TPRI Report

Why AI hasn't killed software developer jobs

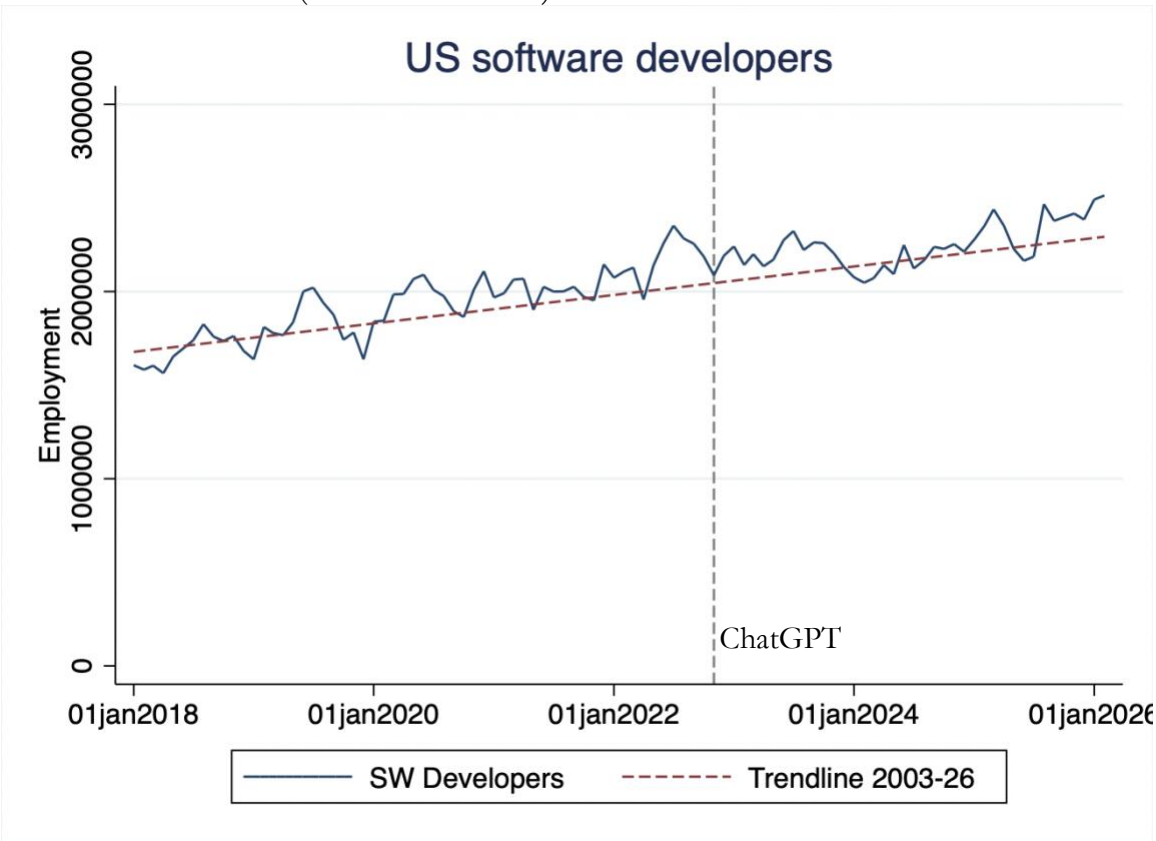
James Bessen
March, 2026

Why AI hasn't killed software developer jobs

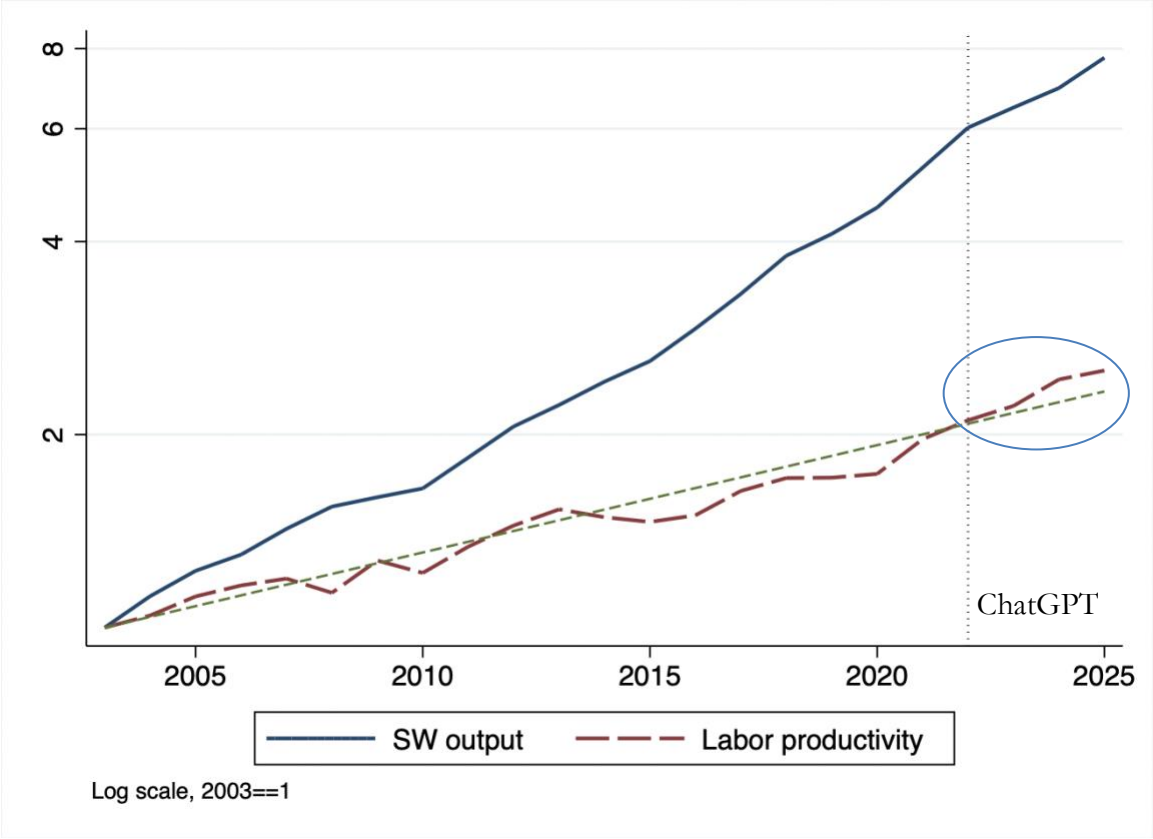
James Bessen, March 31, 2026

Each week new examples emerge of AI performing software development tasks, spurring predictions of mass layoffs of developers, perhaps before the end of the year. And software is a bellwether—if AI can replace software developers, it will surely become "a general substitute for cognitive work," as expressed in a viral blog post by Matt Schumer that rattled financial markets. Many fear a "job-pocalypse" of mass unemployment.

Surprisingly, however, after three years of AI use, software developer jobs have continued to grow robustly, reaching record levels of employment (2.5 million in February). The number of software developers in the US has grown by over 400,000 since ChatGPT was introduced in 2022 (vertical dashed line).



Why isn't this powerful new technology eliminating these jobs? An important and possibly disruptive change *is* happening, but the common view misunderstands what is going on, and that matters for policy makers and businesses. Careful case studies find that AI improves the productivity of software developers—that is, the software produced per developer—by 30 percent, 50 percent or more (see Alex Imas). And aggregate data I have assembled provide evidence—preliminary to be sure—that the rate of productivity improvement in software development is improving.



The dashed line in the figure shows the labor productivity of software developers measured as US deflated output of software divided by the number of software developers (data here; see also Byrne and Corrado and Fleming on comparable price declines). Since 2022, when ChatGPT was introduced, developer productivity has ticked up noticeably. From 2003 to 2022, developer productivity grew at 3.9 percent per year; but from 2022 through 2025, it grew at 6.0 percent per year. While the recent data series is short and therefore only suggestive, it does reinforce the evidence from case studies.

But the number of employed software developers has continued to increase, demonstrating that productivity growth does not necessarily imply job losses. The reason for this is also apparent in the chart: the real output of software has been increasing also, much faster than productivity at about 9.3 percent per year. This implies that demand for software is increasing. If demand is increasing faster than the rate of productivity growth, the number of developers must increase even though the number of developers needed to produce a given quantity of software has been dropping.

Too many people have an impoverished view of what AI does. Automation doesn't simply substitute machines for humans, doing the same things in the same way; it also lowers costs and prices, it improves product quality, and it enables new and better products. Technology, including AI, enables developers to come up with new or improved products that consumers want. When this happens, total demand for software increases, driving output up. When the output increases enough (demand is "elastic" enough), total employment increases even though the amount of labor needed to produce a unit of output goes down. So, improved software productivity has brought us lower prices, better quality, a flood of new products doing new things, and…more developers.

This has been happening in software for decades. AI is not the first technology to boost developer productivity. Developers have gotten new tools, structured coding, automated testing, Agile Development and DevOps, cloud computing, Open-Source software and more. From 2003 to 2025, the real output of software per developer more than doubled, rising 150 percent. But total software output increased even faster. Given this history, it is hard to see why a mere 50 percent improvement in developer productivity would not similarly increase jobs.

While the elastic response to developer productivity has been stable for decades, it might change, possibly leading to job losses. Indeed, we have seen such changes historically. During the 19th and early 20th centuries, technology drastically reduced the amount of labor needed to produce cloth, steel, and automobiles. But employment in these industries grew rapidly for many decades. Until it didn't. In textiles, for example, automation drastically eliminated most of the labor needed to produce cloth during the 19th century. The price of cloth fell dramatically, bringing a large increase in demand that spurred employment. This went on for over a century, but by the mid-20th century demand satiated; closets were full, textiles were used in a wide range of applications beyond clothing, so lower prices no longer elicited much additional demand. Something like that could happen in software if, for example, there was no longer active demand for new or improved software products. Software employment could also fall if software could be produced without *any* human labor. For the near future, it seems, humans are needed to design new products and to verify that software fulfills the design, so the flood of new software products, perhaps enhanced by AI, will continue to create jobs for developers.

Thus, mass unemployment of software developers seems unlikely to happen soon. Nevertheless, AI is dramatically changing the nature of software jobs, and this will likely continue (see Tim Lee). This implies that organizations will need to restructure, and workers will need to endure transitions to new roles and to acquire new skills. Indeed, the occupation title of "computer programmer" has been declining for decades. And other occupations, where demand for more output is not so elastic, might be at risk of substantial loss of jobs.