

Source Link: <https://blog.pragmaticengineer.com/section-174/>

by **Gergely Orosz**

The Pulse: Will US companies hire fewer engineers due to Section 174?

👋 Hi, this is [Gergely](#) with a bonus, free issue of the Pragmatic Engineer Newsletter. In every issue, I cover topics related to Big Tech and startups through the lens of engineering managers and senior engineers. In this article, we cover one out of four topics from today's subscriber-only [The Pulse issue](#). To get full issues twice a week, [subscribe here](#).

In October, we looked into [bootstrapped companies founded by software engineers](#) and the article resonated with many readers; I got lots of messages from bootstrapped founders following that issue. Many messages were complaints about something called "Section 174 changes." Here's what one founder said:

"Have you heard of a change in Section 174 of the US tax law that kicked in a few years ago? This change is making bootstrapped software businesses completely unsustainable.

Basically, all costs related to R&D cannot be expensed, including labor for software development. These costs have to be capitalized and amortized over 5 years – or 15 if labor is done outside of the US. I have no other way to put it: this change is completely insane. Everyone I talk to says the same.

I am curious if this has come up in your discussions with other bootstrapped companies?"

I did some research, and The Wall Street Journal and a handful of other news outlets have [covered](#) this change since last March. Still, founders reaching out shared that it feels like public awareness is low about just how large a problem this tax change is. Back last April, Ben Thompson at Stratechery [covered](#) this change, also expressing his surprise on how impacted companies seemed to be blissfully unaware of this regulation:

"I've been very surprised that few people in tech seem to know about this issue, at least in the private conversations that I have had, even though startups are arguably the hardest hit companies of all."

I [took to social media](#) to gather more information, and a surprising number of complaints from US tech businesses came in.

In this issue, we cover:

- An unexpectedly high tax bill out of nowhere
- A tax change never intended to make it into law
- Why did Big Tech and VCs not raise the alarm?"
- The impact of this change on US tech companies and devs working at them
- Why is forced amortization of software labor objectively bad?
- Will the US be content being a less competitive place for startups?

An unexpectedly high tax bill out of nowhere

Many US software businesses amassed surprisingly high tax bills in 2023, seemingly out of nowhere, due to a tax change which took effect in July the previous year, which many small companies knew nothing about until finalizing their 2022 returns. The change was expected to be repealed (reversed) in December 2022, so many accountants didn't inform customers for that reason. So, businesses got a surprise when the first tax payments fell due last April.

The amendment to S174 means employing software engineers can no longer be accounted as a direct cost in the year they are paid – unlike the norm, globally. Here's a simplified example of the change from the final tax year before the

change.

Take an imaginary bootstrapped software business called “Acme Corp.” This company generates \$1,000,000 of revenue per year running a SaaS service. It employs five engineers, and pays each \$200,000. That is \$1,000,000 paid in labor costs. *For simplicity, we omit other costs like servers and hosting, even though those costs can also fall under the new R&D rules, and have to be amortized.* So, how much taxable profit does this company make?

In 2021, the answer would be zero profit. In 2022, the answer was \$900,000 in profits(!) This is because from 2022, software engineer labor costs *must* be amortized over five years. Here is how amortization works:

- 10% amortized for the first year
- 20% amortized for years 2 to 5
- 10% for year 6

Let’s break this down:

	2021	2022 (Section 174 changes)	2023 (Section 174 changes)
Revenue	\$1,000,000	\$1,000,000	\$1,000,000
Software engineering labor costs paid during the year	\$1,000,000	\$1,000,000	\$1,000,000
Of which deductible for the financial year	\$1,000,000	\$100,000	\$300,000 (10% from this year 20% from last year)
Profit	\$0	\$900,000	\$700,000
Corporate tax due	\$0 (due in 2022)	\$189,000 (due in 2023)	\$147,000 (due in 2024)

pragmaticengineer.com

The impact of Section 174 changes on a company that has \$1M in revenue and \$1M in labor costs.

What happens to a small, bootstrapped company without \$189,000 in cash floating around with which to pay this tax? There are two options:

1. Take out a loan at a relatively high interest rate (likely at around 10% or so.) *I’ve heard of small businesses needing to use personal loans to pay tax.*
2. Lay off a software engineer on \$200,000 per year and use this saving to pay the bill!

Less hiring of software engineers and more software engineering layoffs are already happening at smaller US tech companies. In January last year, in a Wall Street Journal article, policy analyst, Alex Muresianu, [estimated](#) this tax code change would cost the US about 20,000 full-time software engineering jobs. And I have more real-world accounts of how US businesses are being hit, straight from the companies themselves:

- A business [posted](#) a \$90K loss in 2022 by the old rules, but is taxed on a \$1M profit.
- Infra startup Gruntwork is [hiring less](#) in 2024 because of the change.

- A startup that raised \$1.25M and hired five employees, is [likely to face](#) a \$150,000 tax bill it did not account for.
- A seed-stage startup that raised \$3M and was operating at a loss became profitable due to the S174 change, so must budget for higher taxes.
- Another US company let go of 23 engineers employed in India because of the tax change. Software engineers abroad can be amortized only over 15 years.

A tax change never intended to make it into law

So, what are “Section 174 changes?” We need to go back some years to understand. In 2017, then-President, Donald Trump, signed the 2017 Tax Cuts & Jobs act, which overhauled tax codes and reduced tax – for example, it reduced the top tax bracket from 39.6% to 37%. To make the bill pass strict budgetary rules, the Senate used a process called reconciliation: adding in tax code changes that delayed tax increases. These delayed increases “balanced out” the tax reduction.

One of these changes was Section 174, set to come into effect 5 years later, in 2022. These parts [deliver the blow](#) by making it clear that software development costs *need* to be amortized over 5-15 years. Most experts expected Congress to push back the Section 174 amendment to a later date, or simply remove it. But Congressional negotiations to repeal the changes fell apart at the last minute in December 2022, meaning it became law.

Why did Big Tech or VCs not raise the alarm?

Why did we not hear the largest tech companies protesting the tax change? Actually, several did, but their protests failed to cut through on the news agenda.

Large tech companies tend to speak up about issues like this via coalitions, trade associations, and lobbyists. Amazon, Microsoft, Intel, Ford, Lockheed Martin, and other US companies created the US [R&D Coalition](#) in 2018 to advocate in reversing this change. This group [concludes](#):

“[The Section 174 changes are] a dramatic shift in the tax treatment of business investments in research and innovation, and it will leave the United States with a system unlike any other in the industrialized world.

By diminishing the near-term value of R&D expenditures, the Tax Cuts and Jobs Act will reduce incentives for companies to invest in the development of new products, ultimately hurting consumers and businesses alike.”

Here is how the S174 change impacted some companies, based on what I found in their annual reports:

- **Microsoft:** \$4.8B additional tax paid in 2023. *The company generated a \$72B profit that year, so this tax increase was manageable. It's still a very large amount!*
- **Netflix:** around \$368M in additional tax paid – *also manageable with \$4.4B annual profit.*
- **Google:** the tax change was minimal, because Google was voluntarily amortizing software development expenses for most staff, already. This was for all projects that reached “technological feasibility,” which is a milestone products pass before public release.

For companies with large cash reserves, the tax change was inconvenient, but manageable. Over a five-year period, the amount of tax evens out. After five years, there can even be tax benefits to this kind of accounting.

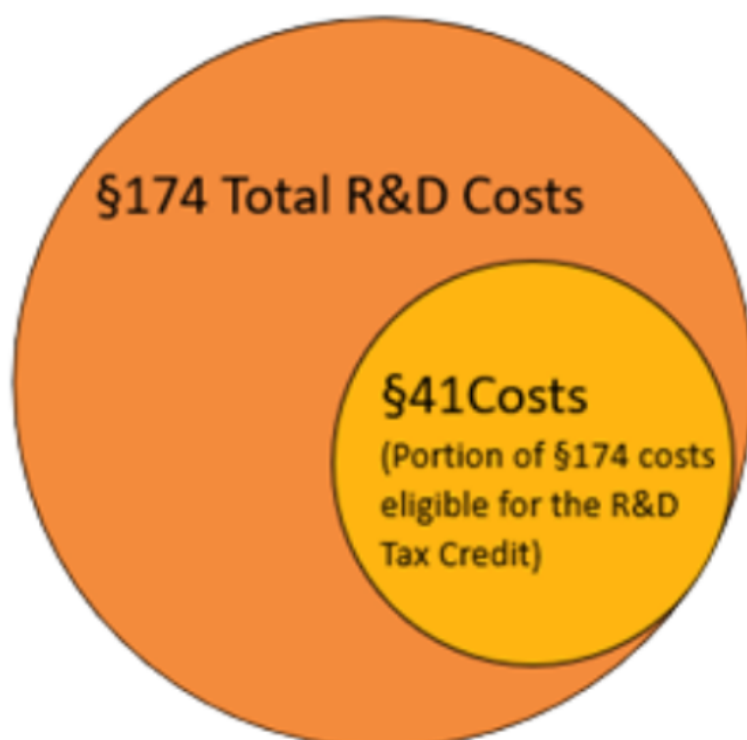
What about VC-funded companies? For loss-making companies this change doesn’t make much of a difference. But the change does impact VC-funded companies near to break even. Most VC-funded companies close to breakeven have big-enough cash buffers with which to pay unexpected tax bills. However, these companies might reduce hiring – or even consider letting go staff.

Impact on US tech companies and devs working at them

Assuming Section 174 stays and all US companies *must* amortize software developer costs over 5 years, this will have an immediate impact on early-stage and small tech companies:

- **Less hiring of software engineers in the US/more layoffs.** Small and medium-sized tech companies facing higher tax bills in 2023 and 2024 will hire fewer developers. Redundancies might happen for cash flow purposes to replace in-house developers with vendors, which brings us to the next likely impact:
- **Firing of non-US software engineers employed by US companies.** The tax change is very hostile to software developers employed abroad: their wages need to be deducted over 15 years. Unless a US company has massive cash reserves, it now makes no sense to remotely employ or contract with individual software developers. An engineer shared how their company fired 23 developers employed in India *because of* Section 174.
- **A boon for SaaS companies and vendors?** US software companies now have a strong reason to buy, instead of build, software in-house. In February last year, we covered the [trend of tech companies aggressively cutting vendor spend](#). This tax change greatly incentivizes US companies to increase vendor spend – and either not hire more devs, or let some go!
- **It makes a *lot* less sense to incorporate tech startups in the US.** Assuming there's a choice to incorporate a startup in the US or somewhere else, then any other country makes so much more sense. In the first few years of a startup, it's typical to make a loss while building something that might not work out. Thanks to the amortization rules, in the US this loss could turn into a taxable profit!
- **Startup software IP will likely be moved out of the US.** Foreign founded venture-backed startups have been overwhelmingly incorporated in Delaware, US, until now. The Delaware company would then hold the intellectual property (IP) of the startup. With the Section 174 changes, the simplest way to not be subject to US amortization requirements is to move the IP to a foreign subsidiary – and then this subsidiary can employ developers, without being subject to the 15-year amortization requirements. This is what newly founded VC-funded startups are doing, already! *As usual, this is not legal or business advice. Consult a professional for guidance on this matter.*
- **Creative workarounds, like "inversion buying."** Say that you run a US company with a Canadian subsidiary, with most developers working in Canada. With Section 174 changes, Canadian developers need to be amortized over 15 years (!). A tempting workaround is to have the Canadian subsidiary buy the US company! Now this is a Canadian company, and Section 174 doesn't apply to developers in Canada. However, such workarounds can be expensive and complicated from a tax perspective, due to Passive Foreign Investment Company (PFIC) rules. Still, plenty of founders are exploring setups like this.

Tech companies are now forced to understand – and use – R&D credits. The S174 amendment forcibly categorizes pretty much all software development as R&D. The upside of this that R&D credits do offset the tax bill, but only partially:



R&D costs eligible for the R&D Tax Credit are typically a smaller subset of all R&D costs. Image source: [Aprio Insights](#)

For more details on tax credits and Section 174, Gusto created a [Section 174 tax liabilities calculator](#). *Do consider getting professional advice, as always, if you are running a business impacted by this change.*

Innovation across all US software companies will take a hit if Section 174 stays. The tax change incentivizes software companies without large cash reserves to invest *less* in research and development. Or they can just move abroad.

But the change isn't just bad for small software companies; it hurts even the largest ones – it is why Microsoft and Amazon are also advocating for its reversal.

Why is forced amortization of software labor objectively bad?

The concept of amortization (also referred to as depreciation) is as follows. Say you purchase a powerful server for \$2,000 to serve traffic for your SaaS business. You pay \$2,000 on the spot. However, the expected lifespan of this server is 4 years, after which you expect to replace it. Therefore, it is reasonable to argue that you are getting about \$500 worth of "value" every year from this server, and so you amortize (depreciate) a quarter of the value of this server every accounting year, until year 4.

This kind of depreciation is sensible, as you could sell your server for around \$1,000 after two years (assuming that is how its value depreciation.) Tax authorities in different countries have come up with depreciation frameworks that businesses follow along these lines. For example, in the US, the [Financial Accounting Standards Board](#) (FASB) created the Generally Accepted Accounting Principles (GAAP) provisions, which include a framework on how to deal with depreciation.

Amortizing software development over years makes sense – but only in some cases. If you have launched a software product, have customers, and you can forecast demand for this product pretty accurately, then investing developer hours to keep this SaaS running feels similar to buying hardware. The software engineer could build a new feature for the SaaS which generates revenue for years to come. So there is *some* argument to treat software development akin to buying a server.

But is comparing physical goods to software, accurate? Take a company that buys a server office it plans to use for 20 years. After a massive one-off investment, the only cost is maintenance, which is a pretty minimal cost.

Now take the example of creating software to generate revenue for 20 years. If you do barebones maintenance, the software becomes obsolete in a year or two! You need to keep adding new features and keep up with the competition. As you add more features, you run into tech debt and architecture issues, so you refactor the software and within a few years you rewrite the whole thing!

Building software is like continuously renovating – and rebuilding – a property. While it is tempting to see why an accountant would want to treat software development amortized like physical goods, this is simply not the reality of how software products behave, or are maintained.

Before Section 174, companies could *choose* how they categorized software developers, and could opt into deducting costs. This is what stable and highly profitable businesses like Google have done: for software products in production (and making money) it deducted developer costs over 5 years. For pre-launch projects, Google simply expensed those developers. This is the sensible way to run a software business, after all!

Google employing lots of devs in Switzerland starts to make a lot more sense, especially now. Ever wondered why Google has such a large software engineering center in Switzerland – despite the high cost of software engineers within Europe? Switzerland has a very powerful research and development incentive: the country [allows expensing](#) 135% of R&D-related salaries in the year they are incurred.

Switzerland's taxation is pretty much the polar opposite of that of the US with Section 174. Is Section 174 stays, I would not be surprised if some US companies considered following Google's lead and explore setting up engineering

offices in this country.

Will the US be content being a less competitive place for startups?

All founders whom I talked with hoped Congress would revert this disastrous tax code change, given its devastating impact on tech startups. However, this has not happened – and it's unclear if it will.

Founders have been trying to get the attention of Congress: sending a [letter signed by 1,000+ software businesses](#), and creating coalitions like the [American Innovators Coalition](#). Another group formed in March this year is the [Small Software Business Alliance](#), created by indie SaaS founder Michele Hansen, supported by over 600 small software company founders.

I asked Michele why she thinks the general public isn't aware of the threat to US companies. She said:

- "1. For years, the perception has been that tax changes impact big companies, not small ones.*
- 2. Tax policy issues are boring for the general public. It takes a strong social impact – for example, mass layoffs – for news like this to break into mainstream media.*
- 3. Section 174 was never supposed to take effect, and was widely assumed that it would be fixed before it took effect. So, there was little interest in a problem likely to go away."*

Michele said there is reason for hope, though. Congress appears to have struck a deal: revert Section 174 changes on R&D expensing. Lawmakers are now [looking for ways to finance this deal](#).

One country making software development less competitive obviously benefits other countries. Thanks to the S174 amendment, starting a tech company *outside* the US is far more attractive – including for raising startup capital. Still, the US has long been the global innovation hub for technology, and the country kneecapping technological innovation might let the rest of the world catch up. But I cannot see this being beneficial for innovation, as a whole, globally.

Thank you to [Michele Hansen](#) and [Jacek Migdal](#) for their insights and reviewing a draft of this section.

Updates to the article after publishing: 6 January — added a reference to [Stratechery](#) that covered this change last April