AI Development Cost Guide

How to budget for AI tools, development work and the cost of running an application.



The work and the tools are different costs

A time-boxed pilot pays for agreed development and a progress review. It is not a promise of a finished application. Agree on the scope and fee separately from AI and hosting costs.

Set an AI budget before work starts

Choose the tools and agree on a spending limit based on expected use. Compare subscriptions with metered API billing, and check what each plan includes. A spending limit is not a quote or a guarantee that the project will be finished within that budget.

The company should control billing and provide named developer access. Approve any increase before spending more. Estimate ongoing costs separately before launch.

Four things to agree before kickoff

1. What to build or test first, and how progress will be reviewed.
2. When work starts, what the pilot costs and when payment is due.
3. Which AI tools and other services are paid for separately.
4. Who owns the account, sets the spending limit and approves increases.

Before kickoff, put the scope, fee and tool budget in writing. Track actual usage during the pilot, then use it to estimate the next phase. Do not infer past AI spending from an estimate without receipts or usage records.

The following pages show the options and supporting numbers. A plain-language glossary is on the last page. All arrangements here are for discussion, not new terms already accepted.

AI is not free.

Professional judgment and paid computing capacity are different costs.

02 / LABOR

What the fee covers

Development on one agreed application

01	**Agree priorities**	Choose what to build and test first.
02	**Develop for two weeks**	Build, troubleshoot and review.
03	**Review progress**	Show what works and what remains.

Human expertise: scope, design, testing and delivery.

02 / TOOLS

What is paid separately

Agree who owns the account and budget

01	**Development AI**	A subscription or measured API usage.
02	**Production operation**	Hosting, storage and customer-facing usage.
03	**Historical spend**	No project-attributable bill verified.

AI capacity: subscriptions or measured provider usage.

AI can accelerate development, but it does not remove the work of designing, reviewing and delivering reliable software. Provider usage is a real expense that needs an agreed owner and budget.

Three separate budget lines

Development services. The proposed $2,500 covers two weeks of development on one agreed app and a progress review. AI tools and other approved provider costs are separate.

Development AI. Coding, troubleshooting, reviews and design iterations consume subscription allowance or metered API usage.

Production operation. Customer-facing image generation, hosting, storage and databases need their own operating budget. A developer subscription does not automatically cover them.

What has already been invested

A local trial has been built and tested. That demonstrates work completed, **not a verified AI bill**. No project-attributable provider invoice or complete usage export was verified for this review. Historical AI dollars spent remain unverified, not zero.

Code volume cannot reconstruct repeated prompts, context reads, caching, reasoning or failed attempts. Retrospective reimbursement needs dated receipts and an agreed allocation method, not an invented token estimate.

Proposed arrangement: ProSwag funds and controls the selected account, with named developer access and a written ceiling. This is a proposal for agreement, not a term already accepted by either party.

Choose the funding route

Select the capacity and account controls that fit the agreed work.

03 / SUBSCRIPTION

A plan with an allowance

A monthly fee does not mean unlimited use

01	**Choose the plan**	Compare included usage and who controls billing.
02	**Watch the allowance**	Check any limits and when they reset.
03	**Check extra charges**	Do not count included credits twice.

Subscription: a predictable base price and allowance.

03 / METERED API

Pay for measured use

Costs vary by model and consumption

01	**Track inputs and outputs**	Record usage against the project.
02	**Check credit rules**	Confirm eligibility and expiration.
03	**Approve a ceiling**	Agree on any overages separately.

API: pay for measured usage, with project-level tracking.

Published subscription options

ChatGPT Pro 100 / 200 / 500: $100 / $200 / $500 monthly. Pro 500 has the highest included usage and Ultrafast access. Lower tiers exist; $500 is not a provider-wide minimum. [1]

Claude Max / Ultra: $100 / $200 monthly on the web. Individual plans are not substitutes for company team administration. [3,4]

Company-managed Claude Team example: one Premium developer seat ($125) + one Standard company seat ($25) = **$150/month** on monthly billing, meeting the two-member minimum. Capacity differs from individual Ultra and Pro 500. [5]

What happens when usage runs out?

OpenAI currently states Pro 100, 200 and 500 have **no five-hour Work/Codex limit**; included allowance still applies. Claude Pro and Max have five-hour and weekly limits; verified Ultra has no five-hour limit. Hitting a session limit does not automatically mean waiting a whole week. Check the account reset time. [2,4]

API billing is a separate route, charged by model and consumption. Some plans include API credits with eligibility and expiry rules. Claude Max/Ultra monthly API credits cannot cover Claude Code or app overages. Do not count the same allowance twice. [6]

Budget for discussion: reserve up to $500 for the first month of development AI, funded by ProSwag. Choose one route and separately approve overages. This is a proposed ceiling, not a verified minimum or a completion guarantee.

Make the assumptions visible

Illustrative two-week API scenarios, not historical project consumption.

04 / VOLUME

Usage changes the bill

Illustrative input/output token scenarios

01 **10M input + 1M output**
$30 for Sol / Sonnet 5.5 in this example.

02 **50M input + 5M output**
$150 under the same rates.

03 **150M input + 15M output**
$450 under the same rates.

Iterations, reviews and retries add to total usage.

04 / SPENDING CONTROL

A ceiling is not a forecast

Set controls before spending starts

01 **At 80%**
Proposed alert to review usage.

02 **At the agreed ceiling**
Stop and review before incurring more.

03 **Provider cutoff**
Check whether the provider enforces it.

Different volumes produce different bills.

Input is what the AI reads; output is what it produces. Tokens are small pieces of that text. The table uses standard uncached input and billed output, with OpenAI short-context rates; no reuse discount is assumed. Volumes are illustrative assumptions, not predictions derived from this repository. [7,8]

Scenario	GPT-6.1 Sol / Sonnet 5.5	Claude Opus 5.5	GPT-6 Astra
10M input + 1M output	$30	$60	$150
50M input + 5M output	$150	$300	$750
150M input + 15M output	$450	$900	$2,250

Rate per million tokens (input / output): Sol and Sonnet 5.5 $2 / $10; Opus 5.5 $4 / $20; Astra $10 / $50. Formula: input millions x input rate + output millions x output rate.

An 80% Sol / 20% Astra token mix costs $54 / $270 / $810 across those rows. Adding a 25% planning reserve gives approximately $68 / $338 / $1,013. Faster modes, long contexts, tools, images, taxes and hosting are excluded.

A ceiling is a control, not a forecast

Propose an alert at 80% and a developer stop-and-review at the agreed ceiling. Verify whether provider controls actually enforce a hard cutoff. Keep paid overages and auto-reload off unless separately approved.

Never add hypothetical API-equivalent costs to a subscription invoice unless separate API or overage charges were actually incurred. When total workflow usage already includes retries, do not multiply for retries again.

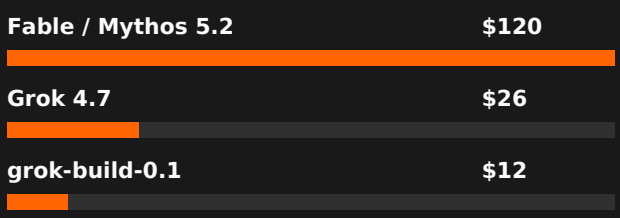
More models. Same assumptions.

Claude Fable / Mythos 5.2 and Grok: direct API comparisons.

05 / SAME TOKEN VOLUME

One example, three prices

10M input + 1M output tokens



Illustrative direct API use; Grok short context.

A model price measures usage, not a finished application.

05 / READING THE NUMBERS

Compare like with like

Separate token math from project cost

01 **Hold volumes constant**
Use the same input and output assumptions.

02 **Check the rate tier**
Grok's 200K threshold is per request.

03 **Review actual results**
Include retries, quality and review time.

Use the same assumed volumes to compare the arithmetic.

Claude Fable 5.2: $8 per million input tokens and $40 per million output tokens. **Mythos 5.2** has the same specifications and prices, but access requires a verified organization. Do not assume ProSwag has access. [12]

Grok 4.7: $2 input / $6 output per million tokens below 200,000 prompt tokens. **grok-build-0.1:** $1 / $2 at short context. These are direct API prices, **not Grok Bot charges**. [13]

Scenario	Fable / Mythos 5.2	Grok 4.7	grok-build-0.1
10M input + 1M output	$120	$26	$12
50M input + 5M output	$600	$130	$60
150M input + 15M output	$1,800	$390	$180

Formula: input millions x input rate + output millions x output rate. Standard uncached global API usage; no batch discounts. Volumes include billed output, not just the final visible answer. Tools, images, taxes, hosting and faster modes are excluded.

At 200,000 or more prompt tokens per request, Grok 4.7 uses $4 / $12 and grok-build-0.1 uses $2 / $4 for all tokens in that request. If every request used those rates, the Grok totals above would double. Fable fast mode also doubles its base rates. [8,13]

No blanket winner: lower token prices do not prove lower cost per accepted result. Compare a representative task, quality, retries, total usage and review time before choosing. More parallel agents can consume more, not less.

[12] Official Fable 5.2 details | [13] Official Grok API pricing

Grok Bot, in plain English

An AI teammate product is not the same thing as a single AI model.

06 / ASSIGN THE BOT

Define the task first

A Bot is a workflow product, not a model

01	**Set a clear role**	Give it a task and a stopping point.
02	**Limit tools and files**	Approve what the Bot can use.
03	**Plan for shared access**	Separate Bots are not security boundaries.

Give each Bot a bounded role and clear approval rules.

06 / KEEP HUMAN REVIEW

Decide what is ready

Automatic model routing is not a picker

01	**Inspect the output**	Check what the Bot did before using it.
02	**Check the allowance**	Bot usage is not a model-by-model API bill.
03	**Approve the next step**	Don’t allow automatic top-ups by default.

People still check the work and decide what is ready.

What it is: Grok Bot can use a cloud computer, tools, websites and files to carry out work and continue in the background. It is a distinct product from the Grok chat app. [14]

Can several Bots work together? Yes. Official documentation supports Bots working and coordinating in parallel. They share the account's cloud computer, files and sessions; each has its own screen. Separate Bots are not separate security boundaries. [14]

Can it use different AI models? Yes, through automatic routing. Grok Bot may use first-party or third-party models. There is no model picker, no guaranteed model per task and no underlying model shown on each usage record. It is not a user-controlled side-by-side model comparison. [14]

What the current plans say

The X Subscription includes Grok Bot on **Premium $30, Super $100 and Ultra $200 per month**, before tax. Free and Plus do not include it. The allowance is shared weekly across Grok, Bot and Cursor; higher tiers have larger pools. Legacy accounts can have different billing. [14,15]

Bot activity is billed as Grok Bot usage, not a visible model-by-model API bill. Separately selecting third-party models in Cursor is billed separately at API list prices. Do not treat those two workflows as the same charge. [14,15]

Before choosing it: check the actual account plan, privacy rules and remaining allowance. Agree a small trial and spending cap; keep automatic top-ups off unless approved. Parallel Bots do not make usage free or guarantee a better result.

[14] Official Grok Bot FAQ | [15] Official subscription pricing
Sources checked November 12, 2026. No Ross or ProSwag account plan or bill was inspected.

Use evidence, then calibrate

Published tests offer context. A small measured trial improves the estimate.

07 / MEASURE THE TRIAL

Track one approved task

Suggested calibration, not a forecast

01	**Record all billed use**	Include retries and unsuccessful attempts.
02	**Separate billing categories**	Track model, project and charges.
03	**Compare with actuals**	Revise the estimate before spending more.

Measure an approved task before forecasting the remainder.

07 / REVIEW THE RESULT

Measure what passed

Raw outputs are not the same as accepted work

01	**Test against agreed checks**	Was the task completed successfully?
02	**Record unfinished work**	Note fixes and review still needed.
03	**Use cost per useful result**	Token volume alone cannot show readiness.

Passing review and acceptance matter more than raw output.

A useful real-world reference

Anthropic reports enterprise Claude Code deployments averaging about **$13 per developer per active day** and **$150-$250 per month**, with 90% of users below $30 per active day. These describe observed deployments, not a ProSwag quote or a guaranteed cap. Do not multiply daily and monthly figures as if they describe the same active-day count. [9]

Why benchmark costs are not app prices

Aider reports historical GPT-5 low / medium / high effort runs costing **$10.37 / $17.69 / $29.08 for the entire 225-exercise suite**. Accuracy also varied. These isolated exercises demonstrate effort/cost tradeoffs, not the price of a finished web application. [10]

Long-horizon GDPx software tests report costs per attempt, not per successfully delivered feature. Their task mix, agent harness, time limits and low completion scores make a direct conversion into an app quote misleading. [11]

Suggested measurement plan

1. Agree on trial priorities and what to review at the end of two weeks.
2. Meter the first two approved development days by project, model and billing category, including unsuccessful attempts.
3. Compare estimates with actuals and unfinished work.
4. Reforecast before increasing the spending limit.

Keep delivery separate from token counts. Use lower-cost models for routine tasks and premium models where they add value. Testing, integration, waiting and human review still determine readiness and schedule.

Company ownership. Clear terms.

Agree the account, budget and acceptance rules before the next phase.

08 / OWNERSHIP

ProSwag owns the account

Confirm access and billing before kickoff

01	**Company billing**	ProSwag pays for the agreed tools.
02	**Named developer access**	No shared passwords.
03	**Written spending ceiling**	Approve more before it is incurred.

Named access and company billing, not shared passwords.

08 / SEPARATE BUDGET

Plan production separately

Estimate running costs before launch

01	**Test a representative image**	Record quality, resolution and retries.
02	**Measure cost per attempt**	Multiply by the expected attempts.
03	**Add hosting and storage**	No monthly total is verified here.

Production image usage and hosting need their own budget.

Confirm together: the $2,500 upfront trial fee, one app and its priorities, the agreed start date, the day-14 progress review, ProSwag-owned AI access, and the separate tool budget. Any employment discussion is separate from the trial.

Production estimate: test the chosen image model, quality, resolution and revision volume; then estimate image attempts x measured cost per attempt, plus hosting/storage. No production per-image price or monthly runtime total is verified here.

Sources checked November 12, 2026

[1] OpenAI Pro tiers

[2] OpenAI usage limits

[3] Claude Max

[4] Claude Ultra

[5] Claude Team

[6] Claude Code billing

[7] OpenAI API rates

[8] Claude API rates

[9] Claude Code costs

[10] Aider leaderboard

[11] GDPx methods

Source labels above are clickable. Prices and limits can change; confirm checkout, eligibility, regional taxes and terms. This is a planning draft, not an invoice, purchase authorization or delivery guarantee. Existing terms remain in force until amended by agreement.

The cover image and diagrams are illustrative, not billing evidence, client endorsement or proof of delivered software.

A plain-language glossary

A quick reference for the terms in the detailed pages.

09 / USAGE TERMS

Follow the work

Small terms that explain an AI bill

01	**Input**	What the AI reads.
02	**Output**	What it produces; billed usage can differ.
03	**Caching**	Eligible reuse may reduce some input charges.

AI is a paid tool used alongside human judgment.

09 / COST TERMS

Follow the money

Plan, metered usage and ceiling are distinct

01	**Subscription**	Monthly fee with an included allowance.
02	**API usage**	Charges based on measured consumption.
03	**Spending ceiling**	An agreed limit, not a cost forecast.

A spending limit is a decision, not a prediction.

AI model. The particular AI system doing the work. Different models have different prices and capabilities.

Subscription. A monthly plan with an included allowance. It is not necessarily unlimited, and extra charges may still apply.

API / metered usage. A way for software to use AI and pay for measured consumption, rather than only a monthly plan.

Tokens; input and output. Small pieces of text used to measure usage. Input is what the AI reads; output is what it produces. Code length does not reveal the total bill.

Context and caching. Context is the information supplied for a task. Caching can give a discount when eligible information is reused; it is not assumed in our table.

Credits and overages. Credits are an allowance with rules and possibly an expiry date. Overages are extra use beyond an allowance. Check what each credit can pay for.

Benchmark and pilot. A benchmark is a published comparison test, not a quote for this app. A pilot is a small, measured part of our own project.

Development versus production. Development is building and checking the app. Production is running it for actual users after release.

Ceiling and reserve. A ceiling is the agreed spending limit. A reserve is extra room in an estimate for uncertainty, not automatic permission to spend.

Acceptance checklist. The agreed list of things the application must do and checks it must pass before it is considered ready.

Bottom line: the service fee pays for the professional work. AI tools and running costs need clear ownership and agreed funding. Numbers in an example are not a bill.