WHITE PAPER

WHAT A CTO MUST BUDGET FOR AI CODING TOOLS:

A Detailed Forecast of AI Coding Costs

Dr. Brian Scott Glassman

This is a research product of https://AInspire.ai

A CTO's guide to AI coding tool costs, including per-developer projections, model pricing trends, and sustainable budgeting strategies.

Executive Summary

CTOs, CEOs, and software development leaders need a clear view of the rising costs of AI coding tools such as Claude Code and OpenAI's Codex, and what to budget for the next quarter and next year. This analysis reviews costs for Claude and ChatGPT-5 and highlights a methodology that can be applied to your organization to project future AI coding tool expenses, because not everyone has a budget of a Fortune 500 company.

The analysis arrived at the $1,500 or $3,000 monthly per-developer budget range by calculating the hourly costs of running AI coding tools (GPT-5 vs Claude for single agents), then modeling realistic usage scenarios where developers would run AI coding assistants for total time of 3–5 hours daily with 1, 3, or 5 concurrent AI agents. The $1,500 budget tier assumes developers use AI coding tools with up to 3 concurrent agents for approximately 3 hours per day (reflecting current usage patterns), while the $3,000 tier accounts for more aggressive usages with up to 5 concurrent agents running for up to 5 hours daily. These projections factor in expected price decreases of 40% and 70% respectively over the next six months due to model optimization and competition, while also accounting for increased usage and productivity gains, ultimately predicting a stable cost equilibrium around $70 per hour of AI used.

The analysis further concludes that establishing a structured monthly AI coding budget for developers can promote more strategic and efficient use of compute resources before executing AI-driven coding prompts. The following table presents my recommended monthly AI coding budgets for teams of various sizes, based on two tiers: $1,500 per developer per month with access to Claude AI (Opus (5% usages) with Claude Sonnet 4.1 (95%), once released, utilizing up to three concurrent agents for a total coding time of three hours per day) and premium tier of $3,000 per developer per month with premium Claude AI (Opus with Claude Sonnet 4.1 utilizing up to five concurrent agents for a total of five hours per day coding). These are significant additional costs, but are to a large degree a must due to the massive efficiency gains from using a frontier AI.

Team Size	$1500 Monthly Budget	$3000 Monthly Budget
1	$1.5k	$3k
5	$7.5k	$15k
10	$15k	$30k
25	$37.5k	$75k
50	$75k	$150k
100	$150k	$300k

Table: Monthly Budgets for AI powered Coding Tools

The key takeaway from this analysis is as follows:

- Frontier LLMs are expected to continue producing more tokens per minute while requiring fewer computational resources. As a result, the cost per input and output token will decline until it reaches what I see as a baseline (approximately $1 per million input tokens and around $10 per million output tokens), after which pricing will primarily reflect perceived value and reputation, and base on some level of competition, rather than be based on operational expenses or R&D recovery.

- Frontier models will likely be offered in two tiers: one optimized for maximum accuracy and quality, and another designed for bulk coding tasks with solid but highly cost-efficient performance. The premium tier will be priced higher 2X the lower model, while the bulk coding model will be more affordable. This approach balances computational demands and cost, allowing the two models to complement each other effectively, expected to average the costs with 5% being on the premium model.

- The quality of coding outputs from these models is projected to improve steadily, with anticipated gains of 20% to 40% per quarter in improvements in first time accurate completions. Over the next several years, it will be challenging for open-source alternatives to replace leading frontier models such as those from OpenAI and Claude. The mentioned closed source frontier models are expected to generate significant revenues from customers, which will further fund accelerated training, enhancements, and release cycles. The business model of open-source alternatives may struggle to remain competitive in this environment (it was nice but I don't see it lasting).

- Token generation rates (tokens per minute) are expected to increase gradually (20 to 40%), unless advancements in hardware, such as LLM-optimized chips from companies like Groq (not xAI's Grok), or Inference only Cards from NVIDIA are applied to OpenAi and Anthropic then we can see 100% to 200% improvements in token rate generation.

- Using GPT-5 for five consecutive hours per day costs about $14, compared to $75 for Claude. With three agents over the same period, costs rise to roughly $42 for GPT-5 and $225 for Claude. Claude's pricing, however, is expected to decrease significantly with the release of Sonnet 4.1.

- Both conservative and aggressive adoption of Claude result in daily AI coding costs stabilizing around $70 per hour, with efficiency gains balanced by price deflation, ultimately delivering far greater productivity without significantly increasing expenses (see that section for clarification).

- Projected workforce costs scale rapidly: at $1,500 per developer per month, teams of 100 can reach $1.8M annually, while at $3,000 per developer per month, costs double to $3.6M per year, underscoring the need for strict budget controls and strategic deployment of AI coding tools so budgets are not blown out. Last thing we want is coders funding their own unauthorized side projects with AI coding hours stolen from their parent company.

- Note: Use this full analysis as a template within your own AI system to calculate your own detailed corporate AI coding cost projections. If you are dissatisfied with any of the assumptions, you can adjust them within your own model. Ask the AI to recreate the model by stating the core assumptions, then generate three scenarios and make the necessary adjustments accordingly specific to your corporation.

- Note: This analysis does not include a detailed review of productivity gains. While it may seem logical to perform a cost-benefit analysis, in practice the productivity impact varies greatly from task to task. Establishing a reliable baseline is nearly impossible without incorporating an organization's specific AI coding use cases. Therefore, each company must conduct its own testing to evaluate productivity improvements accurately.

Introduction

CTOs and Engineering VPs must begin factoring the substantial cost of AI coding programs into their departmental planning today. While adoption may seem early, it is critical to allocate budgets now for upcoming quarters and the next year to sustain momentum in AI-based software development and avoid falling behind.

AI coding providers, along with their cloud inference partners, are set to capture significant revenues as software developer productivity and effectiveness increase dramatically through the use of AI (tens of billions in compute bills). However, this leap in performance will come at a steep price. Technology leadership teams should prepare their CEOs in advance for these additional costs, as the bill for AI-driven productivity will be very high and likely remain that way.

Consequently, the central question for every CTO is: What will the cost per developer be? This article aims to provide that answer.
Let's jump into it.

A Quick Look Back at the Cost of OpenAI and Anthropic Coding Models

When ChatGPT and Claude first became effective at coding at scale, their costs were significantly higher per million tokens. This was because their large foundational AI models contained vast amounts of information, lacked optimization for output speed, and were missing many of the structural improvements that make today's LLMs faster. As a result, they were expensive to run on inference compute, leading to higher prices. The table below shows that pricing with ChatGPT-5 dropped dramatically, primarily because the model is less CPU intensive while still delivering high-quality results, a clear example of innovation driving efficiency.

Model / Release Period	Input Cost (per M tokens)	Output Cost (per M tokens)	Used for Coding
GPT-4o (May 2024)	~$5	~$15	Yes
o1-pro (Mar 2025)	$150	$600	Yes
GPT-5 (Aug 2025)	$1.25	$10	Yes

Table 1: Cost of OpenAI model use for coding as they were released

Next, we examine the historic pricing of Anthropic's Claude models, which has generally trended upward. However, the upcoming September release of Claude 4.1 Sonnet is expected to include a price reduction to better align with OpenAI's pricing. The most capable model for coding, Claude Opus 4.1, remains quite expensive. Consequently, Anthropic recommends using Opus for planning and complex coding tasks while leveraging Sonnet for large-scale coding due to its higher token output rates. I also expect to see a multi-model approach, where the more tedious, complex, or important the task, the more likely organizations will rely on the most expensive AI models, because in practice switching models can be done seamlessly.

Anthropic Claude Models

Model / Release (Date)	Input Cost (per M tokens)	Output Cost (per M tokens)	Used for Coding
Claude 3 Opus (Mar 2024)	$15	$75	Yes
Claude 4 Sonnet (May 22 2025)	$3	$15	Yes
Claude 4 Sonnet (extended context)	$6	$22.50	Yes (≥200k tokens)
Claude 4 Opus (May 22 2025)	$15	$75	Yes
Claude Opus 4.1 (Aug 5 2025)	$15	$75	Yes

Table 2: Cost of Anthropic's model use for coding as they were released

Pricing Trends of AI Coding Models

Below is the pricing for each frontier model at the time of its release.

Month	Model	Provider	Input Cost	Output Cost	Key Insight
2024-03	Claude 3 Opus	Anthropic	$15	$75	Baseline pricing for high-end AI models
2024-05	GPT-4o	OpenAI	$5	$15	Dramatic 66.7% input, 80% output reduction
2025-03	o1-pro	OpenAI	$150	$600	Specialized reasoning model, major outlier
2025-05	Claude 4 Sonnet	Anthropic	$3	$15	Return to low costs: 98% input, 97.5% output drop
2025-08	GPT-5	OpenAI	$1.25	$10	Continued deflation: 58.3% input, 33.3% output drop

Table 3: Pricing trends of major AI coding models at release, showing shifts across OpenAI and Anthropic

We are beginning to see competition between OpenAI and Anthropic drive down per-million-token pricing, alongside the evolution of smaller, more efficient models. OpenAI took a different approach with GPT-5 by introducing reasoning effort = {low, medium, high}, which directs the model to spend more compute cycles before producing an answer, generally improving quality. Nevertheless, this is a less optimal solution compared to deploying two distinct models, one of which is smaller and requires a lighter compute footprint.

Opinion: I expect the input token cost to stabilize at around $1, while the output cost will likely remain in the $10 to $20 range for frontier coding models over the next year. The detailed reasoning behind this expectation is beyond the scope of this article, but I would be glad to discuss it further.

Coding Performance and Benchmark Comparisons

Next, we need to examine coding performance. From the table, it is clear that each model shows excellent improvement on its SWE-bench coding tasks, advancing at a steady pace. Soon SWE-Benchmark test will need a new version. Improved output quality naturally supports higher pricing, particularly when sensitive tasks demand high-quality coding and DevOps AI agents, plus human developers get used to a certain quality of code and do not like to go backwards, as it feels highly counterproductive.

Model	Release Date	SWE-bench Verified	Notes
GPT-5	Aug 2025	74.9%	Top scoring, narrowly ahead of Claude Opus 4.1
Claude Opus 4.1	Aug 2025	74.5%	Incremental but meaningful gain over Opus 4
Claude Sonnet 4	May 2025	~64.9%	Solid mid-tier Anthropic model
Claude Opus 4	May 2025	72.5%	Strong performer on large-scale coding problems
GPT-4o	May 2024	~21.6%	Weakest performer in benchmarks

Table 4: Coding performance benchmark results across models

However, output quality is only part of the equation. We must also consider the speed at which these outputs are generated.

Coding Speed / Token Generation Rate

As developers, we value rapid code generation, but more importantly, we expect the process to continue reliably until all planned tasks are completed. The ability to step away briefly, such as taking a break while tasks are executed seamlessly, fosters a sense of productivity and efficiency. Smaller models like Claude Sonnet often achieve faster token generation, while GPT-5 also delivers impressive speeds through its architecture. However, at higher reasoning levels, GPT-5 performance slows noticeably. In my assessment, attempting to make a large language model function as a universal multi-tool, as seen with GPT-5, is not ideal because the components inevitably influence one another, creating complex internal dependencies.

The key lesson is that AI agent models (like Claude Sonnet) should be optimized for speed since they perform the majority of the work, while the orchestrator LLM (Opus so on) can afford to operate more slowly, as its accuracy is crucial to guiding the overall process. Expect the majority of the token generation speed to be around 30 to around 50 token/sec in practice for the next 6 months. Unless a significant breakthrough emerges to challenge the frontier models, such as Mamba or Diffusion coding models.

Model	Max Token Speed Range (tokens/sec)	Notes on Latency / Behavior	Source
GPT-5 (medium)	~181.5	Very fast throughput, optimized	Artificial Analysis.ai, Reddit
Claude Opus 4.1	45 – 65	Quick first-token latency (3–5 sec)	Medium.com
Claude Sonnet 4	3.4 – 94.6	Highly variable performance	Artificial Analysis.ai, Reddit

Table 5: Token Generation Speed Comparison

The table below provides a more accurate representation of coding speed, as token outputs per minute of coding tend to be more stable and are relatively close to one another. These figures are based on user-reported data rather than questionable company metrics.

Model	Average TPM	Tokens per Hour	Sources
GPT-5	~2,360	141,600	Leanware
Claude Opus 4.1	~2,210	132,600	Artificial Analysis, Medium
Claude Sonnet 4	~3,710	222,600	Reddit, Replicate, Composio

Table 6: User Reported Token Generation Speed Comparison (Per Minute and Per Hour for GPT-5, Claude Opus, and Sonnet)

Cost Per Hour of Coding at Full Speed

Currently, no AI system codes continuously for a full hour, though this may change in the near future (give it 4 months). Today, software developers typically run a coding plan against a large product specification, then evaluate the results, suggest changes and updates, or execute the code to analyze outcomes. In practice, my team operates at full coding speed for roughly 30 to 40 percent of each hour. Over an eight-hour workday, this equates to approximately three hours of sustained AI-driven coding. For a baseline calculation, however, we need to determine the per-hour costs. A referenced source cited an 8-to-1 input-to-output ratio based on context window size and related factors. Using this, I multiplied the output tokens by eight to estimate the per-hour cost at full read-write capacity for the three models.

Although GPT-5 offers the lowest cost per hour, it is relatively new, and developers still require time to evaluate its effectiveness before committing to such a transition. While a CTO may be tempted to mandate adoption purely for cost savings, the strategic wisdom of such a decision warrants careful consideration like a unbiased blind code analysis.

Model	Input Cost/ M	Tokens/Hour Input	Output Cost/ M	Tokens/Hour Output	Total Cost/Hour
GPT-5	$1.25	1,132,800	$10.00	141,600	$2.83
Claude Opus 4.1 (Aug 5 2025)	$15.00	1,060,800	$75.00	132,600	$25.86
Claude 4 Sonnet (extended)	$6.00	1,780,800	$22.50	222,600	$15.69

Table 6: Cost Per Hour of Coding at Full Speed

Agent Cost Comparison

Claude Code with agents demonstrated that the future of software development WILL rely heavily on multi-agent coding. In this model, agents work independently on their respective pieces of code and then close once their portion of the plan is complete. For the purposes of this analysis, I have assumed scenarios with 1, 3, and 5 agents operating concurrently. These are the respective costs. While GPT-5 is generally more cost-efficient, the trade-off between cost and quality must always be considered. It is also important to note that operating with 5 agents drives costs into a range comparable to the hourly wage of many software professionals.

Agents	GPT-5	Claude Sonnet 4
1	$2.83	$15.69
3	$8.49	$47.07
5	**$14.15**	**$78.45**

Table 7: Cost Per Hour of Programming Assuming You Add More Agents

Senior Software Developer Daily AI Costs

Again you can not expect and coder to run the AI all day, so I divided it up into running for a total of 3 hours, 5 hours, and 8 hours. to show expenses for both GPT-5 and Claude Sonnet. I expect the Claude Sonnet 4.1 to be release to drop the pricing by 40% so the 5 and 8 hour pricing for 5 agent will be around ~$150 to ~$350 per day.

Agents	3 Hours	5 Hours	8 Hours
1	$8.49	$14.15	$22.64
3	$25.47	$42.45	$67.92
5	$42.45	$70.75	$113.20

Table 8a: Daily Cost of GPT-5 for 3, 5, and 8 Hours Across 1, 3, and 5 Agents

Agents	3 Hours	5 Hours	8 Hours
1	$47.07	$78.45	$125.52
3	$141.21	$235.35	$376.56
5	$235.35	$392.25	$627.60

Table 8b: Daily Cost of Claude Sonnet 4 for 3, 5, and 8 Hours Across 1, 3, and 5 Agents

Scenario Analysis: Conservative vs Aggressive AI Adoption

I have considered several scenarios that may unfold over the next six months. As a baseline, we assume AI is used for 3 hours per day with 1 agent, which is close to what many programs are experiencing today in their token spend.

In a conservative six-month projection, AI usage remains at 3 hours per day but expands to 3 agents working concurrently. In this case, token speed is expected to increase by 25%, while prices decrease by 50%, resulting in daily costs of around $70 per hour, but with significantly greater output (over 300% increase in code writing compared to 1 agent for 3 hours).

For the aggressive forecast scenario, AI coding is projected to run for 5 hours per day with 5 agents (~730% increase compared to 3 agents for 3 hours).

Although token generation speeds would also increase, I anticipate costs to decline more rapidly, reducing to about 70% of their current levels. Despite higher performance and productivity, the net result is that the cost per hour of AI coding remains roughly $70, but with significantly greater output (over 730% increase in code writing over the conservative case).

Scenario	Hours/Day	Agents	Speed Changes	Price Cuts	Daily Cost	% Change
Base Case	3	1	Baseline	Baseline	$50	-
Conservative	3	3	+25% output	-50% both	$70	+62.3%
Aggressive	5	5	+50% output	-70% both	$70	+62.7%

Table 9: Scenario Analysis of Daily AI Usage Costs

Scenario	Usage Multiplier	Price Multiplier	Net Effect
Conservative	+300% (3x agents, 25% speed)	-50% (halved prices)	+62.3%
Aggressive	+733% (5x agents, 67% hours, speed gains)	-70% (major price cuts)	+62.7%

Table 10: Formula Impact Analysis of Usage and Pricing

Key Insight: Massive usage increases (+300% to +733%) are almost entirely offset by projected AI price deflation (-50% to -70%), resulting in similar cost outcomes.

Monthly Costs Analysis (20 Working Days)

The most important part of this analysis is expected cost per month of deploying these agent will be high, look at current sonnet 4 vs. expected 4.1 (I took half the cost) still at $1,400 per month per programmer only running 3 agents is significant.

Across a development team of 50 programmers, the monthly bill could reach approximately $70,000 (or $840,000 annually) when using Claude Sonnet, while GPT-5 would still amount to about $25,000 per month (or $300,000 annually). Although these costs are substantial, the corresponding productivity gains are expected to be significant. Organizations should prepare their budgets accordingly.

Model	Agents	Daily Cost	Monthly Cost
GPT-5	1	$9	$170
Claude Sonnet 4	1	$48	$940
GPT-5	3	$25	$500
Claude Sonnet 4	3	$140	$2,800
Claude Sonnet 4.1 (expected)	3	$70	$1,400

Table 11: Monthly cost for AI running 3 Hours/Day with 1 to 3 agents

Now say we bump it up to 5 hours per day with 1 to 5 agents, Across a development team of 50 programmers, the monthly bill could reach approximately $190k (or $2.34M annually) when using Claude Sonnet 4.1, while GPT-5 would still amount to about $70k per month (or $840k annually). Yep this is still very expensive.

Model	Agents	Daily Cost	Monthly Cost
GPT-5	1	$14	$280
Claude Sonnet 4	1	$78	$1,570
GPT-5	5	$70	$1,400
Claude Sonnet 4	5	$390	$7,850
Claude Sonnet 4.1 (expected)	5	$195	$3,900

Table 12: Monthly cost for AI running 5 Hours/Day with 1 to 5 agents

Projected Cost for the Workforce

You can manage AI deployment by limiting the number of AI coding hours allocated per developer. While this approach may reduce maximum productivity, it encourages more strategic use of compute resources and better task planning per developer. A budget can be set on a per-developer basis, with the following examples:

Running AI coding for 3 hours a day with 3 agents would likely result in costs ranging between $1,000 and $1,700 per employee per month.
For highly productive developers running AI for 5 hours a day with 3 agents, the monthly budget would be approximately $3,000.

What organizations should avoid, among other issues, is allowing developers to press ‘go’ on a poorly conceived AI coding plan that consumes 20 to 30 minutes of expensive compute only to generate code that must be rolled back. Basically call it AI coding waste, and doing this repeatedly over teams of developers is a great way to waste money and time. One simple way I can see to stop this is to give them a token budget and let them get smarter about using them to maximize their work.

Team Size	$1500 Monthly Budget	Yearly Spend ($1500)
1	$1.5k	$18k
5	$7.5k	$90k
10	$15k	$180k
25	$37.5k	$450k
50	$75k	$900k
100	$150k	$1.8M

Table 13a: Projected Team AI Budget ($1500 Monthly Budget)

Team Size	$3000 Monthly Budget	Yearly Spend ($3000)
1	$3k	$36k
5	$15k	$180k
10	$30k	$360k
25	$75k	$900k
50	$150k	$1.8M
100	$300k	$3.6M

Table 13b: Projected Team AI Budget ($3000 Monthly Budget)

Conclusion

The financial impact of deploying AI coding tools is substantial and will require careful planning by CTOs and technology leaders. While prices may gradually stabilize, they are still likely to remain expensive on a per-user basis. A smart strategy is to enforce budget caps per developer, ensuring that organizations extend their AI budgets effectively while still capturing the productivity benefits of AI-driven software development.

Note: I have deliberately chosen not to include Google Gemini 2.5 or xAI Grok in this analysis. Google's track record as a steward of sensitive data raises concerns about entrusting them with proprietary corporate code, and for that reason I do not consider them a viable option. In addition, Grok remains a very new entrant to the market, and its relative immaturity brings the risk of significant security vulnerabilities. As a result, both platforms are excluded from the projections in this report.