The Pricing Playbook

How founders set a price they can defend

JANE FOUNDER

The Pricing Playbook

How founders set a price they can defend

Jane Founder

Contents

Introduction

Most founders guess at pricing. This short book shows a repeatable method that ten customer interviews can validate in a week.

The guess is understandable. You set the price on launch day, when you had no customers and no data, and you needed a number on the page so that people could buy. So you looked at the nearest competitor and went a little lower, or you picked the three tiers that every pricing page seems to have, or you chose a figure that felt fair to you. It worked well enough. People paid. You moved on to the next fire.

Then something changed. An investor asked why the price was what it was, and you heard yourself improvising. A customer asked for a discount and you gave it, because you had no reason not to. You noticed that your best customer, the one who gets the most out of the product by a wide margin, pays roughly the same as your most demanding one. Or a sales hire asked for a cheat sheet on pricing and you realised there was nothing to write down.

The problem is not the number. Almost any number will find a few buyers. The problem is that there is no reason attached to it, and a price with no reason cannot be defended, cannot be raised, and cannot be explained to the next person who asks.

This book replaces the guess with a method. The method has five steps and a habit:

1. Interview ten customers in one week, using a fixed script.
2. Choose a value metric: the unit you charge for.
3. Draft three tiers around that metric.
4. Set the numbers, each with a reason you can say in two sentences.
5. Say the price out loud, to prospects and to the customers already paying.

6. Review the whole thing every quarter, for ninety minutes, from one page.

None of it requires a data team, a consultant, or a survey tool. It requires ten calendar slots, a scoring sheet, and the willingness to hear what customers actually compare you to.

I have deliberately kept theory out. You will not find demand curves, named studies, or figures about what percentage of companies do what. In my experience those figures get quoted in board meetings and change nothing, because they are about other companies. Instead the book reasons from mechanics, and it works through one fictional company from start to finish: Cronbell, a small monitoring tool for scheduled jobs and data pipelines, whose founder priced it by copying a competitor and now has to fix that. A second fictional company, Doorframe, appears where a contrast helps. The names are invented; the situations are the ones I see over and over.

Who this is for

- A founder who set a price on launch day and has not touched it since.
- A founder who copied a competitor's pricing page and now wonders why the customers who arrive do not match the ones they wanted.
- A product manager who inherited a pricing page that nobody in the company can explain.
- Anyone who has to present pricing to investors, a board, or a first sales hire and wants a reason behind every number.

How to use it

Read it straight through once; it takes about an hour. Then go back and do the work chapter by chapter. Each chapter ends with a concrete task, and the tasks chain: the value hypothesis feeds the interviews, the interview scores drive the metric choice, the metric shapes the tiers, the tiers get numbers, and the numbers get said out loud. Two weeks from now you should have a pricing page you can defend and a review date in your calendar.

Why Guessing Fails

Priya built Cronbell because she was tired of finding out on Monday that a Friday-night job had not run. Cronbell watches scheduled jobs and data pipelines. Each job sends a heartbeat when it starts and finishes; if the heartbeat is late, missing, or reports a failure, Cronbell alerts whoever should know. It is a small product with a clear purpose, and engineering teams like it.

She launched at nineteen dollars per user per month. The reasoning, such as it was: the closest competitor charged twenty dollars per user, and she wanted to be a little cheaper without looking cheap. Eighteen months later Cronbell has forty paying teams, averaging a little over two seats each, for around seventeen hundred and fifty dollars of monthly recurring revenue.

Look at the two ends of her customer list. Her largest customer is a data team of four people who monitor about six hundred pipelines. They pay seventy-six dollars a month and they are delighted. Her smallest is a solo developer with three cron jobs who pays nineteen dollars, writes in every few months to say it is too expensive, and is still somehow a customer. When an angel investor asked Priya why nineteen dollars, she said "it's competitive", and both of them knew that was not an answer.

Four ways to guess

Nearly every founder I have worked with arrived at their first price by one of four routes. Each is reasonable on launch day. None of them survives contact with a real question.

Copying a competitor. You inherit their number without their reasons. You do not know which customer they priced for, what metric they believe tracks value, or whether their price is a mistake they have not fixed yet. Priya copied a per-seat price from a company that sells mostly to large engineering organisations, where seats are a fair proxy for scale. Cronbell's customers are small teams watching a lot of jobs. The number came across; the logic did not.

Cost-plus. You add up hosting, support and a margin, and price a little above that. For software the cost to serve one more customer is usually small and has nothing to do with what the customer gets. A price derived from your cloud bill tells the customer nothing about why they should pay it.

The number that feels fair. This is your own price sensitivity, not the buyer's. Founders are poor proxies for their customers: you know what the product cost to build, you tend to be frugal, and you are afraid of hearing no. All three push the number down.

The template. Three columns at something like nine, twenty-nine and ninety-nine dollars, because that is what a pricing page looks like. The columns exist, but nothing about them reflects how the value actually grows for the customer.

Most pricing pages I see are a blend of all four: a competitor's metric, a template's tier count, a founder's comfort level for the middle number, and a quick check that it clears the hosting bill. Each ingredient seemed harmless. Together they produce a page that nobody in the company can explain, which is exactly the page Priya has.

What a guess costs you

You do not need a study to see the damage; the mechanics are enough.

A revenue ceiling. When the price does not track value, your biggest customers are subsidised by your smallest. Cronbell's six-hundred-pipeline team pays four times what the solo developer pays and gets two hundred times the coverage. Every future customer like them arrives underpriced, and the better the product gets at serving large teams, the wider the gap becomes.

The wrong customers. A low per-seat price attracts people who evaluate on price, and people who evaluate on price churn on price. Priya's loudest complaints come from the customers who pay least.

No defence. The moment anyone pushes, you either cave or bluff. A discount request with no reason to refuse becomes a discount. An investor question becomes an improvisation. A sales hire gets no cheat sheet because there is nothing to put on it. And every discount you give without a reason quietly sets the new price.

A frozen number. If there was no reason for the old number, there is no reason for a new one either. So the price stays where it is, year after year, while the product improves around it.

> A price is a claim about value. A guess is a claim with no evidence.

What "defensible" means

A defensible price is not a correct price. There is no correct price; there is only a price you can explain and revise. Defensible means three things:

1. There is a reason in the customer's terms: what they get, and why it is worth this.
2. The unit you charge for grows when the value grows.
3. You can say the number out loud, to a customer or a board, follow it with a reason in two sentences, and that reason rests on what customers actually told you.

Everything in this book serves those three conditions.

The method on one page

Step	What you produce	Time
Write the value hypothesis	One page: outcome, drivers, alternatives	An afternoon
Interview ten customers	A scoring sheet with ten rows	One week
Choose the value metric	One sentence	An hour
Draft three tiers	A table of limits and fences, no prices	An afternoon
Set the numbers	Three prices, a defence sentence for each, a migration model	A day

Step	What you produce	Time
Say it out loud	Five test conversations, a pricing page, an email to existing customers	A week
Review every quarter	A one-page review and a pricing log	Ninety minutes

Before you go further, do one thing. Write down your current price and, in a single sentence, the reason for it. If the sentence contains the words competitor, felt, seemed or standard, you have a guess. Keep the sentence. You will rewrite it at the end of the chapter on setting the numbers, and the difference between the two versions is the point of this book.

What You Are Actually Selling

Interviews without a hypothesis are a survey. You will get ten pleasant conversations and a pile of notes with nothing to test against. So before you book a single call, write down what you believe you are selling. You will probably be partly wrong, which is what the interviews are for.

Features are what you built; outcomes are what they buy

Cronbell's feature list is easy to write: cron monitoring, pipeline heartbeats, alert routing to email, Slack and a pager, a dashboard, and thirty days of history. None of those is what a customer pays for. They pay to find out that a job failed before anyone downstream does, and they pay to stop worrying about it.

Priya's first attempt at a value statement was "Cronbell monitors your scheduled jobs and pipelines." That is a description. Her second attempt, after an uncomfortable half hour, was: "Cronbell tells the right person within minutes when a job that matters did not run, so the failure is fixed before anyone downstream notices." That is an outcome, and every phrase in it can be tested in an interview. Which jobs matter? Who is the right person? What does "downstream" cost?

A useful way to sharpen the sentence is to describe the customer's Monday morning before and after. Before Cronbell: the analytics lead opens the dashboard, sees yesterday's numbers missing, messages an engineer, and spends an hour discovering that Friday's export never ran. After: a Slack message at six on Saturday morning, a fix by nine, and a Monday that looks like any other. If you cannot write the before and after in two short paragraphs, you do not yet know what you are selling.

The four kinds of value

Most business software delivers value in one or more of four ways: it makes money, it saves money, it saves time, or it reduces risk. Being precise about which ones apply matters, because they are priced differently and argued differently.

Kind of value	How it shows up	Cronbell	Doorframe
Money made	More revenue, sooner	Weak	Indirect: more inspections per week
Money saved	A cost avoided or replaced	Replaces a homegrown script someone maintains	Replaces admin time typing up reports
Time saved	Hours given back	A morning per incident not spent diagnosing	About an hour per inspection report
Risk reduced	A bad day that does not happen	The main one: the missed failure	The missed defect that becomes a complaint

Doorframe is the second fictional company in this book. Marcus built it for property inspection firms: an inspector walks a building with a phone, and Doorframe turns the notes and photos into a report the client can act on. It is a useful contrast to Cronbell because its value is mostly time saved on a repeatable task, whereas Cronbell's is mostly risk reduced.

Risk reduction is the hardest kind of value to price, because the value arrives on the day something goes wrong, and on every other day the product looks like a cost. You will hear this in interviews as "we didn't think we needed it until the night we did." Write that down. It is your best pricing argument, and it is also why those customers rarely leave.

The value driver

A value driver is the thing that, when it increases, increases the value the customer receives. It is not a feature and it is not a price. It is a quantity in the customer's world.

For Cronbell, three candidates present themselves:

- The number of jobs that matter. More jobs, more things that can fail silently.
- The cost of a missed failure. A nightly billing run failing is worse than a cache warm-up failing.
- The number of people who need to know. A solo developer alerts himself; a platform team alerts an on-call rotation.

For Doorframe the candidates are inspections per month and, more weakly, the length of each report. Notice that in neither case is the obvious candidate "number of users". This list of drivers is the shortlist for the value metric later in the book, but do not choose yet. The interviews will tell you which driver customers themselves reach for when they explain why the product is worth having.

The alternative

Every price is compared to something. Nobody evaluates Cronbell in a vacuum; they evaluate it against what they would do instead. Name those alternatives now, honestly, including the unflattering ones:

- Doing nothing, and finding out on Monday that Friday's job failed.
- Cron's built-in email, which everyone filters to a folder within a week.
- The Slack script a senior engineer wrote in an afternoon two years ago, which nobody dares touch.
- A large application-monitoring suite the company already pays for and has never configured for scheduled jobs.
- The competitor at twenty dollars per seat.

The alternatives set the band your price will sit in. The homegrown script is free to buy and expensive to keep; the monitoring suite is expensive and unused. Those two facts will matter more to the numbers than anything about your costs.

Write the hypothesis

Put it all on one page in this shape:

> For [the reader], [product] [does this outcome] so that [this consequence]. The value grows with [driver]. The alternative they compare us to is [X].

Cronbell's version: "For small engineering and data teams, Cronbell tells the right person within minutes when a job that matters did not run, so failures are fixed before anyone downstream notices. The value grows with the number of jobs that matter. The alternatives are a homegrown Slack script and a monitoring suite nobody configured."

It is a hypothesis. It might be wrong about the driver, and it is almost certainly wrong about the alternative in at least a few cases. That is fine. Your task for this chapter is to have this page written, with the outcome in one sentence, three candidate drivers, and the list of alternatives, before you send a single interview request. Bring it to every call, and do not show it to anyone you interview.

Interview Ten Customers

Ten is not a magic number. It is enough to see a pattern and few enough that you can do it yourself, in a week, without a research budget. In my experience the pattern usually appears around the sixth conversation; the remaining four confirm it and give you one or two useful outliers. A week is a forcing function. Interviews that are "ongoing" never end, and pricing decisions wait on them forever.

Who to interview

Do not interview only your fans. A sample of happy customers will tell you the product is wonderful and the price is fine, and you will learn nothing about the people who said no.

Segment	How many	Why
Happy paying customers, renewed or expanded	3	They can tell you what the value actually is, in their words
Signed up in the last sixty days	3	They still remember evaluating: what they compared, what they feared
Lost: churned, or evaluated and declined	2	They will tell you the alternative and the price ceiling
Prospects currently evaluating	2	They will react to the value story live, without loyalty

For Cronbell, Priya picked her six-hundred-pipeline data team and two other established customers, three recent signups, the solo developer who keeps complaining, a team that trialled and went back to their own script, and two prospects from her inbound queue.

Getting the meetings

Send a personal email from the founder. Twenty-five minutes, on a call, no selling, no demo. Say plainly that you are working on pricing and want to understand how they think about the product. People are generous with founders who are honest about what they are doing. Offer a small thank-you if you like; most will not need it.

Expect to ask about twenty people to get ten. Send the requests on a Monday, hold slots Tuesday to Friday, and take the first ten that land. Do not chase a particular individual for a week; the sample matters more than any one person in it.

How to run one

The full script is in the appendix. The conduct matters more than the wording.

Ask for stories, not opinions. "Tell me about the last time a job failed and nobody noticed" will get you a specific incident with a cost attached. "How important is monitoring to you?" will get you "very".

Ask for numbers in their words. How many jobs, how many people, how long did it take, how often does it happen. Do not supply the units; listen for which units they reach for on their own. Cronbell's customers reached for "pipelines", not "seats", and that single observation reshaped the pricing.

Do not pitch, and do not defend the current price. When someone says it is too expensive, the right response is "what were you comparing it to?", not an explanation.

Ask about price last, and ask ranges, not a single number. If you ask "would you pay fifty dollars?" most people say yes to be polite. Instead ask four questions: at what monthly price would it be so cheap you would doubt it; at what price is it a bargain; at what price is it expensive but you would still pay; at what price is it out of the question. The four answers give you a band, and the band is what you can use.

Record with permission, and take notes on five things: the outcome in their words, the driver they reach for, the alternative they named, the price band, and anything that surprised you.

Scoring the interviews

Keep one sheet with a row per interview. Fill it in within an hour of each call, while you still remember the tone.

#	Who	Segment	Outcome in their words	Driver they named	Alternative	Band per month
1	Data team lead, about 600 pipelines	Happy	"We find out before finance does"	Pipelines	Their own script, which rotted	$200 to $400
2	Solo developer, 3 cron jobs	Lost (complaining)	"Peace of mind"	Jobs	Cron email	$0 to $10
3	Platform engineer, about 45 jobs	Recent	"On-call stops paging for nothing"	Jobs, who gets alerted	Monitoring suite	$50 to $150
4	Analytics lead, about 120 pipelines	Happy	"Dashboards are right on Monday"	Pipelines	Doing nothing	$80 to $200
5	CTO of a 12-person startup, about 30 jobs	Recent	"I stopped checking manually"	Jobs	Cron email, then a script	$40 to $100

#	Who	Segment	Outcome in their words	Driver they named	Alternative	Band per month
6	Team that went back to their script, about 80 jobs	Lost	"Ours was good enough"	Jobs	Their own script	$30 to $60
7	Data engineer, about 250 pipelines	Prospect	"It has to page the right person"	Pipelines, on-call	Monitoring suite	$100 to $300
8	Ops lead, about 15 jobs	Happy	"Nobody emails me on Monday any more"	Jobs	Doing nothing	$30 to $80
9	Backend lead, about 400 jobs across 3 teams	Prospect	"Each team owns its own alerts"	Jobs, teams	Building it in-house	$200 to $400
10	Freelancer, about 8 jobs for clients	Recent	"Looks professional to clients"	Jobs, clients	Nothing	$15 to $40

The table is fictional, but its shape is what you should expect: noisy in the details, clear in the pattern.

Read it for patterns, not averages. Eight of Priya's ten described the product in terms of jobs or pipelines, not people. Seven mentioned who gets told as a second driver. The alternatives clustered into two: the homegrown script, which is the real competitor, and the enterprise monitoring suite, which is the real price anchor. And the price bands lined up with job counts far better than with team size: the two people at four hundred jobs and above named bands of two to four hundred dollars, regardless of how many seats they had.

What "validated" looks like

If seven or more of ten name the same outcome and reach for the same driver, your hypothesis holds and you can proceed. If they split into two clear groups with different outcomes and drivers, you have learned that you have two segments, and you may need tiers built on different fences, or eventually two products; that is valuable, not a failure. If nobody can name a number, your value story is not concrete enough yet; go back to the previous chapter and rewrite the outcome until it is about a specific bad day.

By Friday you should have the scoring sheet filled in and three sentences underneath it: the outcome customers name, the driver they reach for, and the alternative they compare you to. Those three sentences are the evidence for everything that follows.

Choose the Value Metric

The value metric is the unit you charge for: seats, jobs, invoices, contacts, gigabytes, inspections. It is the single most consequential pricing decision you will make, more than any number, because it decides whose bill grows and when. Get it right and the customers who get the most value pay the most, without an argument. Get it wrong and you spend years patching the gap with add-ons and special deals.

Six tests

Run every candidate through these. A good metric passes all six; a usable one passes five with a clear reason for the sixth.

1. **It tracks value.** When the metric goes up, the customer is getting more from you. This is the test that matters most, and the one seat-based pricing most often fails.
2. **The customer can predict it.** They can estimate their own bill before they sign, from what they already know about their business.
3. **You can measure it.** Cleanly, in the product, today, without building a new data pipeline.
4. **It grows with their success.** As the customer's business grows, the metric grows, so your revenue grows without a renegotiation.
5. **It does not punish the moment of need.** The customer should never be charged more on the day the product earns its keep.
6. **You can say it in one sentence.** "Priced by X" should not need a footnote.

Cronbell's candidates

Priya's interviews surfaced four candidates. Here is how they fare.

Test	Seats	Monitored jobs	Alerts sent	Projects
Tracks value	No	Yes	Partly	Partly
Customer can predict it	Yes	Yes	No	Yes
You can measure it	Yes	Yes	Yes	Yes
Grows with their success	Weakly	Yes	Yes	Weakly
Does not punish need	Yes	Yes	No	Yes
One sentence	Yes	Yes	Yes	Partly

Seats fail the first test outright. A one-person data team watching six hundred pipelines gets the value of six hundred pipelines, and a ten-person team watching ten jobs gets the value of ten. Seats also invite the behaviour Priya has already seen: shared logins, one account per company, and a quiet resentment about paying for the colleague who checks the dashboard once a month.

Monitored jobs pass everything. Customers described the product in these terms without prompting. A team knows roughly how many scheduled jobs it has, and the number grows as the company does. It is trivially measurable: Cronbell already counts them.

Alerts sent is the tempting one, because it looks like usage-based pricing and usage-based pricing sounds modern. It fails the fifth test badly. The night a bad deploy breaks forty jobs at once is the night the customer needs Cronbell most, and it would be the night their bill spikes. Worse, it teaches customers to turn alerts off to save money, which destroys the value the product exists to deliver.

Projects are predictable but coarse. A team with three projects and five jobs pays the same as a team with three projects and five hundred, which reintroduces the seat problem under a different name.

The choice is monitored jobs as the primary metric. Seats become unlimited, which is both a selling point and the end of the shared-login problem.

Fences are not metrics

A metric is what the bill scales with. A fence is a capability that separates one tier from the next without scaling: single sign-on, longer history, an audit log, a pager integration. Cronbell's interviews pointed clearly at one fence: who gets told. The solo developer wants an email; the platform team wants Cronbell to page whoever is on call. That is not a quantity to meter, it is a line to draw between tiers, and the next chapter draws it. Keep one primary metric and at most one secondary, and put everything else into fences, or your pricing page turns into an invoice.

Doorframe's contrast

Marcus priced Doorframe per seat and could not understand why inspection firms kept buying a single licence. When he asked, the answer was obvious: the office manager types up everyone's inspections, so the firm needs one seat whether it has two inspectors or nine. The value driver was inspections per month, and the customers said so as soon as he asked. Per-seat pricing had been charging the firm for the wrong thing and capping his revenue at the size of one desk.

Tell the customer where they stand

A metric only works if the customer can see it. The moment you choose one, it becomes a number the product must show plainly: how many jobs are monitored, what the tier allows, and how close the two are. Warn well before the limit, in the product and by email, and make crossing it a decision the customer takes rather than a surprise on an invoice. Cronbell adds a small usage line to its dashboard on the day the pricing changes. Nothing about the metric is defensible if the customer first learns about it from a bill.

When the true driver cannot be counted

Sometimes the driver customers name is real but unmeasurable. The cost of a missed failure is Cronbell's deepest value driver, and it cannot be metered. In that case pick a proxy that correlates with it, that customers accept as reasonable, and that you can count. Monitored jobs is a proxy for "things that can fail and matter". Nobody will argue with it, and that is the test.

You will have to migrate existing customers from the old metric to the new one. The chapter on saying the price out loud covers how; for now, choose. Your task for this chapter is to write the metric in one sentence, the way it will appear at the top of the pricing page. Cronbell's is: "Cronbell is priced by the number of jobs you monitor. Seats are unlimited." If yours does not fit in one breath, you have not finished choosing.

Draft Three Tiers

Three tiers is not a law of nature, but it is a good default for a reason. Fewer than three and there is no ladder; the customer chooses between yes and no. More than four and the customer has to think, and thinking is where deals stall. Three gives you a floor, a target and a ceiling. The middle tier is where you want most customers to land. The top tier exists partly to serve the customers who genuinely need it and partly to make the middle look sensible.

Draft the tiers before the prices. If you draft prices first, you will fit the tiers around numbers you like, and the tiers will stop meaning anything.

Name tiers after the customer, not the size

Small, Medium and Large tell the buyer nothing. Name each tier for the person it is for, and make sure that person appeared in your interviews. Cronbell's interviews produced three recognisable customers:

Tier	Who it is for	Seen in the scoring sheet as
Solo	One developer, a handful of jobs, wants an email or a Slack message	Rows 2, 8, 10
Team	A platform or data team with dozens to a couple of hundred jobs and an on-call rotation	Rows 3, 4, 5, 6
Business	Several teams, each owning their own pipelines, with escalation and compliance needs	Rows 1, 7, 9

When a prospect reads the pricing page, they should recognise themselves in one column within a few seconds.

Where to draw the metric lines

Use the distribution from the interviews and, if you have it, actual usage. Look for gaps. Priya's customers clustered at three to fifteen jobs, thirty to a hundred and fifty, and four hundred to eight hundred. Nobody sat at twenty or at two hundred. That is where the lines go: twenty jobs, two hundred jobs, a thousand jobs.

The rule is that a line should sit a little above where a segment naturally lands, so that growing means upgrading rather than being squeezed. A team at a hundred and twenty jobs on a two-hundred-job tier has room to grow and a clear reason to upgrade when they do. A team at a hundred and ninety feels the limit as a tax on success. If your usage data shows a cluster right under a line, move the line up.

Fences: what else separates the tiers

The metric limit is the primary difference between tiers. Fences are the secondary ones. Three rules keep them honest.

Fence on who needs it, not on what was hard to build. The pager integration took Priya a month; that is irrelevant. It goes on the Team tier because the Team customer is the one with an on-call rotation.

Never fence the core outcome. Every tier must deliver "you find out when a job fails". A Solo customer who cannot get alerted is not on a cheaper tier, they are on a broken product, and they will say so publicly.

Fence things only the bigger customer wants. Single sign-on, long retention, audit logs, custom escalation policies, a shared support channel. Smaller customers do not miss these, so they do not feel punished, and larger customers expect to pay for them.

Here is Cronbell's draft, with no prices yet.

	Solo	Team	Business
Monitored jobs	Up to 20	Up to 200	Up to 1,000
Seats	Unlimited	Unlimited	Unlimited

	Solo	Team	Business
Alert channels	Email, Slack	Email, Slack, pager and on-call rotation	Everything in Team, plus webhooks and escalation policies
History	7 days	90 days	1 year
Single sign-on	No	No	Yes
Audit log	No	No	Yes
Support	Documentation	Email	Priority email and a shared channel

Every row can be explained by pointing at an interview. The seven days of history on Solo is enough to debug last week's failure; the year on Business is there because the backend lead in row nine mentioned an audit for the first time.

Hard limits or overage

A line can be a wall or a gate. A hard limit stops the customer adding the twenty-first job until they upgrade; an overage lets them keep going at a per-unit price above the tier's rate. Walls are simpler to explain and are the right default for a first version. Gates are kinder to customers who briefly spike over a line, and they matter more as the lines get higher. Priya launches with hard limits and a clear upgrade prompt, and notes overage as a candidate for the first quarterly review. Either way, the limit should never silently stop monitoring a job; the core outcome is not something you fence.

The empty middle

The most common mistake in a three-tier ladder is making the bottom tier good enough that nobody leaves it, and the top tier so far away that nobody reaches it. The result is a pricing page where the highlighted middle column is the least populated. Check your draft against the interviews: would the modal customer, the one who appeared most often, land on the middle tier and be satisfied there? For Cronbell that is the Team customer, and the Team tier is where the pager integration, the ninety days of history and the two-hundred-job limit all serve them precisely. If your middle tier does not have at least one fence that the middle customer specifically asked for, it is not a tier, it is a price point.

The optional fourth column

If you already have, or expect, customers beyond the top line, you can add a "talk to us" column. You do not need it on day one. Priya adds a single sentence under the table: "More than a thousand jobs? Talk to us." That sentence is a fourth tier with no commitments.

Free tiers and trials

A time-boxed trial of the Team tier is safer than a permanent free tier. A free tier on the value metric competes with your own bottom tier forever. If you must have one, keep it below the smallest paying segment, five jobs for Cronbell, so that it converts on growth rather than replacing Solo.

Your task for this chapter is the table above, for your own product, with metric limits and fences filled in and the price row still empty. Then show it to one customer from each segment and ask a single question: which column are you? If they hesitate or pick the wrong one, the names or the lines need work.

Set the Numbers

Now the part everyone wanted to skip to. The reason to have waited is that you now have three things a guess never had: a value story in the customer's words, a metric that tracks it, and tiers that map to real people. The numbers fall out of those with surprisingly little agonising.

Three bounds

Every price sits between a floor and a ceiling, positioned against a band.

- **The ceiling is the value.** What the outcome is worth to the customer, in money, using their own stories.
- **The floor is your cost to serve plus the margin you need.** For most software it sits far below the ceiling, but it exists, and it matters for the bottom tier, where support costs can eat a small price.
- **The band is the alternatives.** What the customer would pay, in money and effort, to get the outcome another way.

The number lives well below the ceiling, comfortably above the floor, and is argued against the band.

Working the ceiling from the interviews

Take Cronbell's Team segment. Several of them told the same story: a missed failure that cost "most of a morning for two people", plus some embarrassment when finance or a client noticed first. Priya works the arithmetic, and none of it needs a study. The figures are hers; replace them with your own.

- Suppose an engineer's time is worth about a hundred dollars an hour to the business, a plausible loaded cost for a small team.

- Half a day for two people is roughly eight hours, or about eight hundred dollars.
- The interviewees estimated they had one such incident every month or two before Cronbell.

So the direct value to a Team customer is somewhere around four to eight hundred dollars a month, before counting the reduction in on-call noise that three of them raised unprompted. Now look at the bands they named: fifty to a hundred and fifty dollars. That gap is normal. Customers do not pay what a thing is worth; they pay a fraction, because the value is uncertain, the alternative is nominally free, and budgets are budgets. In my experience a price at a fifth to a third of a conservative value estimate sits comfortably inside what customers themselves say, and that is what you see here: a fifth of four hundred is eighty, a third is about a hundred and thirty.

Do the same for each tier. For Business, the stories were about several teams, and about someone downstream, sometimes a customer of theirs, noticing first; the bands were two to four hundred. For Solo, the stories were smaller and the bands were split: the complaining developer said zero to ten, the ops lead and the freelancer said fifteen to eighty.

The ladder

Three rules of thumb, each with a reason rather than a statistic.

About three times between tiers. Much more than that and the upgrade looks like a cliff, so customers stay put and squeeze. Much less and the tier is not worth the customer's attention or yours.

Price per unit falls as you go up. The larger customer gets a volume discount; you get scale and commitment. If your per-unit price rises with the tier, your best customers are being penalised for being your best customers.

Round numbers a person can remember. You and your customers will say these numbers out loud. Twenty-nine is easy; thirty-one is not.

Tier	Jobs included	Monthly price	Price per job	Ratio to previous tier	Interview band
Solo	20	$29	$1.45		$15 to $80 (excluding the outlier)
Team	200	$99	$0.50	3.4x	$50 to $150
Business	1,000	$349	$0.35	3.5x	$200 to $400

Each number sits in the upper half of its band. That is deliberate. The lower half is where the customers who evaluate on price live, and the first chapter explained what they cost you.

The Solo decision deserves a word. The complaining developer's band of zero to ten dollars is real, and it is not a segment Priya can serve at a price that pays for support. She decides, explicitly, that she is not building for that customer at that price, and that she would rather lose him than anchor the ladder on him. That is a defensible decision because it is a decision. The freelancer and the ops lead, who named higher bands, are the Solo customers she wants.

Annual pricing

Offer an annual option at about two months free: the Team tier at ninety-nine dollars a month becomes nine hundred and ninety a year. The reason is cash up front and lower churn, both of which are worth roughly that much to you. Do not go deeper than that. A forty per cent annual discount tells the customer that the monthly price was never real.

The defence sentence

For each price, write one or two sentences you could say to a customer, an investor or a new sales hire, without notes.

- Solo: "Twenty-nine dollars a month monitors up to twenty jobs with unlimited seats. It costs less than the hour you would spend rebuilding the cron email you stopped reading."
- Team: "Ninety-nine dollars a month covers up to two hundred jobs, unlimited seats, and paging your on-call rotation. The teams we built it for told us a single missed nightly job costs them more than that in one morning."
- Business: "Three hundred and forty-nine dollars a month covers up to a thousand jobs across all your teams, with escalation, single sign-on and a year of history. It is priced against what those teams told us it cost when a customer noticed a failure before they did."

If you cannot write the sentence, you do not have the number yet.

Modelling the change

Finally, map your existing customers onto the new tiers using their actual usage. Priya's forty land as fourteen Solo, twenty-one Team and five Business.

Tier	Customers	Price	Monthly revenue
Solo	14	$29	$406
Team	21	$99	$2,079
Business	5	$349	$1,745
Total	40		$4,230

Against seventeen hundred and fifty today. Now be pessimistic: suppose a third of Solo and a fifth of Team leave rather than pay more. That removes roughly five Solo customers and four Team customers, about five hundred and forty dollars, leaving around thirty-seven hundred, which is still more than double the current figure, from fewer and better-fitting customers. The model is not a forecast. Its purpose is to let you see the shape of the change before you announce it, and to notice if the shape is wrong.

Your task: three numbers, three defence sentences, one migration table. Then go back to the sentence you wrote at the end of the first chapter and rewrite it. It should no longer contain the word competitor.

Say the Price Out Loud

A price that lives only on a page is still a guess with better formatting. It becomes real the first time you say it to a person and watch their face. This chapter is about that moment, and about the two audiences you owe an explanation to: prospects, and the customers who are already paying the old price.

Five conversations before you publish

Before the pricing page changes, quote the new prices in five live conversations with prospects. Do not ask what they think of the price; say it, with the defence sentence, and then stop talking.

- They nod and ask about the next thing: the price is fine. Move on.
- They laugh: the price is either well below or well above what they expected. Ask which, plainly. Both answers are useful.
- They go quiet and then ask about discounts: this is normal, and it is not a signal to move. Hold, and use the scripts below.
- They ask how the number is calculated: good. That is what the metric sentence is for.

Five conversations is enough to catch a number that is wrong by a large factor, which is the only kind of wrong you need to catch before publishing. Fine-tuning is what the quarterly review is for.

The pricing page

Keep it plain.

1. One sentence at the top stating the metric: "Priced by the number of jobs you monitor. Seats are unlimited."
2. Three columns, the middle one visually highlighted, each with a one-line description of who it is for.
3. Under each: the metric limit, four to six fences, the monthly price, and the annual option.
4. No "contact us" on any of the three tiers. A price you will not print is a price you are not sure of.
5. A short FAQ containing the defence sentences, starting with "Why do you price by jobs?"

Resist the urge to add a comparison grid with forty rows. The interviews told you which six things each customer cares about; those are the rows.

Telling existing customers

Priya has forty customers on a per-seat price that is about to disappear. Every one of them deserves a personal explanation, and there are three broad ways to handle the transition.

Approach	What it means	When it fits
Grandfather forever	Old customers keep the old price indefinitely	Almost never; it creates a permanent two-class system and hides the true state of your revenue
Grandfather for a period	Old price honoured for a fixed window, then migrate	The default; fair to them, finite for you
Migrate now with notice	New price applies from the next billing date	Only when the change is a decrease for most customers

The default is a fixed window: honour the old price until the customer's next renewal or for ninety days, whichever is longer. For the few customers whose bill would jump by a large multiple, the data team going from seventy-six dollars to three hundred and forty-nine for example, offer the new tier at a stepped price over two or three renewals. They are your best customers and they will remember how you handled this.

The email has five parts, in this order: what is changing; why, in the same words as the defence sentence; what it means for this customer specifically, meaning the tier they land on and the number; when; and a reply address that reaches a human. In my experience customers accept a price change with a reason far more readily than a smaller change without one. "We are moving to pricing by monitored jobs because that is what you told us the product is worth" lands very differently from "we are updating our pricing".

Decoding "too expensive"

You will hear it. It means one of three things, and only one of them is about the number.

- **Not worth it to me.** A value gap. Ask what they would need to see for it to be worth it. Sometimes the answer is a fence you should move; often it is a customer in the wrong segment.
- **I cannot get budget at that level.** A structural problem, not a pricing one. Offer the annual plan, a smaller tier, or a start date after their budget cycle.
- **Reflex.** Many buyers say it because it sometimes works. Silence is the correct response. Wait, and most will continue with the next question.

Never respond to "too expensive" with a discount. The moment you do, you have told the customer that the price was a guess and that the real price is whatever they push you to. If you want a lever to offer, make it something with a reason attached: annual prepayment, a case study you can publish, a longer commitment. A discount with a reason is a price; a discount without one is a confession.

Three short scripts

Quoting: "Team is ninety-nine a month. That covers up to two hundred monitored jobs, unlimited seats, and paging your on-call rotation. Most teams your size land there." Then stop.

"Can you do better?" "The price is the price for what is in the tier. If annual works for you, that is two months free. If two hundred jobs is more than you need, Solo might be the right fit, and you can move up whenever you cross twenty."

"Competitor X is cheaper." "They price per seat; we price per job with unlimited seats. For a team like yours, count your jobs and your people and see which comes out ahead. Most teams find the difference is not the price, it is whether the on-call rotation gets paged."

> The first time you say a price out loud without flinching, you will notice the customer does not flinch either.

Sales hires, investors and boards

When someone asks for the pricing cheat sheet, give them the metric sentence, the three defence sentences and the three scripts. That is the whole sheet. If a new salesperson cannot defend the price from that page, the problem is the page, and you will hear about it in the quarterly review.

Your task: five conversations, then publish; send the email to every existing customer with the reason in it; and from today, log every price objection you hear, with who said it and which tier they were looking at. The log is the raw material for the next chapter.

Revisit Every Quarter

Pricing decays. The product gets better, the customers who arrive change, competitors move, and the reasons behind your numbers drift out of date. Annual reviews are too slow; by the time you look, a year of new customers has been priced on stale evidence. Monthly reviews are thrash; nothing has had time to happen. A quarter is long enough for signals to accumulate and short enough that a wrong number does not compound.

The review is ninety minutes, from one page, and it ends with at most one change.

Who is in the room

Keep it small: the founder, whoever owns the product, and whoever talks to customers about money, which in a small company may all be one person. Sales should be present, because they hear the objections, and they should not run the meeting, because they carry the discounts. Anyone in the room can propose a change; only the log records it.

The seven inputs

Pull these before the meeting, not during it. Most take a query or a spreadsheet filter.

1. **Tier distribution.** Customers and revenue on each tier. The shape tells you whether the ladder is working.
2. **Metric utilisation.** Where customers sit within their tier's limit. A Team customer at a hundred and eighty of two hundred jobs is either about to upgrade or about to be annoyed; you want to know which before they do.

3. **Upgrades and downgrades**, with the reason for each where you know it.
4. **Win and loss reasons involving price**, from the objection log you started in the previous chapter.
5. **Discounts given.** How many, on which tier, and the reason recorded for each. If the reason column is empty, that is the finding.
6. **Churn by tier.** Not overall churn, which hides everything, but the rate on each tier separately.
7. **Cost to serve by tier.** Support tickets, infrastructure, anything that scales with a particular kind of customer.

Reading the signals

Signal	Likely meaning	Response
Most new customers land on the top tier	Prices or lines are too low for the customers you are attracting	Raise the top price, or add a line above it
Everyone sits on the bottom tier and nobody upgrades	The bottom tier is too generous, or the middle has no fence anyone wants	Move a fence, not a price
Frequent discount requests on one tier	The defence sentence is weak, or a salesperson has a habit	Rewrite the sentence; hold the price
Customers hit the limit and leave rather than upgrade	The jump to the next tier is too steep at that point	Add a step, or an overage rate
Nobody mentions price at all	Too low	Test a higher price on new customers
Cost to serve rising on one tier	A fence is missing	Move the expensive capability up a tier
Bands in new interviews shift upward	Value has grown; the number has not	Raise the price, with the reason

None of these is a rule. Each is a prompt to look closer and, usually, to have two or three more interviews with customers on that tier, using the same script as before.

Change one thing at a time

If you move a price, a line and a fence in the same quarter, you will not know which one did what, and next quarter's review will be guesswork again. Pick the strongest signal and make the one change that addresses it. Typical moves, roughly in order of how often they turn out to be the right one:

- Move a fence between tiers.
- Move a metric line up or down.
- Add an overage rate above a limit, so growth does not force an immediate upgrade.
- Raise a price, new customers first.
- Add or remove a tier.

Raising prices

The signals that say it is time: new customers consistently landing at the top, few or no price objections in the log, and a product that is materially better than it was when the numbers were set. The method: raise it for new customers first, and watch for one quarter. Then raise it for existing customers at their renewal, with notice and with the reason, exactly as in the previous chapter. Keep the ladder ratios intact; if the Team tier rises, check that Business still sits about three times above it, or the middle becomes the empty one.

Cronbell, two quarters in

Priya's first review, ninety days after launch, shows six new Business customers, three of them already above nine hundred jobs. That is the first signal in the table. She adds a fourth line rather than raising the top: a Scale tier at five thousand jobs for nine hundred and ninety-nine dollars, which keeps the ladder's ratio close to three and gives the "talk to us" sentence a number. That is her one change.

Her second review shows Solo customers churning around the sixty-day mark. Three short interviews later she has the reason: they never set up Slack, and seven days of history feels stingy. She does not touch the price. She moves history on Solo to thirty days and adds a setup prompt in the product. Churn on that tier improves the following quarter. Both stories are fictional, but the shape of them, one signal, one change, one quarter of watching, is exactly what the review is for.

The pricing log

Keep a single document with five columns: date, what changed, the reason, what you expected to happen, and what actually happened. Every quarterly review adds a row or confirms that nothing changed and why. Two years from now this log is the answer to every investor question about pricing, and the reason a new head of sales does not quietly reinvent your discounts.

Your task: put the first review in the calendar now, ninety days out, and list the seven inputs with where each will come from. At the review, change one thing, write the row in the log, and book the next one.

Conclusion

Priya started with nineteen dollars per seat and a shrug. She finishes with a metric she can say in one sentence, three tiers named for the customers she met, three prices each with a two-sentence reason behind it, an email that her best customer replied to with "makes sense, thanks for explaining", and a date in her calendar for the next review. Her revenue is higher. More importantly, the next time someone asks why, she has an answer that came from her customers rather than from a competitor's pricing page.

The method, one more time:

1. Write the value hypothesis: outcome, drivers, alternatives, on one page.
2. Interview ten customers in one week, using the script, and score every conversation on the same sheet.
3. Choose the value metric that passes the six tests, and put everything else into fences.
4. Draft three tiers, named for people, with lines drawn just above where the segments naturally sit.
5. Set the numbers between the value ceiling and the cost floor, against the band the customers named, about three times apart, and write the defence sentence for each.
6. Say it out loud: five conversations, then the page, then the email with the reason in it.
7. Review every quarter, from seven inputs, and change one thing.

The method is deliberately small. It fits in a fortnight, it uses tools you already have, and it does not ask you to believe anything about other companies. That is on purpose. A pricing decision you can hold in your head is one you can defend in a meeting.

None of this produces the right price, because there is no right price. It produces a price with a reason, and a reason is what lets you hold the number under pressure, raise it when the evidence says to, and explain it to a customer, an investor or yourself without improvising.

You will notice the difference in a small moment. Someone will say "that seems expensive", and instead of reaching for a discount you will say the sentence you wrote for that tier, and wait. Most of the time they will nod and ask the next question. That is what a defensible price feels like from the inside.

The ten interview requests can go out on Monday. Send them.

Appendix: The Interview Script

Twenty-five minutes, on a call, in this order. Do not skip ahead to the price questions; the earlier answers are what make the later ones meaningful. Ask, then stay quiet for longer than feels comfortable. Where a question says [problem], [product] or [driver], substitute your own words from the hypothesis page.

1. **Walk me through what you were doing about [problem] the week before you started using [product].** This establishes the baseline, what life looked like without you, in their words, before they start comparing.
2. **Tell me about the last time [problem] actually happened. What did it cost you?** Stories carry numbers and consequences that opinions do not; this is where the value ceiling comes from.
3. **Who else noticed, or would have noticed if it had gone on longer?** This reveals who else receives value, and often surfaces a fence, such as who needs to be alerted.
4. **What did you try before [product], and what happened to it?** This is the real alternative, and its fate (it rotted, nobody maintained it, it was never configured) is your best argument against "we could build this".
5. **If [product] stopped working tomorrow, what would you miss first?** This is the outcome, stated in the words you will put on the pricing page.

6. **If you had twice as many [driver candidate], what would change for you?** This tests whether the driver you guessed is the one they feel; if the answer is "nothing", it is not the driver.
7. **Roughly how many [jobs, inspections, or whatever the candidate is] do you have today, and where is that number going?** This sizes the segment and tells you whether the metric grows with their success.
8. **When you evaluated [product], what did you compare the price to?** This gives you the band: the alternative's cost in money, time or headcount.
9. **Four quick ones: at what monthly price would it be so cheap you would doubt it; at what price is it a bargain; at what price is it expensive but you would still pay; and at what price is it out of the question?** Ranges are honest where single numbers are polite, and the four together show you where the number can sit.
10. **What should I have asked that I did not?** The best interviews end with something you did not expect, and this question is the only reliable way to get it.

If someone declines to answer the price questions, do not push. Note the refusal, ask question ten, and move on; the earlier answers are still worth the call. If someone starts negotiating in the middle of the interview, say that you are not selling today and return to the script.

Thank them, tell them what you will do with the answers, and fill in the scoring row within the hour.