
THE SAAS FINANCIAL MODELING BIBLE

The Complete Guide for Founders

~30 chapters

Yanni Papoutsis
Author of the Ready Series

RAISE READY
raisereadybook.com

The SaaS Financial Modeling Bible: The Complete Guide for Founders

Copyright © 2026 Yanni Papoutsis. All rights reserved.

Published by Raise Ready Ltd.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means without the prior written permission of the publisher.

raisereadybook.com

Part I: Foundations

Chapter 1: What a SaaS Financial Model Actually Is (and What It Isn't)

A SaaS financial model is not a dashboard. It is not a set of pretty charts. It is not a five-year forecast with arbitrary percentage growth rates plugged in at each step. A SaaS financial model is a structured, logic-driven representation of how a software business will acquire customers, generate revenue, incur costs, convert cash, and grow or sustain itself over time. It is a system for stress-testing your business assumptions against reality, identifying which variables matter most, and communicating to investors exactly how you plan to build a substantial company.

The word 'model' here is critical. A model is not a prediction. It is not meant to be right. It is meant to be useful. A financial model's job is to force discipline into your thinking. It requires you to define exactly how customers are acquired, exactly what they pay, exactly what it costs to serve them, and exactly when cash flows in and out. The moment you try to specify these things precisely, gaps in your logic become visible. Assumptions that seemed airtight become questionable. And the model becomes a tool for making better decisions.

What a SaaS model is not: It is not a replacement for judgment. A model with poor assumptions is worse than no model at all because it generates false confidence. It is not a crystal ball. A five-year projection is not a forecast of the future; it is a statement of what needs to be true for your business to succeed. It is not a one-time document that you build and then ignore. A financial model lives. It is updated monthly, refined quarterly, and challenged constantly. It evolves as you learn more about your customers, your cost structure, and your market.

A SaaS financial model serves three distinct audiences, and a professional model must work for all three. First, it serves you. It is the tool you use to make decisions about where to spend money, what to prioritize, when to hire, and whether you are on track. Second, it serves your board or investors. It communicates your vision for how the business will scale, proves that the unit economics work, and shows that you have thought through all the key drivers. Third, it serves due diligence. When you raise capital, investors will stress-test every assumption in your

model. They will ask hard questions about customer acquisition costs, churn rates, expansion revenue, and cash burn. A model that survives due diligence is one that has internally consistent logic, realistic assumptions grounded in data, and clear documentation of how every number is calculated.

The Three Layers of a SaaS Model

Professional SaaS models have three distinct layers, each serving a different purpose. The first layer is the assumptions layer. This is where you document every single input to your model: your customer acquisition assumptions, your pricing, your churn rates, your hiring plan, your cost structure, and everything else that drives the business. The assumptions layer is the foundation. If someone reads only the assumptions and nothing else, they should understand exactly how your business works. This layer should be auditable and defensible. Every number should have a source or a justification.

The second layer is the calculations layer. This is where the magic happens. Taking inputs from the assumptions layer, you calculate what it means for the business. You compute monthly recurring revenue from customer acquisition and churn. You calculate gross profit by subtracting cost of goods sold from revenue. You compute unit economics like customer acquisition cost and lifetime value. You build out headcount and payroll. You project cash flow. This layer is where assumptions become actionable insights. The calculations layer is also where most errors occur. A single formula error can cascade through an entire model and invalidate all downstream conclusions. That is why models must be built carefully, with clear cell logic, and with sensitivity analysis to catch errors.

The third layer is the output layer. This is the dashboard, the summary sheets, the charts, the metrics that you show to investors or your board. The output layer distills the calculations layer into the story you want to tell. It shows revenue growth, unit economics, path to profitability, cash runway, and key metrics that matter for your business. The output layer should be clean, clear, and immediately understandable. If someone has to ask what a chart means, the output layer has failed.

Why Most SaaS Models Are Wrong

Most founder-built SaaS models are deeply flawed, not because founders are stupid, but because building a financial model is genuinely difficult and requires discipline that most founders have not been trained in. The most common mistakes are: (1) Assumption layering. Founders build a single sheet with all inputs and calculations mixed together. When someone asks where a number comes from, there is no clear answer. (2) Inconsistent time periods. Some numbers are monthly, some are annual, and the relationships between them are unclear. (3) Deterministic single-point forecasts. The model shows a single growth path with no sensitivity analysis or scenario planning. In reality, your business could perform well above or well below the base case, and investors need to see how the model responds in those scenarios. (4) Missing cash flow reconciliation. Revenue does not equal cash. If you do not model deferred revenue, prepaid expenses, and working capital correctly, your cash runway calculation will be wrong. (5) Disconnected unit economics. The customer acquisition cost, lifetime value, and payback period are calculated independently without a logical foundation that ties them back to the

revenue and cost projections.

A professional model ties all of these pieces together with complete internal consistency. Every number connects to the source assumption. Revenue, costs, headcount, cash flow, and metrics are all calculated from the same underlying drivers. When investors ask 'why did churn increase in month 18?', you can trace it to the specific assumption change. When they ask 'what is the payback period on customer acquisition?', the answer is calculated directly from the monthly cohort data, not guessed. This consistency is what separates investor-grade models from founder sketches.

Chapter 2: The Architecture of a Fundraising-Grade Model (Tabs, Flow, Structure)

The structure of your model matters as much as the content. A well-architected model is easy to understand, easy to update, easy to audit, and easy to defend. A poorly architected model is confusing, error-prone, and loses credibility the moment an investor digs into it. There is a standard architecture that institutional investors and venture firms expect, and deviating from it without good reason signals that the builder may not have experience with professional models.

A comprehensive SaaS model is built across multiple interconnected tabs, each with a specific role. The typical structure includes: (1) an assumptions tab where all inputs are documented; (2) a revenue sheet that calculates MRR, ARR, and recognizes revenue monthly; (3) a costs sheet that projects COGS, operating expenses, and headcount; (4) a P&L that brings together revenue and costs into an income statement; (5) a cash flow statement that converts accrual-based profits into actual cash movement; (6) a balance sheet that documents assets, liabilities, and equity; (7) a metrics tab that calculates unit economics, LTV, CAC, magic number, NRR, and other KPIs; (8) scenario tabs for base case, bull case, and bear case; and (9) a summary dashboard that pulls the most important numbers from all the other tabs.

The Assumptions Tab: The Foundation of Everything

The assumptions tab is the most important tab in the entire model. It is where every input is documented, sourced, and justified. A strong assumptions tab allows someone to understand your entire business model without reading a single formula. The assumptions tab should be organized by category: market assumptions, customer acquisition assumptions, revenue assumptions, cost of goods sold assumptions, operating expense assumptions, headcount assumptions, and financing assumptions.

Within each category, every assumption should have four things: a clear label, the value, the unit, and a brief justification. For example: 'TAM (Total Addressable Market): \$50 billion USD (Source: Gartner 2024 report, Enterprise Software category)'. 'Monthly Churn Rate: 3.5% (Historical data from year 1 cohorts, normalized for post-launch maturation)'. 'Blended CAC: \$8,500 (Sales team fully loaded cost of \$120k/year plus 30% marketing overhead, divided by 18 customer closes per year)'. Do not assume the reader knows what you mean. Document

it.

One critical principle: Every assumption should be based on either historical data or a clear, defensible logic. Do not guess. If you do not have data yet because you are pre-launch, say so. Make a reasonable assumption based on comparable companies and clearly flag it as a forecast, not history. Investors will push back on assumptions without data, but they will respect a founder who is honest about it. They will not respect a founder who presents guesses as facts.

Chapter 3: The Assumptions Tab and Why It's the Most Important Sheet

The assumptions tab is where your entire business model lives or dies. Everything downstream—every calculation, every metric, every projection—is built on the foundation of these assumptions. If the assumptions are weak, the entire model is weak. If the assumptions are sound and auditable, the model has credibility even if some of the specific outputs turn out to be wrong. Professional investors understand this. They spend more time analyzing the assumptions tab than any other part of the model because they know that is where the real thinking has to happen.

A professional assumptions tab is organized into clear sections with consistent formatting. Each section should be self-contained and should document all the assumptions needed to build that part of the model. For example, the revenue assumptions section should document the number of starting customers, the monthly growth rate, the average contract value, the payment terms, and the recognized revenue monthly. The operating expense assumptions should document the hiring plan, salaries by role, benefits, rent, software licenses, and everything else that costs money. The COGS assumptions should document hosting costs, payment processing, support labor, and anything that varies with revenue.

Beyond just listing assumptions, a professional assumptions tab explains the logic. Why is churn 5% per month? Because customer cohort analysis shows that 60% of customers retain after year one, which calculates to a 5% monthly churn rate. Why is CAC \$12,000? Because your sales team costs \$150k per year fully loaded, each salesperson closes 15 deals per year, so the fully loaded cost per deal is \$10,000. Plus marketing overhead at 20%, so \$12,000 total. This logic allows an investor to either accept or challenge the assumption. They might say 'I think CAC is too low because you are not accounting for marketing overhead', or 'I think churn will be higher because your customer profile is more at-risk than typical cohorts'. Those are productive conversations.

One advanced technique used by professional modelers is scenario-based assumptions. Instead of a single set of assumptions, you document a base case, a bull case, and a bear case. The base case reflects what you think is most likely given current data. The bull case reflects the upside if execution is excellent and the market tailwinds are strong. The bear case reflects a more conservative view or what happens if some key assumption turns out to be wrong. By modeling all three, you show that you have thought about risks and upside, not just blindly projected the most optimistic scenario.

Finally, the assumptions tab should include a sensitivity analysis section. This shows how sensitive the model is to changes in key variables. For example, a 1% increase in churn might reduce three-year revenue by 15%. A 10% decrease in CAC allows you to hit profitability one year earlier. These sensitivities tell you which

assumptions matter most and which are second-order. They also give investors a sense of how resilient the business is. A model that is devastated by small changes in key variables is riskier than one that is relatively robust.

Part II: Revenue Modeling

Chapter 4: Driver-Based Revenue Modeling (MRR/ARR from First Principles)

The foundation of any SaaS financial model is revenue modeling, and the foundation of revenue modeling is understanding the drivers of MRR (Monthly Recurring Revenue) and ARR (Annual Recurring Revenue). Most founders think about revenue as 'How many customers times average price?', which is a start, but it is insufficient. Professional revenue modeling requires that you understand where every dollar of revenue comes from, when it arrives, how long it stays, and what assumptions drive it up or down.

MRR is a snapshot: the amount of revenue that will recur every month based on the current customer base and their subscription values. If you have 100 customers paying \$10,000 per month, your MRR is \$1,000,000. ARR is simply MRR times 12. But here is where most models go wrong: They treat MRR as a single number that grows from month to month, but they do not model the components that drive it. In reality, MRR in any given month is the result of four distinct flows: (1) the MRR you carried in from the previous month (the starting base), (2) minus the MRR you lost due to churn, (3) plus the new MRR from new customer acquisition, plus the additional MRR from expansion within existing customers.

This framework—starting MRR, minus churn, plus new business, plus expansion—is the foundation of professional SaaS revenue modeling. It sounds simple, but it forces you to model each component separately. New customer acquisition and expansion revenue have completely different characteristics. Expansion depends on your existing customer base getting larger contracts or upselling to additional products. New customer acquisition depends on your sales and marketing efforts. They should be modeled independently.

The Components of MRR Growth

Starting MRR is the MRR you carried into the month from previous months. Most customers do not churn, so month over month, most of your MRR persists. The starting MRR is calculated simply as the previous month's ending MRR. The key is to build this correctly so there is no circular dependency that breaks your model.

Churn MRR is the amount of MRR lost when customers cancel their subscriptions. This is typically modeled as a percentage of starting MRR. If you have \$100k starting MRR and a 5% monthly churn rate, you lose \$5k in MRR that month. The key is to be precise about what 'churn' means. Are you modeling logo churn (the percentage of customers who cancel) or MRR churn (the dollar amount of MRR lost as a percentage of starting MRR)? These are different. A company might have 5% logo churn but 3% MRR churn if the customers who leave tend to be smaller accounts. A professional model documents this distinction clearly.

New customer MRR is the MRR you acquire from new customers in the current month. This depends on your customer acquisition assumptions: how many customers you acquire, and what is their starting ACV (Average Contract Value). If you acquire 25 new customers and the ACV is \$4,000, your new customer MRR is \$100k. But be careful: in most SaaS models, customers do not start their subscription mid-month, so the timing matters. If you acquire customers uniformly throughout the month, the average new customer contributes half a month of revenue in their first month. This is called the half-month convention, and professional models use it consistently.

Expansion MRR is the additional revenue generated from existing customers, either through upsells to higher tiers, additions of more seats, or add-on products. Some SaaS businesses have substantial expansion revenue (examples like Slack and Zendesk can derive 30-50% of growth from expansion), while others have almost none. You cannot model expansion MRR at all if you do not understand your customer behavior deeply. Are your customers buying larger teams over time? Are they upselling to higher tiers? Do you have add-on products? How much expansion happens in an average customer relationship? This should be based on historical data, not guesses.

Chapter 5: Modeling New Business, Expansion, Contraction, and Churn Separately

A sophisticated SaaS revenue model breaks down the drivers of MRR into separate, independently modeled components. This is different from a simple 'customer count times ACV' model because it recognizes that not all revenue is created equal. New business (revenue from new customers) has different characteristics than expansion revenue (additional revenue from existing customers). Both have different characteristics than contraction (existing customers reducing their spending) and churn (existing customers leaving).

Understanding the difference matters because it changes how you think about the business and how you communicate to investors. A business with 100% growth driven by new customer acquisition and minimal expansion is fundamentally different from a business with 50% growth driven by new business and 50% from expansion. The latter has stronger retention and unit economics, stickier customers, and typically better long-term fundamentals. It is also more resilient. If new customer acquisition slows, the expansion engine keeps driving revenue. This is why Net Revenue Retention (NRR) is such an important metric for investors.

Modeling New Customer Acquisition

New customer acquisition is typically modeled as a function of your sales and marketing spend and the efficiency with which you convert that spend into customers. The basic formula is: customers acquired = (marketing pipeline generated) times (sales conversion rate). Marketing pipeline is generated through paid advertising, sales development representatives, partnerships, inbound, and other channels. Each channel has a different cost per pipeline opportunity and different conversion rates.

A professional model tracks new customer acquisition by channel and by month. If you are in a land-and-expand model, the starting ACV of new customers might be lower than the mature ACV after expansion. For example, your new customers might start at \$2,000 per month but grow to \$5,000 by year two. That expansion should be modeled separately from new customer acquisition. In your first month with 25 new customers, you add \$50k in new customer MRR (assuming \$2,000 ACV). But that is not the full value of those customers; you also get the expansion over their lifetime, which should be captured in your LTV calculation.

Modeling Expansion Within Existing Customers

Expansion revenue is the most under-modeled component of SaaS revenue. Many founders do not model it at all, either because they have not achieved much expansion yet or because they do not track it carefully. This is a mistake. Even if your expansion is currently modest (10-15% net revenue retention), modeling it properly allows you to see how much expansion contributes to your growth trajectory and where optimization efforts should go.

Expansion is best modeled as a percentage of the installed base. For each cohort of customers, you can calculate the average expansion percentage. For example, customers acquired 12 months ago are now contributing 20% more revenue than when they started. You can use historical patterns to project how expansion will grow in future cohorts. A simple approach: each customer cohort experiences expansion of X% per month on average. So a customer acquired in January at \$2,000 per month might be at \$2,100 in February (5% expansion), \$2,205 in March (5% more), and so on. Over a year, that compounds to meaningful expansion.

Modeling Churn and Contraction

Churn is often treated as a simple percentage, but professional models are more nuanced. There are at least two types of churn to consider: voluntary churn (customers actively choose to cancel) and involuntary churn (payment failure, delinquency, or administrative cancellation). These have different rates and different implications. Involuntary churn might be addressable with better payment retry logic or customer outreach. Voluntary churn signals a fundamental product or fit issue.

Additionally, not all churn is binary. Many customers do not fully cancel; they contract to a smaller plan or reduce the number of seats. A professional model captures this separately from full churn. If 8% of customers fully churn and 3% contract to a smaller plan, those are different movements that should be tracked. Contraction is particularly important if you sell based on usage or seats because customers are still paying you, just less.

Understanding contraction allows you to target at-risk customers before they fully churn and address the product issues that are causing them to reduce spending.

Chapter 6: Cohort-Based Revenue Forecasting (The Gold Standard)

Cohort analysis is the most rigorous and investor-credible approach to revenue modeling for SaaS businesses. Instead of modeling a single blended growth rate that applies to the entire customer base, cohort analysis models the behavior of customer cohorts (groups of customers acquired in the same month or quarter) independently. Each cohort starts at the same point and evolves according to observed historical patterns. By modeling cohorts rather than trying to predict aggregate growth, you are using historical data to inform projections, not making up growth rates.

A cohort analysis starts with a cohort table. The rows represent customer cohorts (customers acquired in January, February, March, etc.). The columns represent months since acquisition (month 0, month 1, month 2, etc.). Each cell in the table shows the MRR generated by that cohort in that month. For example, the January cohort in month 0 (their first month) might generate \$50k in MRR. In month 1, after some churn and expansion, they might generate \$47k in MRR (down from churn, up from expansion). By month 12, they might generate \$48k in MRR as expansion catches up to churn.

The power of cohort analysis is that you can see patterns. If every cohort loses 5-8% of MRR in their second month due to churn, you can apply that same pattern to future cohorts. If expansion consistently adds 2% per month to customer MRR across all cohorts, you can project that new cohorts will follow the same pattern. You are not guessing; you are interpolating from historical patterns. When an investor asks 'why do you think churn will be stable?', you can show the cohort table and point to the historical pattern. When they ask 'how much expansion revenue is embedded in your forecast?', you can calculate it directly from the cohort expansion percentages.

Building a Cohort Table from Historical Data

To build a cohort table, you start with historical data. Go back through your customer database and categorize every customer by acquisition month. For each cohort, calculate the MRR at each month of their lifecycle. This requires tracking customers longitudinally—not just looking at total company MRR, but understanding which part of that MRR comes from which cohort.

As you build the table, you will see patterns emerge. Do all cohorts stabilize after month 6? Do early cohorts have higher churn than later cohorts (suggesting product improvements)? Do all cohorts show consistent expansion, or does expansion vary by cohort (suggesting that customer profile or product fit varies)? These patterns are gold for an investor because they show that you understand your business at a detailed level.

Once you have historical data filled in, you can extend the table forward with projections. For months beyond your historical data, you apply the observed patterns. If months 3-6 historically show 2% churn and 1% expansion for a net decline of 1%, you apply the same to future cohorts. For months 7-12, if you see 1% churn and 2% expansion for a net growth of 1%, you apply that. This creates a forward projection that is grounded in data, not speculation.

Chapter 7: Pricing Model Variations (Per-Seat, Usage-Based, Hybrid) and How Each Changes the Model

The pricing model you choose fundamentally changes how you build your revenue model. Per-seat pricing, usage-based pricing, and hybrid pricing models all have different mechanics, different cash flow implications, and different unit economic characteristics. A professional model reflects these differences accurately.

Per-Seat Pricing

Per-seat pricing (also called per-user pricing) is the most common SaaS pricing model. Customers pay a fixed price per user or per team per month. Examples include Slack, Zendesk, and HubSpot. The revenue model is straightforward: customers times average seats per customer times price per seat equals MRR. The key variables to model are: (1) starting seats per customer when they are acquired, (2) net seat growth per customer over time (most customers add seats as they grow), and (3) the pricing per seat by product tier.

The advantage of per-seat pricing for modeling is that it is predictable. You can see historical data on seat expansion and use that to project future expansion. The disadvantage is that it creates a challenge when customers want to scale. If a customer is paying \$100 per seat and wants to add 50 new teams, the incremental cost to them is very high, and you risk losing them or having them find a way to work around the per-seat constraint. Many companies that start with per-seat pricing eventually move toward usage-based pricing or hybrid pricing as they scale to avoid this friction.

Usage-Based Pricing

Usage-based pricing (also called consumption-based or metered pricing) charges customers based on how much they actually use the product. Examples include cloud infrastructure (AWS, Azure, Google Cloud), observability tools (Datadog), and communication platforms (Twilio). The revenue model is different: MRR depends on total customer usage in the month, not customer count times fixed price. If total company usage is 100 million API calls per month and you charge \$0.0001 per API call, MRR is \$10k.

Usage-based pricing requires a different modeling approach. Instead of cohorts based on customer count, you model cohorts based on usage volume. Historical data shows how usage patterns evolve for a given customer or

customer segment. A customer acquired in January might use 1 million API calls per month in month 0, grow to 2.5 million per month by month 6, and level off at 3 million per month by month 12. By modeling these patterns, you can forecast future usage and revenue.

The advantage of usage-based pricing is that it aligns perfectly with value and removes friction from customer growth. As a customer scales, they naturally pay more. There is no 'seat hoarding' or negotiation about cost per user. The disadvantage is that revenue is less predictable. A single large customer doubling their usage can materially increase MRR, while churn manifests as usage decline, not cancellation. This makes forecasting and budgeting harder.

Hybrid Pricing Models

Many SaaS companies use hybrid pricing: a base fee per month plus usage charges. For example, a customer might pay \$5,000 per month for a base tier of features and then pay \$0.50 per API call for usage beyond a certain threshold. The revenue model becomes a combination of per-seat and usage-based modeling. You forecast the base revenue (customers times base fee) and add a forecast of variable revenue based on usage patterns.

Hybrid pricing is powerful because it offers two things: (1) predictable baseline revenue from the fixed component (which customers commit to), and (2) upside from usage (which grows as customers expand). The modeling challenge is that you have two independent variables to track and project. But hybrid pricing is increasingly the standard because it works well for both customers and the vendor.

Chapter 8: The Difference Between Bookings, Billings, and Revenue Recognition

A critical mistake many founders make is confusing bookings, billings, and revenue. These three measures are fundamentally different, and your financial model must track all three correctly. Confusing them will mess up your cash flow forecasting and your revenue projections.

Bookings is the total value of contracts signed in a period, regardless of when revenue is recognized. If you sign a 12-month contract worth \$120,000 in January, that is \$120,000 in bookings for January. Billings is the amount of money actually invoiced to customers in a period. If that \$120,000 contract is billed upfront in January, that is \$120,000 in billings. If it is billed monthly (25% upfront, 25% in quarter 2, 25% in quarter 3, 25% in quarter 4), then you have \$30,000 in January billings, \$30,000 in April billings, etc.

Revenue, under accrual accounting, is the amount recognized according to revenue recognition rules (typically ASC 606). For the \$120,000 annual contract, regardless of billing, you recognize \$10,000 per month in revenue under the subscription model. So in the month of January, you have \$120,000 in bookings, some amount of billings (depending on payment terms), and \$10,000 in revenue recognition.

This matters enormously for cash flow. If you sign a \$120,000 contract upfront and bill it upfront, you get \$120,000 in cash in January. But your P&L only records \$10,000 in revenue that month. For the first three months, you have cash in the bank that has not yet been recognized as revenue. This is deferred revenue, a liability on your balance sheet. This gap between cash and revenue is critical for forecasting your runway and understanding your cash position.

Professional models track all three. The bookings tab or forecast shows what sales you expect to close. The billings forecast shows when you expect to collect cash based on payment terms (net-30, net-60, upfront, or a mix). The revenue forecast shows what you will recognize on your P&L each month according to ASC 606. They are three different pictures of the same underlying business, and all three matter for different purposes: bookings drive sales targets, revenue drives investor growth narratives, and billings drive cash forecasting.

Part III: Cost Structure

Chapter 9: COGS in SaaS (Hosting, Support, CS - What Belongs and What Doesn't)

Gross margin is one of the first metrics venture investors look at in a SaaS business. A high-margin business (70%+) is inherently more valuable than a low-margin business because it generates more cash to reinvest in growth. But gross margin calculations depend critically on what you include in COGS (Cost of Goods Sold). Get this wrong, and your margin calculation is wrong, which leads to wrong decisions about pricing and wrong assumptions about scalability.

COGS in SaaS includes only the direct costs required to deliver the product to customers. These are costs that scale directly with revenue. The key costs typically included are: (1) hosting and infrastructure costs (servers, cloud providers, CDNs, backups), (2) payment processing fees (typically 2-3% of subscription revenue), (3) third-party API costs and integrations that scale with usage, (4) customer support and customer success labor directly tied to delivering the product to customers, and (5) any hardware or physical goods shipped to customers.

What is NOT included in COGS: Sales and marketing (even though it is required to generate revenue, it does not scale directly with each customer), product development and engineering (building the product is not the same as delivering it), administrative overhead, and fixed infrastructure costs that do not vary with customer count or usage. The rule of thumb is: If the cost goes away or decreases when you lose a customer, it is COGS. If it persists regardless, it is operating expense.

Hosting and Infrastructure Costs

Hosting costs are often the largest component of COGS. For most SaaS companies, this is cloud infrastructure from AWS, Google Cloud, or Azure. The cost structure is typically pay-as-you-go, with costs depending on

compute (vCPU), storage, and bandwidth. As your business grows and you serve more customers, your infrastructure costs should scale roughly linearly with usage. However, not perfectly linearly because of economies of scale. A single customer using 1TB of storage costs much more per GB than 1,000 customers sharing 1PB of storage.

A professional model should project infrastructure costs based on usage expectations, not as a flat percentage of revenue. If you understand your customer usage patterns (API calls, data stored, reports generated, etc.), you can estimate monthly infrastructure costs. Track this historically as you grow, and you can project how infrastructure costs will scale. Many SaaS companies see infrastructure costs as 15-30% of gross revenue early-stage, but this should decline as fixed costs are spread across larger scale.

Payment Processing Fees

Every dollar of subscription revenue requires payment processing. Stripe, Square, and other processors charge 2-3% of the amount processed (plus a fixed fee per transaction). These costs are trivial at the transaction level but material in aggregate. For a \$10 million ARR business, payment processing is \$200,000-\$300,000 per year. This is unambiguously COGS.

In your model, assume a percentage of revenue (typically 2.5% for Stripe, 3% for alternatives). This is variable with revenue, making it perfect for the COGS line. One subtlety: if customers pay annually, you might receive the full year's payment upfront and split the processing fee. Some models batch process at specific times. The exact mechanics depend on your billing cadence, but the aggregate cost is straightforward to forecast.

Support and Customer Success as COGS

Here is where many founders get confused. Is support a COGS cost or an operating expense? The answer is: it depends. The COGS part of support is the work directly required to keep customers happy and using the product—responding to feature requests, fixing bugs they report, basic training, etc. Operating expense is the work around building better support infrastructure, creating knowledge bases, hiring managers, and strategic customer success initiatives.

A better way to think about it: the direct labor cost of supporting customers at your current scale is COGS. If you employ 3 support engineers and each handles 50 customers, the fully loaded cost of those 3 engineers (\$500k per year) should be in COGS. But the manager overseeing them (\$150k) is arguably operating expense because that managerial function does not scale directly with customer count. Similarly, the cost of your support ticketing system and knowledge base is arguably fixed infrastructure cost and operating expense, not COGS.

This can get nuanced, and different companies allocate support differently. The key is to be consistent and to document your allocation. If you allocate support as 20% of revenue, justify it. Show that it is based on actual hours and burdened cost of your support team divided by revenue. This allows investors to test your assumptions and to compare you to other companies.

Chapter 10: The Hiring Plan (The Most Expensive Line Item Done Right)

For most SaaS companies, headcount is by far the largest expense category. In a typical bootstrapped or VC-funded SaaS business, fully loaded labor costs are 50-75% of total operating expense. Getting your hiring plan right is therefore critical. A hiring plan that is too aggressive will burn cash unsustainably. A hiring plan that is too conservative will leave growth on the table and cause you to lose talented people to competitors.

A professional hiring plan is not just a list of people you want to hire. It is a detailed forecast of when you hire each role, what they cost (base salary, benefits, taxes, equipment, all-in), and how they contribute to the business. The hiring plan should be built bottom-up by department, with clear logic connecting hiring to business drivers.

Engineering and Product Headcount

Engineering is typically the first department you build because you need engineers to build the product. The hiring plan should reflect the phases of your product roadmap. Early stage, you might hire a VP of Engineering and 4-5 mid-level engineers. As you grow, you add more engineers, but also layer in engineering managers, DevOps specialists, QA, and specialized roles. A typical SaaS company has one engineer per \$1-2M in ARR at scale, though this varies widely based on product complexity.

The model should track (1) headcount by role and level, (2) the fully loaded cost of each role (base salary, bonus, benefits, payroll taxes, equipment), and (3) the contribution of engineering to the business (primarily enabling product development, but also customer success tools, analytics, and operational efficiency). When you hire a new VP of Engineering at \$250k all-in, why? Because you need that person to manage the growing team, or because you need leadership for a new product initiative. The hiring plan should make that logic explicit.

Sales and Marketing Headcount

Sales and marketing headcount is often the most volatile part of the hiring plan because it depends heavily on your go-to-market strategy and growth targets. If you are targeting rapid growth through direct sales, you might have a large sales team. If you are focused on product-led growth with minimal sales, you might have almost no sales headcount. A professional model reflects this explicitly.

A typical structure: Sales Development Representatives (SDRs) source and qualify leads, Account Executives (AEs) close deals, and Customer Success Managers (CSMs) expand and retain customers. You should model each role and calculate efficiency metrics. How many leads does an SDR generate per month? How many of those become pipeline? What is the close rate for an AE? These metrics allow you to work backward from your revenue target to determine required headcount. If you need 100 new customers per month and each AE closes 10

per month on average, you need 10 AEs. If each AE converts 50 qualified leads at 20% close rate, you need 250 qualified leads per month. If each SDR generates 50 qualified leads per month, you need 5 SDRs. This logic-based hiring plan is far more credible than arbitrary headcount targets.

Operating Expense Headcount

Operating expense headcount includes Finance, HR, Legal, IT, Administration, and other roles that keep the business running but do not directly generate revenue or build product. These functions scale slower than revenue. You do not need a CFO at \$1M ARR, but you probably do need one by \$20M ARR. A professional model reflects this with clear hire dates and justifications. When you hire a CFO in month 18, why? Because you are approaching Series B fundraising, or because your financial complexity has reached the point where you need dedicated CFO-level expertise.

Chapter 11: OpEx Categories and How to Model Them by Department

Operating expenses (OpEx) are all the costs of running the business that are not COGS. This includes payroll (which we covered in Chapter 10), but also rent, software licenses, legal, professional services, marketing, and a hundred other things. A professional model organizes OpEx into clear categories and models each with transparent logic.

The typical OpEx categories in a SaaS business are: (1) Headcount (payroll, benefits, taxes, equipment), (2) Rent and facilities, (3) Software and subscriptions, (4) Marketing, (5) Legal and professional services, (6) Travel and entertainment, (7) Contractors and freelancers, and (8) Other/Miscellaneous. Within each category, you should document the assumptions and the drivers.

Rent and Facilities

Rent is typically a fixed cost based on office size and location. A good model documents the square footage, cost per square foot, and when you expect to add more space. Early stage, you might be in a small shared office at \$5,000 per month. As you grow to 50 people, you might take an 8,000 sq ft office at \$15,000 per month. As you grow to 200 people, you might have a second office location. These decisions should be modeled explicitly with dates and costs.

One nuance: as more SaaS companies go fully remote or distributed, rent costs decline or disappear. Some fully remote companies assume \$0 in rent. Others maintain a small headquarters for company gatherings. The key is to be explicit about your assumption and to update it if your policy changes.

Software, Subscriptions, and Tools

SaaS companies use a lot of other SaaS. You pay for Salesforce, Slack, Figma, GitHub, AWS (in addition to the AWS bill in COGS), Stripe, monitoring tools, HR software, and dozens of other tools. In aggregate, this is often \$500-\$2,000 per employee per year. In your model, estimate this as a per-employee cost and multiply by headcount, or list out the tools you expect to use and document the cost. As you grow, you will use more tools and more expensive versions of tools, so this cost scales with headcount.

Marketing Spend

Marketing is a variable investment that should be planned based on your go-to-market strategy. If you are doing paid advertising, you might spend X% of revenue on marketing. If you are doing product-led growth, marketing spend is lower. If you are doing enterprise sales with lots of event sponsorships and content marketing, it might be higher. A professional model documents marketing spend by channel and tracks the CAC (Customer Acquisition Cost) and payback period that these dollars generate. Marketing is not an expense to minimize; it is an investment in growth with a return that should be measured and optimized.

Chapter 12: Stock-Based Compensation and How to Handle It

Stock-based compensation (SBC), also called equity or equity grants, is a critical component of SaaS company cost structure and a common source of confusion. When you grant an employee options or restricted stock units, when do you expense it? How much is the expense? And how do you model it?

Under ASC 718 (Accounting for Stock Compensation), all stock-based compensation must be expensed on your P&L. The expense is calculated using a fair value method (typically Black-Scholes for options). When you grant an employee 10,000 options over a four-year vesting period, you do not take the full expense in month one. Instead, you recognize 1/48 of the grant value each month over the four-year period. This is called amortization.

For modeling purposes, SBC is typically 8-15% of total compensation on top of base salary and benefits. So if your total payroll is \$10M per year including base, benefits, and taxes, you might add another \$1.2M for SBC as an operating expense. This is an area where many founders underestimate because the impact on cash flow is muted (the actual cash outflow of SBC is primarily when the company is first funded and equity is created, not when it is expensed). But the P&L impact is real and should be modeled.

A professional model tracks SBC by cohort of employees (early hires typically get larger grants, later hires get smaller grants) and models the expensing schedule. When you hire the first engineer at a \$100k salary in month 1 with a \$200k grant vesting over four years, the monthly cash cost is \$100k salary but the monthly P&L cost is \$100k salary plus $\$200k/48 = \$104.2k$. This is why SaaS company P&L losses are often larger than cash burn in

early stages: SBC is a big non-cash expense.

Part IV: Unit Economics

Chapter 13: CAC Calculated Correctly (Blended vs Segmented, by Channel)

Customer Acquisition Cost (CAC) is one of the most important unit economics metrics in SaaS. It tells you how much you have to spend to acquire one customer. If your CAC is \$10,000 and your customer pays \$1,000 per month, then you break even on that customer in 10 months and start generating profit after that. CAC is fundamental to understanding whether your unit economics work. Many investors will make a yes/no decision on a business based primarily on CAC and LTV (Lifetime Value). Get this right.

The formula for CAC is deceptively simple: Total Sales and Marketing Spend / Number of Customers Acquired = CAC. But the implementation details matter enormously. What is included in 'Sales and Marketing Spend'? Is it just salaries or also software, tools, events, travel? What time period? If you spend \$1M on marketing in January and close 100 customers in February, does February get credit for all that spending or some of it? How do you handle indirect spend? These details can easily move CAC by 50%.

Blended CAC vs Segmented CAC

Blended CAC is the simplest calculation: total sales and marketing spend divided by total customers acquired. This is useful as a single metric to track trends, but it is not very useful for decision-making because it hides important details. Different customer segments have very different acquisition costs. An enterprise customer acquired through a complex sales process costs \$50,000 to acquire. A self-serve customer acquired through freemium conversion costs \$500. Blending these together into a single number is misleading.

A professional model calculates CAC by segment and by channel. You model enterprise sales (high CAC, high contract value), mid-market (medium CAC, medium contract value), and SMB/self-serve (low CAC, lower

contract value) separately. You calculate how many customers each channel generates and at what cost. You look at the CAC payback period for each segment (how long until the customer pays back their acquisition cost). Enterprise might have a 12-month payback, while self-serve might have a 2-month payback. This granularity is critical because it shows you where your money is being spent well and where it is being wasted.

Fully Loaded CAC

Many companies understate CAC by only including direct marketing spend and excluding sales headcount and overhead. A more accurate picture is 'fully loaded CAC', which includes all sales and marketing costs: payroll (including base salary, bonus, benefits, taxes), software licenses, events, travel, marketing spend, agencies, and so on. If your sales team costs \$2M per year and closes 200 customers, that is \$10,000 in payroll-based CAC, even before you add any other marketing spend.

Calculating fully loaded CAC requires discipline. You need to know your total sales and marketing headcount (every role in sales, marketing, customer success, even sales operations), their fully loaded cost, and how much of their time is devoted to acquisition versus retention. A customer success manager retaining existing customers is not part of acquisition CAC; a sales development representative generating leads is. A fully loaded CAC calculation might look like: \$500k in salaries and benefits for sales, \$200k for marketing contractors and tools, \$100k for event sponsorships, divided by 100 customers acquired = \$8,000 CAC. This is much more realistic than \$500 CAC based only on advertising spend.

Chapter 14: LTV Done Right (With and Without Gross Margin, Cohort-Based)

Lifetime Value (LTV) is the total profit a customer will generate over their entire relationship with your company. It is the complement to CAC. If your LTV is \$100,000 and your CAC is \$10,000, you have a 10:1 ratio, which is excellent economics. If LTV and CAC are close, your unit economics do not work. LTV must be calculated with precision because a small error (like forgetting to subtract COGS) can make a marginally profitable customer look highly profitable or vice versa.

There are two versions of LTV: LTV with gross margin (sometimes called LTV to Gross Profit) and LTV with full margin (LTV minus all operating expenses). The most commonly used and most useful is LTV with gross margin. The formula is: Average Revenue Per Account * Gross Margin * Average Customer Lifetime (in years) = LTV. For example, if a customer pays \$2,000 per month (ARPA), you have 80% gross margin, and customers stay for 3 years on average, then $LTV = \$2,000 * 12 \text{ months} * 3 \text{ years} * 80\% = \$57,600$.

This is useful because it shows the value generated from each customer after covering the direct costs to serve them. The remaining gross profit is available to cover sales and marketing, operating expenses, and eventually profit. An LTV to CAC ratio of at least 3:1 is the rule of thumb for a healthy business; ratios of 5:1 or higher are excellent.

Cohort-Based LTV

The most accurate LTV calculation is cohort-based. You track actual customer cohorts from acquisition through their lifecycle (or projected future), measure their actual ARPA and churn, and calculate LTV based on observed behavior. This is far better than using blended averages because different cohorts often have very different lifetimes.

For example, if you implemented a product improvement that reduced churn by 20% starting in month 18, customers acquired after month 18 have longer lifetimes than earlier cohorts. If your later cohorts will have longer lifetimes, your LTV is actually improving. But you would miss this if you only look at blended company averages. Cohort-based LTV forces you to see the real driver: cohort quality is improving.

To calculate cohort LTV, you take each customer cohort, calculate the average revenue they generate per month, and project their remaining lifetime based on observed churn patterns. If a January cohort has stabilized at \$1,000 per customer per month and 3% monthly churn (implying ~33 month lifetime), the LTV of that cohort is $\$1,000 * 80\% \text{ margin} * 33 \text{ months} = \$26,400$.

Chapter 15: Payback Period and the Magic Number

Payback period is the number of months it takes for a customer to generate enough revenue to cover the cost of acquiring them. It is simple but powerful. If your CAC is \$10,000 and a customer generates \$3,000 per month in gross profit (ARPA times gross margin), payback is $\$10,000 / \$3,000 = 3.3$ months. A payback period under 12 months is generally healthy; under 6 months is excellent.

Payback period tells you how quickly you recover your acquisition investment. A short payback period means you have more cash available to reinvest in growth. A long payback period means you need more capital to fund growth because cash is tied up in customer acquisitions for longer. For a business trying to reach profitability, payback period is critical because it directly determines how much cash you burn before reaching break-even.

The 'Magic Number' is a different metric that is particularly useful for tracking sales efficiency over time. Magic Number = $(\text{Quarterly ARR Growth}) / (\text{Sales and Marketing Spend in that Quarter})$. For example, if you add \$500k in ARR in a quarter and spent \$100k on sales and marketing that quarter, your magic number is 5. A magic number above 0.75 indicates efficient growth; above 1.0 is excellent.

Magic Number is particularly useful because it is a forward-looking metric that you can measure quarterly and track trends. If your magic number is declining over time, it suggests your sales and marketing efficiency is getting worse, which might indicate market saturation, increasing CAC, or other issues. If it is improving, you are becoming more efficient.

Chapter 16: Net Revenue Retention - The Metric That Predicts Everything

Net Revenue Retention (NRR) is the percentage of beginning period MRR from existing customers that is retained or grown by the end of the period, expressed as a percentage. It is calculated as: $(\text{Beginning MRR} - \text{Churned MRR} + \text{Expansion MRR}) / \text{Beginning MRR}$. For example, if you start the year with \$100k MRR, lose \$5k to churn, and gain \$10k from expansion, your annual NRR is $(\$100k - \$5k + \$10k) / \$100k = 105\%$.

NRR is the single most predictive metric for SaaS business health. A company with 100%+ NRR is adding revenue from existing customers faster than it is losing it to churn. This is the holy grail of SaaS because it means your installed base is growing in value, which allows sustainable growth even if new customer acquisition slows. Companies with 100%+ NRR typically grow faster with lower customer acquisition costs than companies with 90% NRR, even if they are in the same market.

The reason NRR is so predictive is that it captures the product's stickiness, unit economics, and the product-market fit. If NRR is below 100%, you are losing more to churn than you gain from expansion, which is ultimately unsustainable. If NRR is above 120%, you are creating a compounding growth engine where the business becomes easier to grow over time. Investors will focus heavily on NRR, especially at Series A and beyond.

In your financial model, NRR should be explicitly calculated and tracked quarterly or annually. It should be based on cohort data, not blended averages (see Chapter 14 on LTV). By tracking NRR by cohort, you can see if your product improvements are working and if newer cohorts have better retention and expansion than earlier cohorts. This is the kind of granular insight that separates investors from founders who understand their business at a deep level.

Part V: Cash Flow and Balance Sheet

Chapter 17: The Cash Flow Statement Most Founders Get Wrong

The cash flow statement is the single most important financial statement for founders. It shows when actual cash comes in and goes out. Your P&L (income statement) might show \$10M in revenue and \$2M in net profit, but you could still run out of cash and go bankrupt if that revenue is not collected or if expenses come due before revenue arrives. Understanding cash flow separates founders who survive from founders who do not.

Most founders understand this intellectually but get it wrong in their models. They build revenue projections and cost projections, then assume those convert directly to cash. They do not. Revenue is recognized over time according to GAAP rules, but cash is collected based on customer contracts and payment terms. Expenses are incurred when you commit to them, but you might not pay them for 30-60 days. This gap between revenue/expense and cash creates working capital movements that are ignored in naive models.

The Three Sections of a Cash Flow Statement

A professional cash flow statement has three sections: operating cash flow, investing cash flow, and financing cash flow. Operating cash flow is the cash generated or consumed by the business from normal operations. Investing cash flow is cash used to buy assets or make investments. Financing cash flow is cash from raising capital or debt financing. For an early-stage SaaS company, financing cash flow is usually the dominant source of cash (you raise capital), operating cash flow might be negative (you burn cash), and investing cash flow is minimal (small equipment purchases).

Operating cash flow starts with net income (the bottom line of your P&L), then adds back non-cash expenses (like depreciation and stock-based compensation), and adjusts for working capital changes. Working capital changes are the key. If your deferred revenue (cash you received upfront for services you have not yet delivered)

increases by \$500k, that is a source of cash, so you add it back. If your accounts receivable increases by \$300k (customers owe you money you have not collected), that is a use of cash, so you subtract it.

This is where most founder models fail. They do not track deferred revenue correctly, so their working capital analysis is wrong, and their cash flow forecast is wrong. A customer paying annually upfront creates a \$120k liability (deferred revenue) when you collect the cash but only \$10k in revenue that first month. The deferred revenue is a source of cash, and as you recognize revenue, deferred revenue decreases, which becomes a use of cash later. Get this wrong, and your cash runway forecast is completely unreliable.

Chapter 18: Working Capital in SaaS (Deferred Revenue, Prepaid Expenses)

Working capital is the difference between your current assets and current liabilities. In SaaS, working capital is unique because you typically have negative working capital (your liabilities exceed your assets), which is actually a good thing. It means customers pay you before you have to pay most of your expenses, creating a cash float that funds growth.

The three main working capital items in SaaS are deferred revenue, accounts receivable, and prepaid expenses. Understanding and modeling these correctly is critical for accurate cash flow forecasts.

Deferred Revenue (The Founder's Secret Weapon)

Deferred revenue (also called unearned revenue or customer prepayment) is the biggest working capital advantage in SaaS. When a customer pays for a 12-month subscription upfront, you receive cash immediately but only recognize revenue over the 12 months. The cash creates a liability on your balance sheet (deferred revenue). Each month, as you deliver service, you move deferred revenue down and revenue up on the P&L.

This creates a huge cash benefit. If you have 1,000 customers paying \$12,000 per year upfront, you collect \$12M in cash in January (assuming all customers are annual). But your P&L only recognizes \$1M in monthly revenue. You have \$12M cash in the bank and only \$1M in liability. For a company needing to fund growth, this cash float is incredibly valuable. It means you do not have to raise capital to fund the gap between revenue recognition and cash collection.

In your financial model, deferred revenue should be calculated based on payment terms. If X% of customers pay annual upfront, that creates immediate deferred revenue equal to 11 months of their revenue. If Y% of customers pay monthly, there is minimal deferred revenue. As your billing shifts from annual upfront to monthly, your deferred revenue will decline relative to revenue, which is a use of cash. This timing effect is critical to model correctly, or your cash forecasts will be wrong.

Accounts Receivable

Accounts receivable is the opposite problem. If you invoice customers on net-30 terms and do not collect immediately, you have accounts receivable (money owed to you that you have not yet received). This is a use of cash. If you sell \$1M in services in January and collect 50% in January and 50% in February, you have \$500k in accounts receivable as of January 31st. Your cash balance is lower than your revenue because you have not collected the cash.

In a SaaS business with upfront annual billing, accounts receivable is typically minimal because you are paid upfront. But in businesses with monthly billing or payment terms, accounts receivable can be significant. A professional model projects this based on your billing and collections assumptions. The industry standard for SaaS is to assume 95%+ collection rates because SaaS customers are reliable payers (they cannot use the product if payment fails). But still model the small percentage that takes longer to collect.

Prepaid Expenses and Accrued Liabilities

Prepaid expenses are the inverse of deferred revenue. If you pay for annual software licenses upfront (\$50k), you have a prepaid expense asset. Each month, you expense \$4,167 and reduce the prepaid asset. This does not affect cash in the month you expense it because you already paid in the past. But it does affect working capital.

Accrued liabilities are expenses you have incurred but not yet paid. Payroll is a good example. You accrue payroll expense in the month earned but pay it in the first few days of the next month. This creates a small working capital benefit (the liability floats the cash). Similarly, professional services, utilities, and other expenses might be accrued and paid later. In aggregate, this creates a modest amount of working capital, but it is often small compared to deferred revenue effects.

Chapter 19: The Balance Sheet That Proves Your Model Is Internally Consistent

The balance sheet is the least sexy financial statement, but it is incredibly important for validating the internal consistency of your model. The balance sheet equation is simple: $\text{Assets} = \text{Liabilities} + \text{Equity}$. If your balance sheet does not balance, you have an error somewhere in your model. And this balance sheet check catches many hidden errors.

Your balance sheet should include assets (cash, deferred revenue, capitalized software), liabilities (deferred revenue, accounts payable), and equity (starting capital, accumulated losses/profits, new capital raised). At the end of each month in your model, the balance sheet should balance. If it does not, there is an error in your cash flow statement, your P&L, or your working capital calculations.

Additionally, the cash balance in your balance sheet should exactly match the cash balance calculated in your cash flow statement. This is a critical check. If you have a cash flow statement and a balance sheet, the ending

cash each month in the cash flow statement should equal the cash line item in the balance sheet. If they differ, you have an error to find and fix.

A more advanced validation: The change in accumulated retained earnings on your balance sheet should match net income on your P&L. Starting retained earnings + net income = ending retained earnings. This is another consistency check that catches errors. If you build these checks into your model, any major errors will be immediately visible and you can fix them before presenting to investors.

Chapter 20: Runway Modeling and the Fundraising Trigger

Runway is how many months you can operate with current cash, given current burn rate. It is the ultimate founder metric for early-stage companies. It answers the question: when do I run out of money and need to raise capital or reach profitability? The formula is simple: $\text{Cash Balance} / \text{Monthly Burn} = \text{Runway in Months}$.

Most founders think about runway as a static number ('we have 12 months of runway'). But runway is dynamic. It changes every month as you burn cash and hopefully increase revenue. A professional model tracks runway monthly and projects when you will hit critical decision points. If you have 12 months of runway today but project 18 months of runway in 6 months (because revenue is growing and burn is decreasing), you might be on track to profitability without raising capital. If you have 12 months of runway but project 6 months of runway in 6 months (because burn is accelerating), you need to raise capital now.

The 'fundraising trigger' is the point at which you decide to start fundraising. Most VCs want you to raise when you have 6-12 months of runway remaining. If you wait until you have 2 months of runway, you are negotiating from a position of weakness and will get worse terms. If you raise at 12 months, you have time to run the process properly (which takes 3-4 months) and end up with capital in the bank while you still have runway. A professional model shows your fundraising trigger explicitly: 'When runway reaches 6 months, we begin fundraising.'

Part VI: Scenarios and Sensitivity

Chapter 21: Three-Scenario Planning (Base, Bull, Bear)

Most founders build a single scenario: the base case, which typically reflects what they think is most likely. But reality is a distribution. Some outcomes will be better, some worse. Professional models show all three. Base case is most likely. Bull case is the upside if execution is excellent. Bear case is what happens if things go worse than expected.

The three-scenario approach serves two purposes. First, it prepares you mentally for different outcomes and forces you to think about how you would respond. If the bear case happens, what is your playbook? Second, it shows investors that you have thought about risks and upside, not just blindly projected one optimistic path. Investors see founder models with only upside cases every day. It makes them skeptical. A model that shows realistic scenarios with documented assumptions gains credibility.

Base Case

The base case should reflect your honest assessment of what is most likely, based on historical data and reasonable assumptions. If your actual churn is 4% per month, do not assume 2% in the base case to make the model look better. If you are achieving \$8,500 CAC historically, do not assume \$6,000 CAC in the base case unless something material has changed. The base case should be defensible with data.

Bull Case

The bull case is the upside scenario. What has to go right for the business to massively outperform? Typically, this means lower CAC (sales and marketing is more efficient than expected), higher expansion revenue (customers upsell more than expected), and lower churn (product improvements and better retention than expected). Do not make the bull case unrealistic. Do not assume 0% churn or 5X expansion revenue. Make it ambitious but plausible. What would need to happen operationally to achieve bull case? This is a good thought exercise.

Bear Case

The bear case is the scenario where things do not go as planned. This might mean higher customer acquisition costs (it takes longer and more money to close deals), lower expansion revenue (customers expand slower than expected), or higher churn (product issues or market saturation causes churn to rise). The bear case is not a doomsday scenario; it is a realistic 'what if things are harder than we expect?' scenario. What is your payback period and LTV in the bear case? Can the unit economics still work?

The bear case should inform your decision-making. If the bear case results in bankruptcy in 18 months, you need a Plan B (cut costs, pivot the business, find alternative sources of capital). If the bear case still results in a viable business (lower growth, but still reaching profitability), you have a higher confidence level in the business. Investors look at the bear case to assess downside risk. A company where the bear case is still okay is a lower-risk investment than one where bear case means you are out of business.

Chapter 22: Sensitivity Analysis That Answers Investor Questions Before They Ask

Sensitivity analysis shows how a specific output (like time to profitability or three-year revenue) changes when you vary one input (like churn or CAC). It is a table (or graph) that shows: If churn changes by 1%, how does revenue change? If CAC changes by 10%, how does payback period change? This analysis is invaluable because it shows investors which assumptions matter most and how robust your model is.

A one-way sensitivity analysis varies one assumption while holding others constant. A two-way sensitivity analysis varies two assumptions simultaneously. For example, a two-way table showing how revenue changes as both churn and CAC vary. Create a table with churn along one axis (-2%, -1%, 0%, +1%, +2%) and CAC along another axis (-20%, -10%, 0%, +10%, +20%), and show the resulting three-year revenue in each cell.

A professional sensitivity analysis answers the questions you know investors will ask: 'What if CAC is 20% higher than you expect?' Show the table. 'What if churn is 1% higher monthly?' Show the impact. 'Which assumption matters most for profitability?' Calculate the impact of each assumption on the time to profitability. This preparation means you can answer investor questions instantly with data instead of guessing.

Chapter 23: The Board Reporting Model vs the Fundraising Model

The model you build for fundraising and the model you use for board reporting are related but different. A fundraising model is typically a 5-year projection showing how you will scale and become valuable. A board reporting model is a quarterly reforecast that compares actual results to prior projections and updates expectations.

For fundraising, you present a base case with clear assumptions and supporting data. You show the bull and bear cases. You show unit economics, CAC, LTV, payback period, and other metrics that prove the business works. You show a clear path to profitability or dominant market position. You might include growth traction that has been achieved to date (months 1-6 of actual data) and then project forward from month 6 onward.

For board reporting, you build a much more detailed model (often 12-24 months of monthly projections). You reforecast based on actual results in the current quarter. You compare actual revenue and spend to prior forecasts and investigate variances. You update assumptions based on what you have learned. A board reporting model is a working document that changes monthly as you learn more and adjust your outlook.

A professional practice is to build one comprehensive underlying model and then create two different 'views' of that model for different purposes. The fundraising view shows the summary at a high level. The board reporting view shows the detailed monthly breakdowns and actual vs forecast analysis. Both are built from the same underlying source of truth, just presented differently.

Part VII: Stage-Specific Guidance

Chapter 24: Pre-Seed/Seed Models (What to Include, What to Skip)

At pre-seed and seed stage, you are still proving the model works at a small scale. Your financials are typically simple, and your historical data is limited. A pre-seed/seed model should be simple enough that you can build it and update it quickly, but complete enough that it forces discipline into your thinking.

A seed-stage model typically includes a 24-36 month forecast with the following tabs: assumptions, revenue (with customer acquisition and churn modeled), cost of goods sold, operating expenses (including hiring plan), P&L, cash flow, and metrics (CAC, LTV, payback, magic number, NRR). You do not need separate scenario tabs; a base case with sensitivity analysis is sufficient. You do not need a detailed balance sheet. You need enough to show that unit economics work and that you have a path to profitability or Series A economics.

The goal at seed stage is not to predict the future accurately (you will be wrong). The goal is to show that you have thought through the business model and can execute against it. Your assumptions should be based on early traction: first paying customers, early churn rates, actual customer acquisition channels that are starting to work. Avoid assumptions that are pure guesses or not supported by any data.

Chapter 25: Series A Models (The Bar Has Changed)

By Series A, the bar is much higher. You have to show not just that the model works in theory, but that you have proven the model at scale and are on a path to be a substantial business. Series A investors expect a comprehensive financial model with multiple scenarios, granular unit economics, and detailed forecasts that are grounded in actual results.

A Series A model should include everything from the seed model, plus: multi-scenario analysis (base, bull, bear), detailed cohort analysis (showing that cohorts are stabilizing or improving), separate modeling of new customer acquisition vs expansion, by-channel analysis of customer acquisition, customer success and retention metrics, and a clear path to either profitability or an 8-figure ARR at scale. You should have 3 years of actual results (even if just traction) to ground your forward projections.

Series A investors will stress-test every assumption. They will ask about unit economics by customer segment. They will dig into churn and understand why some customers are churning. They will model out what happens if growth slows. They will calculate CAC payback and LTV and question whether your assumptions are realistic. Your model should be bulletproof before you enter the room.

Chapter 26: Series B+ and the Path to Profitability Model

By Series B and beyond, the model has shifted. Early stage is about growth and unit economics. Series B and beyond is about the path to profitability and unit economics in a mature business. Investors at this stage want to see that you can grow rapidly and also eventually be profitable. You need to model not just growth, but expansion margins and the path to profitability.

A Series B model should be comprehensive and detailed. Include a detailed hiring plan by department. Model expansion margins as you scale (typically OpEx should decline as a percentage of revenue as you get bigger). Show operating leverage in the model: that every 10% of incremental revenue eventually drops X% to operating expense, proving operating leverage. Model a clear path to profitability, typically within 5 years and often within 3.

Additionally, model the 'Rule of 40' metric: Growth Rate (%) + Profit Margin (%) should add to 40 or higher. A fast-growing unprofitable company might be at 50% growth + (-10%) margin = 40. As you scale, the growth rate will slow but the margin will improve, and you should still be above 40. This is the metric that proves you can sustainably grow and eventually be a profitable business.

Part VIII: Common Mistakes

Chapter 27: The 20 Most Common Financial Model Mistakes and How to Fix Them

After reviewing hundreds of founder financial models, certain mistakes appear consistently. These are not exotic errors; they are common mistakes that almost every founder makes the first time they build a model. By knowing these mistakes, you can avoid them.

Mistake 1: Circular References in Cash Calculations

A circular reference happens when you try to calculate something that depends on itself. For example, ending cash = starting cash + operations, but you calculate ending revenue as a function of ending cash. This creates a circular loop that either breaks the model or produces a false equilibrium. Always calculate cash flows from operations independently, then add/subtract to get ending cash. Do not make ending values depend on themselves.

Mistake 2: Mixing Monthly and Annual Numbers Without Clear Unit Conversions

Your model should be built in a consistent time period (typically monthly). If you have annual numbers, convert them to monthly. MRR should be monthly revenue. If a customer pays \$120k per year, that is \$10k per month. Make the conversion explicit and clear in your model. Mixing monthly and annual without clear conversion is a

common source of errors that make the model 12X wrong.

Mistake 3: Not Accounting for Deferred Revenue Correctly

This is the #1 cash flow error in founder models. If a customer pays \$120k upfront for a 12-month subscription, you recognize \$10k in revenue each month but get \$120k in cash in month 0. If you only project \$10k in cash that month, your cash forecast is wrong by a factor of 12. Calculate deferred revenue explicitly and adjust your cash flow statement for the working capital movement.

Mistake 4: Blending Customer Segments That Have Very Different Unit Economics

You cannot have a single CAC, LTV, or churn rate if you have enterprise, mid-market, and self-serve customers. They have completely different acquisition costs and lifetimes. Model them separately. If they account for different percentages of revenue, weight the blended metrics accordingly. But do not pretend they are the same.

Mistake 5: Projecting Customer Acquisition Based on Arbitrary Growth Targets Instead of Actual Channels

Many founder models say 'we will grow 10% per month' and then just apply that to customer count. Where does this growth come from? What channels generate it? At what cost? A professional model links customer acquisition to actual sales and marketing activities: SDRs and AEs, paid advertising, product-led growth, partners. Start with actual historical traction by channel and project from there.

Mistake 6: Assuming Zero Customer Churn

Some founder models assume customers never leave. This is fantasy. Every SaaS business has churn. If you have actual churn data, use it. If you are pre-launch, look at comparable businesses. Even the best SaaS companies have 1-3% monthly churn. Most have 3-5%. Assuming zero churn is a red flag that destroys your credibility.

Mistake 7: Underestimating Headcount and Hiring Costs

Most founder models underestimate how many people they will need to hire. They think 'I can do this with 5 people,' but then they hire and discover they need 10. The fully loaded cost of people is also often underestimated. A \$120k salary is really \$180k fully loaded (benefits, taxes, equipment, office space, etc.). Build a detailed hiring plan and use realistic fully loaded costs.

Mistake 8: Not Modeling the Half-Month Convention for New Customer Revenue

When you acquire a new customer mid-month, they do not contribute a full month of revenue that month. On average, they contribute half a month. If you acquire 100 customers in January at \$5,000 ACV, you add \$250k in revenue that month, not \$500k. The other \$250k comes in February. This half-month convention is standard in SaaS modeling. Not using it makes your first few months of revenue overstated.

Mistake 9: Forgetting to Include COGS

COGS includes hosting, payment processing, customer support labor, and other direct costs of serving customers. Forgetting COGS makes gross margin appear much better than it actually is. $\text{Gross margin} = (\text{Revenue} - \text{COGS}) / \text{Revenue}$. Get COGS right.

Mistake 10: Over-Optimistic Payback Period Calculations

Payback period should be calculated as $\text{CAC} / (\text{Monthly ARPA} * \text{Gross Margin})$. Some founder models only divide by ARPA (not including gross margin), which makes payback look much better. Others use blended customers and ARPA instead of calculating by segment. Be precise. If payback is 18 months when calculated correctly, it is 18 months, not 12 months.

Mistake 11: Calculating LTV Without Gross Margin

LTV should be calculated as $\text{ARPA} * \text{Gross Margin} * \text{Average Customer Lifetime}$. Some models calculate $\text{ARPA} * \text{Lifetime}$ without gross margin, which overstates LTV. The LTV that matters for evaluating unit economics is LTV after COGS, because that is the profit available for sales, marketing, and operating expenses.

Mistake 12: Not Tracking CAC by Channel or Segment

You have multiple customer acquisition channels (direct sales, self-serve, partnerships, inbound). Each has different CAC and different LTV. If you only calculate blended CAC, you miss the reality that some channels are highly efficient and others are loss-making. Model by channel, understand which channels work, and allocate budget accordingly.

Mistake 13: Assuming All Revenue Growth Comes from New Customers

This is the inverse of mistake 5. Instead of assuming growth comes from nowhere, some models assume growth comes from new customers but do not model churn or contraction. Your revenue growth is new customer growth minus churn plus expansion. All three matter. Do not ignore churn or contraction by assuming revenue only grows from new customers.

Mistake 14: Building a Model Without Version Control or Documentation

You will change assumptions as you learn more. If you do not document what changed and why, you lose the ability to understand your own model or defend it to investors. Document all assumptions. Create a change log. Tag versions. Make your model auditable.

Mistake 15: Not Running a Sensitivity Analysis

A sensitivity analysis is the best way to validate your model and to prepare for investor questions. Which variables matter most? If your model is extremely sensitive to a 1% change in churn, churn is what you need to focus on. If the model is robust to changes in most variables but very sensitive to CAC, then CAC is the key metric to drive. Run sensitivity analysis and use it to inform where to focus.

Mistake 16: Assuming Stock-Based Compensation Has No Cost

SBC is typically 8-15% of total compensation and is a real P&L expense (non-cash, but still an expense). Do not ignore it because it does not hit cash. For a \$10M payroll, SBC might add another \$1M in expenses. This affects your profitability path and your cash burn.

Mistake 17: Inconsistent Customer Definitions Across the Model

Define what a 'customer' is and use that definition consistently. Is it a company, an end-user, a contract? Does a customer who downgrades to zero seats but still has an account count as a customer? Define it upfront and use it consistently.

Mistake 18: Assuming Costs Will Scale Linearly When They Actually Have Step Functions

Some costs are variable (they scale with volume). Some are fixed (they do not scale). Some are step-function (they are fixed at one level, then jump to another level). A server hosting 100 customers costs \$5k. A server hosting 200 customers costs \$10k (one server per 100 customers). Most models treat hosting as a smooth percentage of revenue, which is fine for aggregate modeling, but be aware that it actually steps up.

Mistake 19: Not Reconciling Cash Between the Balance Sheet and the Cash Flow Statement

These should match exactly. If your cash flow statement says you have \$500k cash at the end of month 3 and your balance sheet says you have \$480k, you have an error. Find it and fix it before presenting.

Mistake 20: Presenting a Single Scenario When Investors Expect Multiple Scenarios

Show base, bull, and bear cases. Show sensitivity analysis. Show what you are uncertain about. Investors expect sophisticated models. A single upside-only scenario looks like you do not understand risks or you are trying to hide them. A multi-scenario model shows maturity.

Conclusion

Building a professional SaaS financial model is not easy. It requires precision, discipline, and deep knowledge of your business. But the time you invest in building a rigorous model pays dividends. A solid model clarifies your thinking, drives better decisions, and gives investors confidence in your ability to execute.

The financial model is not meant to predict the future perfectly. It is meant to be useful. Use it to understand your business, to identify which variables matter most, to test different strategies, and to communicate your vision to others. Update it monthly as you learn more. Let actual results drive changes to your assumptions. Use it to manage the business, not to simply report metrics.

The founders who build the best financial models tend to be the ones who raise capital on the best terms, manage their cash most effectively, and build the most successful companies. Start building. Get the fundamentals right. And iterate from there.