

# The Idea Vetting Checklist

## 42 Questions Every Founder Must Answer Before Building

Stop wasting money on development. Start here.

---

### Section 1: The Problem

Before you build anything, you need to prove the problem is real, painful, and worth solving.

#### 1. What specific problem does your product solve?

One problem. One sentence. If you can't explain it simply, you don't understand it well enough.

**Lazy:** "It helps people be more productive"

**Specific:** "It eliminates the 3 hours per week small business owners spend manually creating invoices"

#### 2. Who experiences this problem most painfully?

Forget "everyone." Who actually loses sleep over this? The more specific you get, the easier everything else becomes —your marketing, your messaging, your sales.

**Impossible to market:** A product for "busy professionals"

**Writes its own ads:** A product for "solo accountants drowning in client onboarding paperwork"

#### 3. How are they currently solving this problem?

Every problem has an existing solution. Maybe it's a competitor. Maybe it's a spreadsheet and prayer. Maybe it's hiring an assistant. Maybe it's suffering in silence. You need to know what you're replacing, because that's your real competition.

#### 4. Why is the current solution inadequate?

This is your opportunity. Is it too slow? Too expensive? Too complicated? Too embarrassing to use? The gap between what exists and what should exist is where your product lives.

#### 5. How frequently do they experience this problem?

Daily problems create habits. People will pay monthly for something they use daily. Yearly problems need to be extremely painful to justify a purchase. Frequency determines pricing power.

#### 6. What is the emotional cost of this problem?

People buy with emotion and justify with logic. Does this problem make them feel stupid? Overwhelmed? Behind their competitors? Embarrassed in front of their team? The emotional hook is often more powerful than the practical one.

#### 7. What is the financial cost of this problem?

Put a dollar amount on it. If you can show someone they're losing \$1,000/month to this problem, charging \$99/month is an obvious win. If you can't quantify the cost, you'll struggle to justify any price.

---

## **Section 2: The Market**

A real problem with no reachable market is just an interesting observation.

### **8. How many people have this problem?**

Actual research. Not guesses. Is your total addressable market 5,000 people or 5 million? This determines whether you're building a lifestyle business or a venture-scale company. Both are fine, but you need to know which game you're playing.

### **9. Can these people afford to pay for a solution?**

Having a problem and having budget are different things. College students have lots of problems. They have no money. Enterprise companies have fewer problems but large budgets. Who are you actually targeting, and what's their purchasing power?

### **10. Where do these people congregate?**

You need to find them to sell to them. What subreddits? What LinkedIn groups? What conferences? What podcasts do they listen to? What newsletters do they read? If you can't answer this, you can't market efficiently.

### **11. Are they actively searching for solutions?**

Check search volumes. Look at Google Trends. Are people typing this problem into search bars? Active searchers are easier to convert than people who don't know they have a problem yet. Educating a market is expensive.

### **12. What would make them switch from their current solution to yours?**

Switching costs are real. Learning curves, data migration, team retraining, breaking existing habits. Your product doesn't just need to be better—it needs to be enough better to justify the pain of switching.

---

## **Section 3: The Value**

If you can't articulate your value clearly, neither can your customers.

### **13. What is the dream outcome your product delivers?**

Not features. Outcomes.

**Features:** "automated email sequences"

**Outcomes:** "more sales while I sleep"

**Features:** "project management software"

**Outcomes:** "projects that finish on time without constant check-ins"

Sell the destination, not the vehicle.

### **14. How quickly do they get that outcome?**

Speed is a multiplier on value. "Get more clients" is okay. "Get your first new client within 14 days" is compelling. The faster the result, the more valuable the offer.

## **15. How likely are they to achieve that outcome with your product?**

A program that works for 90% of users is worth more than one that works for 30%. What can you do to increase their probability of success? Onboarding? Support? Templates? Guarantees?

## **16. What effort is required from them to get that outcome?**

Lower effort equals higher value. If they have to watch 40 hours of training and completely change their workflow, that's a hard sell. If they can get results in 15 minutes, that's an easy sell.

## **17. Can you articulate your value equation?**

**Value = (Dream Outcome × Likelihood of Success) ÷ (Time to Result × Effort Required)**

Your job is to maximize the top and minimize the bottom. Every product decision should move this equation in the right direction.

---

## **Section 4: The Business Model**

Ideas don't pay bills. Business models do.

### **18. How will you make money?**

Subscription? One-time purchase? Freemium? Transaction fees? Usage-based pricing? Ads? Pick a model and understand its implications. Subscriptions create predictable revenue but require ongoing value delivery. One-time purchases mean constantly finding new customers.

### **19. What will you charge?**

Actual number. Not a range. "\$47/month for the core plan, \$147/month for premium with additional features." Vague pricing means you haven't done the thinking yet.

### **20. Why will people pay that amount?**

Connect price to value. If you charge \$100 and they get \$1,000 in value, the sale makes itself. If the value isn't obviously greater than the price, you'll fight for every customer.

### **21. What is your customer acquisition cost?**

How much money to get one paying customer? Add up ads, content creation time, sales calls, free trials that don't convert—everything. Most founders drastically underestimate this number.

### **22. What is your customer lifetime value?**

How much does the average customer pay over their entire relationship with you?

**Formula:** Monthly price × average months retained = LTV

**Example:** A \$50/month product with 12-month average retention has a \$600 LTV.

### **23. Is your LTV at least 3x your CAC?**

This is the benchmark. If it costs you \$100 to acquire a customer worth \$300, you have a healthy business. If it costs \$100 to acquire a customer worth \$150, you're in trouble. The math has to math.

## **24. What are your gross margins?**

Revenue minus direct costs, divided by revenue. Software typically runs 70-85% margins. Services run 30-50%. Lower margins mean you need higher volume to build a real business.

---

## **Section 5: The Competition**

If you think you have no competition, you haven't looked hard enough.

### **25. Who are your direct competitors?**

Products that solve the same problem for the same people. Name them. Use them. Understand them deeply. Ignorance of your competition isn't a strategy.

### **26. Who are your indirect competitors?**

This includes doing nothing, hiring someone to handle it, using generic tools like spreadsheets, or any other way people currently address the problem. These are often bigger competitors than the obvious ones.

### **27. What do competitors do well?**

Be honest. Where are they winning? What do their customers love? Understanding their strengths helps you avoid competing where you'll lose.

### **28. What do competitors do poorly?**

Read their one-star reviews. What complaints come up repeatedly? Where are customers frustrated? These gaps are your opportunities.

### **29. Why will customers choose you over them?**

Not "because we're better." Specifically, what do you offer that they can't or won't? Lower price? Better experience? Specific niche focus? Faster results? You need a clear differentiator.

### **30. What's your unfair advantage?**

What do you have that's hard to replicate? Domain expertise from 10 years in the industry? Proprietary technology? Existing audience? Key relationships? Strategic partnerships? If there's no moat, you're building on borrowed time.

---

## **Section 6: The Product**

Features are easy to list. The right features are hard to choose.

### **31. What is the minimum viable product?**

The smallest version that delivers the core value proposition. Not your full vision—your starting point. Most founders include too much in "minimum." Cut deeper.

### **32. What features are essential for launch?**

List them. Then ask yourself: "Would someone pay for this without that feature?" If yes, cut the feature. Repeat until you're uncomfortable. That's your launch scope.

### **33. What does the user journey look like?**

Map every step. How do they discover you? What makes them sign up? What happens in their first session? How do they reach the "aha moment"? What brings them back? Where might they drop off?

### **34. What technical requirements do you have?**

- **Platforms:** web, iOS, Android?
- **Integrations:** what other tools must you connect with?
- **Performance:** how fast does it need to be?
- **Security:** what data are you handling and what compliance matters?

### **35. What is your realistic development timeline?**

Whatever you think, double it. Then add 50%. Software always takes longer than expected. Plan for delays so they don't destroy your runway.

### **36. What is your realistic development budget?**

Get actual quotes from actual developers. Add 30% contingency. Underfunding is the most common startup killer. Know your real numbers.

---

## **Section 7: The Reality Check**

Optimism builds companies. Delusion kills them.

### **37. Why will this succeed when similar ideas have failed?**

Others have tried something similar. They probably failed. Do you know why? What's different about your timing, approach, market, or execution?

### **38. What's the biggest risk to this business?**

Be honest. Is it that the market doesn't exist? That a big player could crush you? That the technology is too hard? That you'll run out of money? Name your biggest threat.

### **39. How will you mitigate that risk?**

Knowing the risk isn't enough. What's your plan? If market risk is highest, validate more before building. If competition risk is highest, build faster or find a defensible niche.

### **40. Why are you the right person to build this?**

Domain expertise? Technical skills? Existing relationships with target customers? Previous startup experience? Not everyone should build every idea. Why this idea for you?

### **41. Can you survive 18 months without revenue?**

Most products take longer to reach profitability than founders expect. What's your runway? Can you extend it? Do you need outside funding, and if so, when?

## **42. Are you building this because the market needs it, or because you want to build it?**

Honest answer only. Passion projects can become businesses, but only if the market agrees. Are you solving your own problem or a real market need? Markets don't care about your passion. They care about their problems.

---

## **What Happens Next**

Answering these questions is step one.

**Step two is turning your answers into action:**

1. **Validate with real customers.** Talk to at least 10 people who fit your target profile. Don't pitch. Listen.
2. **Build your documentation.** Executive summary, market analysis, product requirements, financial model. One document that captures everything.
3. **Run the numbers.** If the math doesn't work on paper, it won't work in reality.
4. **Then—and only then—find a developer.**

You'll show up prepared. You'll earn respect. You'll get better work at better rates.

And you'll have done what 90% of aspiring founders never do: the real work before the build begins.

**Your idea might change the world. Prove it here first.**

---

*The Idea Vetting Checklist — Stop wasting money on development. Start here.*