Appendix 2. Who captures the savings from AI in software development

Egor Kugno. Data checked as of 30 October 2026.

When AI writes the software, the cost of delivering a product falls. This appendix uses calculations and data to show who gets the difference: the customer, the contractor, the compute provider, or the laid-off engineer. Tables 2 and 5-8 are the author's models with assumed inputs. The other tables rely on cited sources.

1. What a person with a good job loses

At a large technology company, bonuses and stock are added to base salary. An employee receives new stock grants year after year, so a job change has to be compared on total annual compensation.

A Microsoft posting for Software Engineer II and Senior Software Engineer, published on 7 October 2026, lists these base salary ranges [1].

Table 1. Base salary in the Microsoft posting, before tax.

Level in the posting	Main US range	San Francisco and New York metro areas
Software Engineering IC3	$102,100-202,200	$133,800-219,200
Software Engineering IC4	$119,800-234,700	$160,200-261,000

These are the employer's ranges for one posting. Another source shows total compensation: according to Levels.fyi as of 29 October, the median annual package for a Software Engineer at Microsoft in the US is $218 thousand, including salary, stock, and bonus. The platform collects data from the professionals themselves [2].

The US Bureau of Labor Statistics (BLS) shows the scale of the American market. In May 2025, the median annual wage of a software developer was $135,980. The lowest-paid 10% earned less than $82,460, and the highest-paid 10% earned more than $214,670. This is cash pay under BLS methodology; stock and some irregular payments are not included [3, 4].

The spread is wide even within one profession. A person can stay a developer and earn much less than at the former corporation. A job posting alone does not show whether the former income will return.

In the model below, public-company stock is counted at the value of the part the employee receives in a year. All amounts are before tax.

Table 2. Keeping the salary and losing a third of total pay. Author's model, thousand dollars per year.

Package component	Former job	Offer A	Offer B
Base salary	200	200	180
Annual bonus	40	20	18
Stock received in the year	100	0	40
Total	340	220	238
Change from the former package		-35.3%	-30.0%

In Offer A the salary is kept in full, but the person loses $120 thousand a year before tax. Monthly income changes comparatively little, while the money that used to go into investments, a down

payment, or help for the children shrinks sharply. Offer B includes both a bonus and stock, but the total is still $102 thousand a year lower than before.

This is what a pay cut looks like without leaving the profession. The person is employed again and does not appear in unemployment statistics, but can no longer save and invest the former amounts. A return to full annual income is a better measure of how well someone came through a layoff.

2. What has changed in the work itself

An agent gets access to a project, edits files, runs checks, and goes back to fix errors. A person sets up the working environment and defines the conditions under which a result can be accepted.

On 5 February 2026, OpenAI said that early versions of GPT-5.3-Codex were used to debug the model's own training, manage its deployment, and diagnose test results [5].

The same day, Anthropic published an experiment by Nicholas Carlini: a team of 16 agents built a C compiler in Rust that can compile the Linux 6.9 kernel. The work took about two weeks and nearly 2,000 sessions, with API costs of roughly $20 thousand [6]. Carlini built the tests, the feedback, and the conditions for the agents to work in parallel. The machine code the compiler generates is less efficient than GCC's, and the $20 thousand does not include the researcher's time or the project's infrastructure.

In this kind of work, the engineer describes the behavior the program needs, builds the checks, and works out the consequences of machine-made changes. Their contribution has to be judged by the quality of the working product, not by the number of files created.

3. Where faster code ends and shipping a product begins

Fiona Chen and James Stratton studied 655 firms from January 2021 to August 2026. The 28 October revision of their paper estimates the additional effect of agents 12 months after adoption, relative to using assistants [7].

Table 3. Estimated changes after adopting agents.

Measure	Change
Lines of code added	+44.0%
Commits (saved changes)	+36.0%
Pull requests	+35.7%
Calendar time from submission to approval	+56.1%
Jira issues closed	+13.7%
Jira epics completed	+18.0%
Number of active engineers	-2.6%

The estimates for lines of code and closed issues are borderline statistically. The estimates for epics and employment are statistically indistinguishable from zero [7].

In a separate paper presented in June, Mert Demirer, Leon Musolff, and Liyuan Yang studied more than 100 thousand GitHub developers. Using several generations of tools together is associated with roughly 180% more coding activity and roughly 30% more releases. In the app stores they examined, supply grew, but usage of new apps has not noticeably increased. The authors name review, integration, release, and limited user attention as the later constraints [8].

A developer still has to test the change on the customer's data and confirm it works with the existing infrastructure. The first version gets cheaper, but review, integration, and maintenance remain, and

they take a growing share of the cost.

In their model, Chen and Stratton look at the future of review. The result depends on assumptions about technology and demand [7].

Table 4. Scenarios in the study's model.

Assumption	Product output	Employment
Technology fixed; the firm finishes adjusting to it	+16.1%	-13.2%
By 2031, writing speeds up and review productivity stays fixed	+57.8%	-10.3%
By 2031, review speeds up along with writing	+341.4%	-76.2%

4. What an accepted result costs

A customer needs a finished task, such as a new feature in production. Its cost is made up of labor, compute, and later fixes. Take a hypothetical release that requires 100 person-hours, at a fully loaded cost to the employer of $100 an hour, including costs beyond salary.

In the original process, people spend 30 hours writing, 40 hours on review and testing, and another 30 on integration and deployment. In the first variant, writing becomes ten times faster. In the second, review also becomes twice as fast. AI costs are assumed to be $600 per release.

Table 5. The economics of one release. Author's model.

Item	Original process	Writing 10x faster	Plus: review 2x faster
Writing, hours	30	3	3
Review and testing, hours	40	40	20
Integration and deployment, hours	30	30	30
Labor, total hours	100	73	53
Cost of labor	$10,000	$7,300	$5,300
Additional AI costs	$0	$600	$600
Total cost	$10,000	$7,900	$5,900
Cost reduction		21%	41%

In the first variant, 27 hours of labor disappear. They cost $2,700, so after paying for AI, $2,100 is saved. In the second, 47 hours disappear and the savings reach $4,100.

This sets a limit on what machine help can cost. With quality and volume unchanged, AI costs can reach $2,700 before they consume the labor savings in full. If a release needs expensive reruns or creates more fixes later, the margin shrinks. The cost of a finished release includes time for checks and rework, then the cost of maintenance after launch, and those costs show how much the company really saved on development.

A company can use the freed-up hours in three ways: increase output, shorten working time, or cut paid positions. Which one it chooses depends on demand. The worker's share of the savings depends on whether the company directs part of them to pay or to shorter working time.

5. Cheaper compute and a growing bill

In the 2025 AI Index, Stanford University estimated that the cost of using a model at GPT-3.5 level fell more than 280-fold between November 2022 and October 2024. This is a historical comparison of the price of the same level of performance [9].

The total bill also depends on how many operations are ordered. Suppose the price of one comparable operation in a hypothetical service falls by 80%, and the original bill is set at 100.

Table 6. The compute bill after an operation gets cheaper. Author's model.

Scenario	Operation price vs. original	Volume of operations	Total bill, index
Original	100%	1x	100
Same volume of work	20%	1x	20
Volume up 5x	20%	5x	100
Volume up 10x	20%	10x	200

New bill = old bill × relative price × relative volume. A fivefold rise in usage fully absorbs a fivefold price drop.

In development, several implementations of one task can be run, followed by additional checks and bug hunting. Spending on people can fall while spending on machines rises. Demand for GPUs also depends on overall demand, quality requirements, model architecture, and hardware efficiency.

The participants' interests diverge. The customer wants a cheaper finished product, the model developer wants to sell more compute, the company adopting AI wants to keep the gap between old and new costs, and the worker tries to hold on to the price of their own contribution.

6. How adjacent digital businesses change

Large technology companies buy services from development contractors, integrators, maintenance providers, and recruiters. These firms' revenue depends on how much human labor is bought and on the form of the contract.

Who gets the savings in custom development

Take a contractor that sells 10 thousand hours at $100. Its revenue is $1 million. Direct staff costs are $70 an hour, or $700 thousand. That leaves $300 thousand for overhead and profit.

After adopting AI, the same amount of work accepted by the customer takes 5 thousand hours. Direct staff costs fall to $350 thousand, and machine tools cost another $50 thousand.

Table 7. One result, different contracts. Author's model, thousand dollars.

Scenario	Revenue	Staff	AI	Remainder
Original 10,000 hours	1,000	700	0	300
5,000 hours, same hourly rate	500	350	50	100
5,000 hours, same fixed price for the result	1,000	350	50	600
5,000 hours, fixed price cut to $700 thousand	700	350	50	300

With hourly billing, the contractor's remainder falls to a third. With the price of the result unchanged, it doubles. In the last row, competition passed $300 thousand of savings to the customer, and the contractor kept its former remainder with half as much human labor.

With a fixed price, the contractor keeps or increases the remainder for overhead and profit while buying half as many staff hours. For employees, this means fewer paid hours on that order.

For large integrators, the subjects of negotiation become guaranteed results, deadlines, liability for failures, and later maintenance. The better a customer is at comparing offers, the harder it is for a contractor to hold on to the full original savings for years.

How a drop in hiring affects recruiting

A recruiting agency faces pressure from two sides: fewer filled vacancies and a lower salary on which the fee is calculated. Assume the agency earns 20% of the first annual salary.

Table 8. Recruiting revenue squeezed twice. Author's model.

Measure	Original scenario	After demand changes
Vacancies filled	100	60
Average annual salary	$200,000	$180,000
Fee per hire	$40,000	$36,000
Agency revenue	$4,000,000	$2,160,000
Change in revenue		-46%

Hires fell by 40%, salary by 10%, and revenue by 46%. The agency still has to pay its staff and office out of the smaller revenue.

Preparing a newcomer for real work

Courses can teach a language and produce a practice project. An employer assesses something else: the ability to work safely with a real system, where requirements are incomplete, the data is awkward, and a mistake affects a customer. Practice code is easy to get from an agent, so showing it does less to set a prepared candidate apart. Verifiable experience, understanding of the subject area, and the ability to take work through to production become more valuable. The first such experience is hardest to get for someone nobody has hired yet.

A business that sells a fast entry into a well-paid profession faces a costlier job: arranging practice and certifying the graduate's preparation. Recorded lessons scale almost for free, while mentoring, access to real projects, and review of finished work take people and time.

7. What is left of the career promise

A good job at a technology corporation brought high current income, stock, and the prospect of promotion. The employee in the Table 2 example loses part of their annual income and the ability to invest regularly when they change jobs. At a new job, the engineer may get a better-paid role or the same title with a smaller annual package. To assess the change, one needs to know how many professionals regained their former income after one, three, and five years, how many lost it for a long time, and how many graduates never got a first job.

Years of study and professional growth were supposed to make it possible to buy a home and build capital. If total pay falls by a third, a person can still cover current living costs, but will need longer to save the intended sum.

For a business owner, the same shift looks like a higher return on capital: fewer paid hours per unit of result sold. The gain is shared among the customer, the contractor, the model owner, and the compute provider. In Table 7, one clause of the contract decides how much of the savings goes to the customer and how much stays with the contractor.

A laid-off engineer has to choose from the offers available while savings cover the family's expenses. The longer the search, the harder it is to turn down a job with lower pay.

Notes on the data and calculations

The salary ranges refer to the Microsoft posting named above, the Levels.fyi data to the platform's user database, and the BLS data to a separate nationwide survey. The types of pay are labeled in the text. The BLS data refer to May 2025, and the Levels.fyi data were updated on 29 October 2026.

Tables 2, 5, 6, 7, and 8 are the author's models with explicitly stated starting conditions. They show mechanisms and arithmetic. Salaries, hourly cost, recruiter fees, and AI costs in these models are chosen for the calculation. Taxes, inflation, and changes in stock value are not modeled. The contractor's remainder in Table 7 is calculated before overhead and taxes; transition costs of cutting staff are also left out.

Table 3 gives point estimates from a working paper that compares firms with different adoption timing. Table 4 gives scenarios. Calendar review time includes waiting. Percentage changes are relative. The 28 October revision of the Chen and Stratton paper is used here; the August revision (718 firms) reports different estimates: roughly +30% lines of code and +49% time from pull request to merge, with a statistically insignificant rise in closed issues.

The Demirer, Musolff, and Yang paper exists in two versions: the VoxEU column (June 2026) reports a sample of more than 100 thousand developers, while a later revision of the NBER working paper expands it to more than 500 thousand developers and estimates the effect on commits at 240%, with the same 30% for releases.

The technical examples are described from their authors' publications. They show what has been built and how the work is organized; the sources give no comparable audited estimates of the full cost of these projects.

Sources

1. Microsoft. Senior Software Engineers + Software Engineer IIs. Employer posting on the AnitaB.org Talent Network, 7 October 2026. Microsoft application page.

2. Levels.fyi. Microsoft Software Engineer Salaries, United States. Updated 29 October 2026; median total annual compensation $218,000.

3. U.S. Bureau of Labor Statistics. Software Developers, Quality Assurance Analysts, and Testers. Occupational Outlook Handbook, pay data for May 2025.

4. U.S. Bureau of Labor Statistics. Occupational Employment and Wage Statistics: Frequently Asked Questions. Types of pay included.

5. OpenAI. Introducing GPT-5.3-Codex. 5 February 2026.

6. Carlini N., Anthropic. Building a C compiler with a team of parallel Claudes. 5 February 2026.

7. Chen F., Stratton J. Artificial Intelligence in the Firm: Bottlenecks in Software Production. Harvard University, working paper, revision of 28 October 2026 (earlier revision of 4 August 2026).

8. Demirer M., Musolff L., Yang L. Writing code versus shipping code: Productivity effects across generations of AI coding tools. VoxEU / CEPR column, 21 June 2026; NBER Working Paper 35275.

9. Stanford HAI. The 2025 AI Index Report. Section on falling inference costs, 2022-2024.