Minimum Viable Product (MVP)

Entrepreneurship & Small Business

Baton Exchange
ChatGPT, January 2026

A **Minimum Viable Product (MVP)** is a business and product-development concept that focuses on building the *simplest version of a product* that still delivers real value to customers and allows a team to learn from actual market feedback. Rather than launching a fully polished product with every possible feature, an MVP includes only the **core functionality necessary to solve a specific problem** for a defined audience.

The goal of an MVP is **learning, not perfection**. By releasing a product early, businesses can test assumptions about customer needs, pricing, usability, and demand. This helps teams avoid spending excessive time and money building features customers may not want or need.

An MVP must still be *viable*. This means it should be functional, reliable, and valuable enough that users are willing to try it and provide feedback. An MVP is not a rough idea or broken prototype; it is a thoughtful, intentionally limited product designed to answer critical questions such as:

- Do customers actually want this?
- Does it solve a real problem?
- Are they willing to use or pay for it?
- What should be improved or added next?

Common examples of MVPs include:

- A basic app with only one key feature
- A landing page that explains a product and measures sign-ups
- A manual or behind-the-scenes process that simulates automation
- A pilot program with a small group of users

The MVP approach supports **faster learning cycles**. Teams build, measure, and learn in short iterations. Customer feedback drives future development, ensuring that new features are added based on real demand rather than assumptions.

Using an MVP reduces risk. It prevents overbuilding, shortens time to market, and allows companies to pivot or refine direction early if the concept is not working. Many successful companies started with MVPs that looked far simpler than their final products.

In summary, a Minimum Viable Product is a strategic starting point—not a shortcut. It is a disciplined way to test ideas, learn quickly, and build products that are shaped by customers rather than guesswork.

ASK YOUR MENTOR

1. In your experience, how do you define the line between “minimum” and “too minimal” in an MVP?
2. What are the most common mistakes you see teams make when building their first MVP?
3. How do you decide which features are essential for learning versus nice-to-have?
4. Can you share an example where an MVP revealed a wrong assumption early—and why that mattered?
5. How do you balance speed to market with quality and credibility in an MVP?
6. What signals tell you that an MVP has done its job and it’s time to iterate or scale?
7. How do you involve customers meaningfully in MVP feedback without overreacting to every opinion?
8. When should a team *not* move forward after MVP feedback?
9. How does an MVP differ across industries (e.g., software, services, nonprofit, or physical products)?
10. How do you align leadership or stakeholders around learning goals instead of feature delivery when launching an MVP?

GO DO, Action Step Ideas

Pause and pray, “Lord, as I look through the below action steps, is there one that I should focus on first?”

1. **Define the Learning Goal**
   Write down the single most important question your MVP needs to answer. Design the MVP specifically to test that question—nothing more.
2. **Strip It to the Core**
   List all features or components you think your product needs. Cross out everything that does not directly support the MVP’s learning objective.
3. **Talk to Real Users**
   Schedule at least three conversations with potential users this month to observe how they interact with or respond to your MVP concept.
4. **Set Clear Success Signals**
   Decide in advance what outcomes will indicate success, failure, or the need to pivot (e.g., usage, willingness to pay, repeat engagement).
5. **Plan the Next Iteration**
   Based on expected MVP feedback, outline two possible next steps: one if assumptions are validated and one if they are not.