more satisfying to hear stories where merit (rather than a degree of luck) explains why a particular tech founder succeeds – and then we extrapolate from those stories a prediction that every technology or business model they touch in the future will also turn to gold.

Critically interrogating the successes of the Musks, Altmans, and Zuckerbergs of the world requires us to shatter our comfortable narratives of Silicon Valley meritocracy. As Zitron goes on to argue:

> *Accepting that requires you to also accept that the world itself is not one that rewards the remarkable, or the brilliant, or the truly incredible, but those who are able to take advantage of opportunities, which in turn leads to the horrible truth that those who often have the most opportunities are some of the most boring and privileged people alive.*

Many of us have assumed that technologies that have succeeded commercially must be superior to alternative solutions, and that the people who developed those technologies must be superior to other kinds of people. But if other things explain those successes (things like luck and privilege and the types of subsidies and lobbying we'll talk about in coming chapters), then our brains are fooling us when they extrapolate from past successes to predict that a future techno-solution will succeed in fixing a problem.

Scores of researchers have added other cognitive biases to Kahneman and Tversky's list over the years, and some of these also help explain our predispositions to techno-solutionism. Behavioral economists like George Akerlof, for example, have often focused on "hyperbolic discounting." This documented preference for immediate rewards, even over bigger future rewards, helps explain some of the policy decisions made around techno-solutions. If our public policymakers discount the rewards of the real solutions they could eventually achieve, preferring instead the short-term quick fixes being offered by Silicon Valley, then they're going to be reluctant to get in the way of techno-solutions and be discouraged from taking practical steps towards solving problems on their own.

Something called "automation bias" is also at play in all of this. Our brains are often quite happy to defer unquestioningly to the output of computers, because they assume that the answers generated by computers will be more accurate and legitimate than anything we flawed humans could come up with. This can lead us to believe that technological innovations will do a better job of solving problems than other approaches. The truth is, though, that computer programming is messy, it's widely considered impossible to develop bug free code, and we likely wouldn't be so deferential to technology if we knew how the sausage were made. Software engineers remain flawed human beings who sometimes wing it, just like the rest of us – as Ellen Ullman put it, "it has occurred to me that if people really knew how software got written, I'm not sure if they'd give their money to a bank or get on an airplane again." Technological development is not a precise science, but our collective automation biases can make it feel like it is.

## Techlash

To sum all of this up, it's often much easier to believe that a tech business will live up to the hype than it is to think through all the different ways it could fail to deliver. As Kahneman says, "disbelieving is hard work," and so, dear reader, it turns out that I've been asking an awful lot of you as we've unpacked together the many ways in which fintech is failing us. But it's not guaranteed that seeing through tech hype will always be such hard work. Kahneman and Tversky also underlined the importance of "framing effects:" our decisions are influenced by how issues are presented, and so if the framing of our conversations about technology changes, it could become cognitively easier to reject techno-solutionism than to embrace it. The availability bias perpetuates techno-solutionism, for example, because our minds have so many positive interactions with technology in our daily lives to draw upon. But if Silicon Valley keeps overpromising and underdelivering, and if tech businesses continue to be overengineered and enshittified, then the availability bias may start to cut in the other direction (particularly if critical media coverage of Silicon Valley increases).

Now, I don’t want to overstate the likelihood of such a correction. What David Nye has called “the American technological sublime” is a central tenet of our national identity and story and it will not be lightly abandoned – belief in technological innovation is as American as apple pie. But the American Dream is also a central tenet of our national identity and story and yet, in the wake of the 2008 financial crisis, there has been an increasing realization that the American Dream is simply not available to everyone. If we can see the cracks in the American Dream, maybe we’ll also start to see cracks in the American technological sublime: one Pew Research survey found that the share of Americans who saw technology companies as having a positive impact on the United States tumbled from 71% to 50% between 2015 and 2019.

The Silicon Valley elite benefit mightily from innovation worship and techno-solutionism, though, and they’re not going to give in without a fight. Next chapter, we’ll see who we’re up against…