

Idea to MVP in 30 Days

A practical planner for founders and small business owners who need to move fast, stay focused, and ship something real. This guide gives you a battle-tested roadmap, a scope checklist, and a risk log — everything you need to go from raw idea to working product in one month.

FOUNDERS & SMES

30-DAY SPRINT

MVP PLANNER

Why 30 Days Works

Most MVPs fail not because the idea is wrong, but because the timeline is too long. Thirty days creates the right amount of pressure to make hard decisions quickly, cut scope ruthlessly, and stay accountable. It's long enough to build something meaningful — short enough to prevent over-engineering.

30

Days to Ship

A focused sprint beats a 6-month roadmap every time.

1

Core Problem

Solve one problem brilliantly, not ten poorly.

3

Weekly Milestones

Plan, Build, and Validate — one week each.

80%

Scope Cut

Most successful MVPs ship with 80% less than originally planned.

The 30-Day Roadmap

Four structured weeks, each with a clear objective and deliverable. Stick to the sequence — skipping ahead is the most common mistake founders make.

Week 1 – Define

Lock in the problem, the target user, and the single core value proposition. Write a one-page brief.

1

2

Week 2 – Scope

Map every feature, then cut to the essential three. Complete the scope checklist and set your build constraints.

3

Week 3 – Build

Execute with no new feature requests. Daily standups, weekly risk log review. Ship the core loop first.

4

Week 4 – Validate

Put it in front of 10 real users. Collect structured feedback. Decide: iterate, pivot, or double down.

Scope Checklist

Before writing a single line of code or hiring a designer, every item on this list must have a clear answer. If you can't answer it, you're not ready to build.

Must Answer First

- What is the one problem this solves?
- Who is the single target user?
- What does success look like in 30 days?
- What is the riskiest assumption?
- What is explicitly out of scope?

Feature Gate Questions

- Does this feature directly validate the core hypothesis?
- Can a user complete the core journey without it?
- Can it be added post-launch without major rework?
- Does removing it break the core value prop?
- Is it a "nice to have" or a "must have"?

📄 If a feature doesn't pass at least 3 of the gate questions, it goes on the backlog — not in the MVP. No exceptions.

Week 1 Deep Dive: Define the Problem

The most valuable week is week one. Founders who rush past definition spend weeks three and four building the wrong thing. Block time every day this week for structured thinking — not brainstorming, but validation.

1

Write the Problem Statement

One sentence. Name the user, the pain, and the context. No solution language allowed yet.

2

Talk to 5 Potential Users

In-person or video calls only. Ask about the problem, not the product. Listen for emotion, not politeness.

3

Map the Current Workflow

Draw how users solve this problem today. Identify the most painful, manual, or embarrassing step.

4

Write the One-Page Brief

Problem, target user, proposed solution, success metric, and key risks — one page, signed off by the founding team.

Week 2 Deep Dive: Ruthless Scoping

Scope creep is the silent MVP killer. This week, your job is to kill features — not protect them. Use the MoSCoW method as a forcing function, then apply the checklist to every item in the "Must Have" column again.

Must Have

Without this, the MVP cannot be tested. Typically 2–3 features maximum.

Should Have

Adds polish or completeness. Defer unless build time remains after core is done.

Could Have

Nice-to-have. Goes straight to the post-launch backlog without discussion.

Won't Have

Explicitly excluded this sprint. Write it down so there's no ambiguity.

Risk Log Template

Every MVP has risks. The founders who ship are the ones who name their risks explicitly, assign ownership, and review them weekly. Use this log every Monday morning — it takes 15 minutes and saves days of rework.

Risk Description	Category	Likelihood	Impact	Mitigation Action
Core assumption about user pain is wrong	Validation	Medium	High	Run 5 problem interviews by Day 5
Key dependency (API/tool) breaks or changes pricing	Technical	Low	High	Identify fallback or alternative on Day 1
Build takes longer than estimated	Execution	High	Medium	Cut to smaller scope before extending timeline
No-shows for user testing in Week 4	Validation	Medium	Medium	Recruit 15 users to secure 10 sessions
Founder bandwidth drops due to day job/ops	Resource	High	High	Pre-block build hours in calendar before Day 1

📄 Add a new row every time a risk is identified. Never delete rows — archive resolved risks with a ✓ and date so you can learn from them post-launch.

Week 3 Deep Dive: Build Without Drift

Week three is execution week. New ideas will surface — log them, don't build them. The goal is a working core loop: user can arrive, do the one main thing, and leave with value. Nothing else counts.



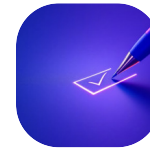
Daily Build Rhythm

Start each day with a 10-minute standup: what shipped yesterday, what ships today, what's blocked. No exceptions — even if you're a solo founder, write it down.



The No New Features Rule

Any new idea that surfaces goes into a "post-launch" list immediately. Even good ideas. Especially good ideas. You can always add — you can't un-delay a launch.



Daily Milestone Tracking

Break the week into 5 daily deliverables on Day 15. If you miss a daily milestone by more than 4 hours, escalate to a scope cut — not an overtime session.

Week 4: Validate Like a Pro

Validation is not asking friends if they like it. It's structured observation of real users attempting to complete real tasks — without your help. Follow this protocol exactly for every session.

01

Recruit the Right Users

Reach out to 15 people who match your target profile. Aim to confirm 10 sessions. Use LinkedIn, communities, or warm introductions — not family.

02

Set a Clear Task

Give users one specific task to complete. Observe without guiding. Note every moment of confusion, hesitation, or unexpected behaviour.

03

Ask the Three Key Questions

"What were you expecting?" · "What would stop you using this?" · "Who else do you know with this problem?" Listen for patterns across sessions.

04

Score Each Session

Rate task completion (Yes/Partial/No), willingness to pay (1–5), and likelihood to recommend (1–5). Aggregate across all 10 sessions before drawing conclusions.

05

Make the Call

Based on data, not gut: **iterate** (same direction, improve UX), **pivot** (same user, different problem), or **double down** (launch and scale). Document your reasoning.

Your 30-Day Launch Commitment

Print this. Sign it. Put it where you work every day. The only way this planner fails is if it stays a PDF.

Day 1 Action

Write your one-sentence problem statement and book 5 user conversations for this week. Do it before you close this document.

Day 7 Commitment

Deliver a signed one-page brief and a completed MoSCoW scope map. No brief = no build. Non-negotiable.

Day 30 Promise

Ship something real people can use. Not a landing page. Not a prototype. A working core loop in front of 10 real users.

The goal of an MVP is not to build a product. It's to answer the most important question as fast as possible — with the least amount of effort.

 START TODAY

 30-DAY SPRINT