MULTISYN TECH · 2026 EDITION

The Founder's Product Roadmap: idea to launch, one stage at a time.

The tools product teams use at each stage, explained in plain words, with a worksheet after each one. Fill it in before you spend money on development, and bring it to your first conversation with a developer.

01	02	03	04	05
PRODUCT THINKING	PRE-LAUNCH	LAUNCH	POST-LAUNCH	SUCCESS METRICS

HOW TO USE IT

Each stage has a guide page and a worksheet. Work through them in order. Tick a box only when you could show the evidence to someone else. If a stage has empty boxes or a red flag, finish it before moving on.

Fall in love with the problem, not the app.

Product thinking means starting from what users need and why, before deciding what to build. Three simple tools help you understand the problem properly.

TOOL 1

5W1H

Answer What, Who, Why, Where, When and How about the problem. If you cannot answer all six in plain words, you do not know the problem well enough yet.

TOOL 2

The Mom Test

When you interview people, ask about their life, not your idea. Ask what happened the last time the problem came up and what they did about it. Compliments are not evidence; past behaviour is.

TOOL 3

Jobs to be done

People "hire" a product to get a job done. Write it as: When [situation], I want to [motivation], so I can [outcome]. The job stays the same even when the tools change.

OUTCOME, NOT OUTPUT

Success is not "we shipped 12 features". It is a change in what users can do or how they feel. Write every feature idea as the outcome it should create, then check if it does.

> Good interview questions: "Tell me about the last time this happened." "What did you do about it?" "What did that cost you?" "What have you tried already?" Avoid: "Would you use an app that...?"

Describe the problem in their words.

Fill this in after your interviews, not before. Use the words your future users used.

WHAT IS THE PROBLEM?

WHO HAS IT MOST?

WHY DOES IT MATTER TO THEM?

WHERE DOES IT HAPPEN?

WHEN DOES IT COME UP?

HOW DO THEY SOLVE IT TODAY?

JOB TO BE DONE: WHEN ___ , I WANT TO ___ , SO I CAN ___

- ◯ I have interviewed at least 10 people who have this problem, and none of them are friends or family.
- ◯ My notes are about what they did in the past, not what they say they would do.
- ◯ I know what their current workaround costs them in time or money.
- ◯ I have picked the one job version 1 must do well.

RED FLAGS · STOP AND FIX BEFORE MOVING ON

- → "Everyone is my customer."
- → Every interview ended with "nice idea", and none with a problem story.

Know where you are going before you plan the route.

Before anything is built, write down the vision, the value you offer, and a simple roadmap. These three pages keep you, your team and your developer pointed the same way.

VISION

The future you want to create

Your vision says what changes in the world if you succeed. It stays stable for years. Your mission is what you do every day to get there.

VALUE PROPOSITION

Why someone picks you

A common, simple formula: We help [who] do [what] by [how]. One sentence, no jargon, something a customer would say back to you.

ROADMAP

Now, next, later

Group work into Now (this build), Next (after launch) and Later (ideas). Use outcomes, not dates and feature lists, for anything beyond Now.

VALUE PROPOSITION CANVAS · MATCH THEIR SIDE TO YOURS

CUSTOMER SIDE	YOUR PRODUCT SIDE
Jobs What they are trying to get done	**Products and services** What you offer for that job
Pains What annoys, blocks or costs them today	**Pain relievers** How you remove or reduce each pain
Gains Results they want or would be happy to get	**Gain creators** How you deliver those results

> Pitching to investors or partners later? Keep it concrete: the problem, who has it, what you have learned from users, and what you will build next.

Scope small, own everything.

This is where budgets are won or lost. A tight first version and accounts in your own name protect you whichever developer you work with.

VISION: IF WE SUCCEED, WHAT CHANGES?

VALUE PROPOSITION: WE HELP ___ DO ___ BY ___

MUST-HAVE FEATURES FOR VERSION 1 (FIVE AT MOST)

PARKED FOR LATER

- ◯ A clickable prototype has been tested with at least 5 target users, and I changed it based on what they did.
- ◯ The scope is written down and agreed, with milestones and a demo at each one.
- ◯ The domain, code repository, hosting, app store and payment accounts are all in my company's name.
- ◯ The contract says I own the code, and includes a bug-fix warranty after launch.
- ◯ I have decided which actions to track (sign-up, first value, payment) before the build starts.

RED FLAGS · STOP AND FIX BEFORE MOVING ON

- → The developer holds the accounts or the code.
- → A fixed price with no written scope.

Launch the smallest thing that tests your riskiest guess.

An MVP is not a cheap version of the final product. It is the fastest way to learn whether people want it. Launch early, learn from real users, then improve.

STEP 1

Find the riskiest assumption

Which belief, if wrong, would sink the idea? Usually: people have this problem, they will switch, or they will pay. Test that first.

STEP 2

Pick the cheapest test

A landing page with a sign-up, a clickable prototype, doing the service by hand for 5 customers, or a one-feature app. Build more only when the test passes.

STEP 3

Launch, then launch again

A launch is not one big day. Launch to a small group, fix what breaks, then launch again to a bigger group with a better story.

YOUR LAUNCH STORY

Who it is for, the problem it solves, what is new, and why now. If you cannot say it in four short sentences, the product or the audience is not clear yet.

> Timing matters less than founders think. A small launch to the right 50 people teaches more than a big announcement to strangers. Press and public posts come after the product works for those first users.

Launch to people who will tell you the truth.

Plan the first test and the first launch here. Success is the first 50 real users reaching value without your help.

MY RISKIEST ASSUMPTION

THE TEST I WILL RUN, AND WHAT RESULT COUNTS AS A PASS

LAUNCH DATE

WHERE MY FIRST 50 USERS COME FROM

- ◯ The product has been tested on real phones and the main browsers, not only the developer's laptop.
- ◯ A new user reaches the first moment of value without a call or a manual.
- ◯ Payments work end to end, including failed cards and refunds.
- ◯ Error tracking, uptime alerts and automatic backups are switched on.
- ◯ App store reviews are approved, with time left before launch day.
- ◯ A support channel is live and someone owns it every day.

RED FLAGS · STOP AND FIX BEFORE MOVING ON

- → Launching to everyone at once with no way to collect feedback.
- → No one on call for bugs in the first two weeks.

Measure what users do, keep the ones you win.

After launch, usage data and conversations tell you what to fix, what to build and whether to change direction.

ANALYTICS

Track actions, not page views

Record the steps that matter: signed up, reached first value, came back, paid. Then you can see exactly where people drop off.

RETENTION

Value up, effort down

People stay when the value they get keeps growing and the effort to use the product keeps shrinking. Fix friction in the first session before adding features.

PIVOT

Change one thing, keep what works

If numbers stay flat after real effort, change one part: the customer, the problem, the solution, the channel or the price. Keep what users already love.

TWO FORMULAS WORTH KNOWING

Churn rate = customers lost in the month ÷ customers at the start of the month × 100.
Retention rate = (customers at the end − new customers that month) ÷ customers at the start × 100.

> Example: you start the month with 200 customers, win 40 and lose 20, ending with 220. Churn = 20 ÷ 200 × 100 = 10%. Retention = (220 − 40) ÷ 200 × 100 = 90%.

Listen, fix, then build what the data asks for.

Run this page every week for the first three months after launch.

THE THREE THINGS USERS ASK FOR MOST

WHY USERS WHO STOPPED, STOPPED (ASK THEM)

WHERE MOST PEOPLE DROP OFF

IF NOTHING IMPROVES IN 8 WEEKS, WHAT WOULD I CHANGE?

- ◯ I read every piece of feedback each week and tag it by theme.
- ◯ Bugs that block the first moment of value are fixed before any new feature.
- ◯ I have spoken to at least 5 users who stopped using the product.
- ◯ The next 3 features are chosen by evidence, not opinion.
- ◯ Maintenance is planned and budgeted: phone and browser updates, security patches, library updates. A common rule of thumb is 15 to 25% of the build cost per year.

RED FLAGS · STOP AND FIX BEFORE MOVING ON

→ Adding features while most new users never reach the first moment of value.

Pick a few numbers that drive decisions.

Page views and downloads feel good but rarely decide anything. Metrics fall into three families. Pick one or two from each, not all twelve.

BUSINESS

Is it a business?

- Conversion rate: share of visitors or trial users who pay
- New customer growth: new paying customers per month
- Customer acquisition cost (CAC): marketing and sales spend ÷ new customers
- Customer lifetime value (LTV): average revenue per customer per month × months they stay

ENGAGEMENT

Do people use it?

- Active users: people using it each week (or day)
- Sessions and session length
- Feature usage: share of users using each key feature
- Net Promoter Score: how likely users are to recommend you, 0 to 10

EXPERIENCE

Is it easy?

- Time on task: how long a key task takes
- Error rate: how often users hit errors or dead ends
- Abandonment: share who start a task and quit
- Customer satisfaction: a quick rating after key moments

INPUT VS. OUTPUT

Output metrics (revenue, churn) tell you what happened. Input metrics (activation, weekly active users) are things your team can move this week. Pick one focus metric your whole team can influence, and review it every week.

> A healthy early sign: LTV is clearly higher than CAC, and the share of users still active after 4 weeks is stable or growing.

Your weekly scorecard.

Write a target for each number before launch, then fill in what actually happened.

METRIC	MY TARGET	WEEK 1	MONTH 1	MONTH 3
Activation New sign-ups who reach the first moment of value				
Retention Users still active 4 weeks after sign-up				
Active users People using it each week				
Conversion Active users who pay				
Churn Paying users who cancel each month				
CAC and LTV Cost to win a customer vs. what they are worth				

MY ONE FOCUS METRIC RIGHT NOW, AND WHY

- ◯ Each metric is tracked automatically, not counted by hand.
- ◯ I review these numbers on the same day every week.

Filled it in? Let's look at it together.

Multisyn Tech is a software studio that helps founders scope, build and launch MVPs, SaaS products and mobile apps. Send us your workbook and we'll go through it with you on a free 30-minute call: what to keep in version 1, what to park, and where the risks are.

BOOK A CALL

multisyn.tech/book-a-call

OR EMAIL

info@multisyn.tech

FURTHER WATCHING · FREE Y COMBINATOR TALKS, SEARCH THE TITLES ON YOUTUBE

- → How to Plan an MVP · Michael Seibel
- → How to Launch (Again and Again) · Kat Mañalac
- → Analytics for Startups · Ilya Volodarsky
- → All About Pivoting · Dalton Caldwell
- → How to Improve Conversion Rates · Kevin Hale

FURTHER READING

- → The Mom Test · Rob Fitzpatrick (book)
- → Value Proposition Design · Strategyzer (book)