GUIDE B1

TURN8

HOW TO **DESIGN YOUR MVP SPECIFICATION** IN A CORPORATE VENTURE CONTEXT

Solution design & MVP specification

WHERE THIS IS USED

- Venture Studio programs
- Corporate Incubators
- Foundry-as-a-Service engagements
- AI Studio deployments

AUDIENCE

- Venture Builders
- Product Managers
- CTOs
- Innovation Program Managers
- Founding Team Members

PHASE

Phase Two: Validation & Design → MVP Design Sprint (Weeks 4–7)

1

EXECUTIVE SUMMARY

An MVP in a corporate venture context is not a product — it is a structured learning vehicle designed to test the riskiest assumption about your business model. This guide walks product and venture teams through a disciplined process: translating customer insights into a prioritized feature set, defining clear boundaries on what the MVP will and will not do, writing a specification that engineers and investors can both read, and obtaining executive approval before a single line of code is written.

The output is a signed MVP Specification Document.

2

THE CORE PROBLEM

Why Most Teams Get This Wrong

- Corporate teams build 'polished MVPs' that include every stakeholder request — this is not a minimum viable product.
- MVP scope creep is driven by fear of presenting something unfinished to senior leadership.
- Without a clear spec, engineering teams gold-plate features or build the wrong thing entirely.
- MVPs are designed to satisfy internal audiences rather than to test external assumptions.
- Teams skip the 'minimum' and spend 6–12 months building what should take 6–10 weeks.

PREREQUISITES

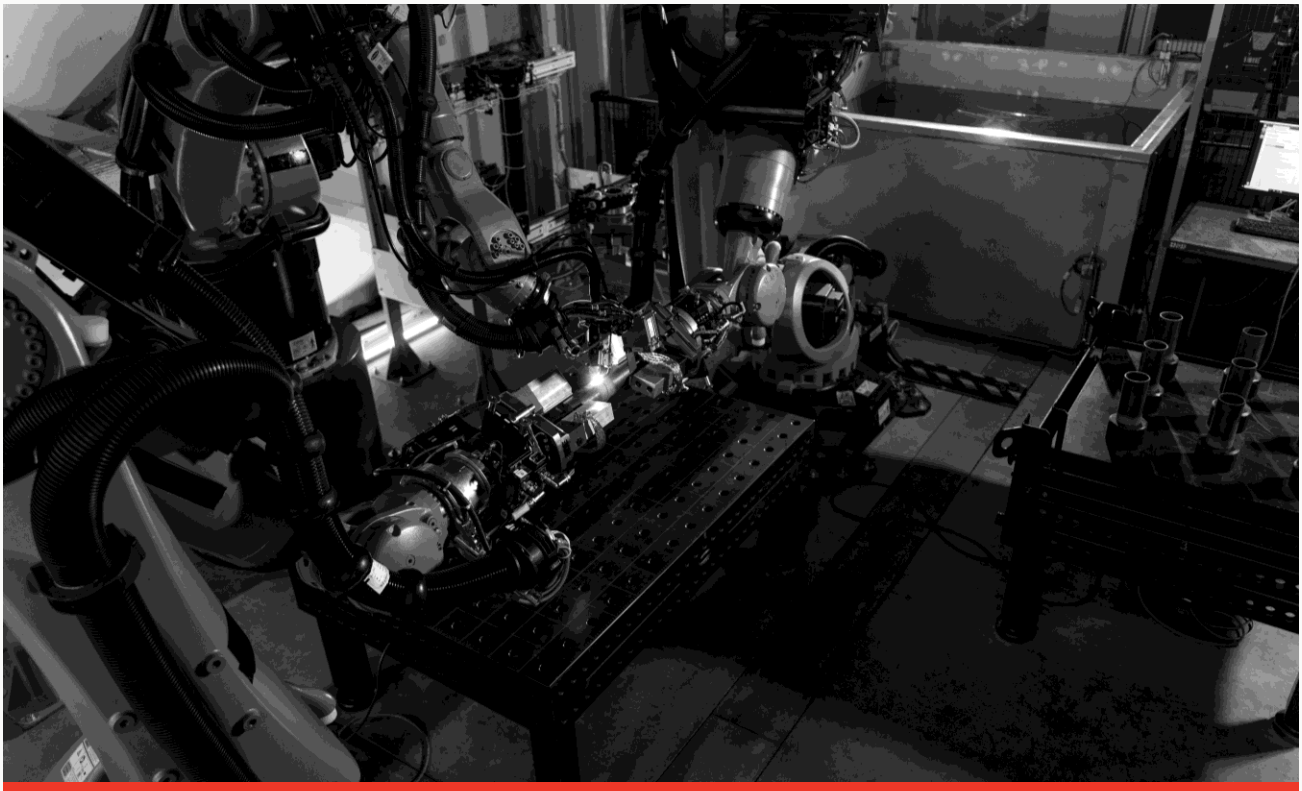
- Completed Guides A1, A2, and A3: Customer Insight Report, prioritized pain point, and recommended solution direction
- Executive sponsor committed and available for a 2-hour MVP Design Workshop
- At least 3 team members with product, technical, and business perspective
- A clear statement of the single riskiest assumption you are trying to test
- Budget and timeline authorization for the MVP build (even if rough)

4

EXPECTED OUTPUT/ SUCCESS CRITERIA

You Have Succeeded When:

- ☑ A signed MVP Specification Document exists with: Problem Statement, Core User, Core Use Case, MVP Feature Set, Explicit Out-of-Scope list, Success Metrics, Timeline, and Budget
- ☑ The team can describe the MVP in 2 minutes without mentioning any feature that is not in the spec
- ☑ The engineering or build team has reviewed the spec and confirmed it is buildable within the stated timeline
- ☑ Executive sponsor has signed off on scope and authorized the MVP build budget



STEP-BY-STEP INSTRUCTIONS

STEP 1 DEFINE THE SINGLE RISKIEST ASSUMPTION

1.1 Write this sentence: '*Our venture will fail if it turns out that ___.*' Fill in the blank with the one thing that would kill the business if it proved false.

1.2 This assumption becomes the organizing principle of your MVP. Every feature that does not test this assumption is out of scope.

1.3 Examples: *'Customers will not pay a monthly subscription for X.' / 'The API integration with Y is not possible at our price point.' / 'Enterprise procurement cycles are too long for our revenue model.'*

STEP 2 RUN THE MVP SCOPE WORKSHOP

2.1 Facilitate a 2-hour workshop with your team and executive sponsor using these 4 exercises:

- **Exercise 1 (20 min):** User Story Mapping — list every action your priority customer needs to take to solve their pain. Map them on a whiteboard from left (first action) to right (final action).
- **Exercise 2 (30 min):** Must Have / Should Have / Won't Have — for each user story, vote. Only 'Must Have' stories that directly test the riskiest assumption go into MVP scope.
- **Exercise 3 (30 min):** Minimum Success Criteria — define what a successful MVP looks like in numbers: how many users, what retention rate, what conversion rate, over what time period.
- **Exercise 4 (30 min):** Build / Buy / Configure — for each MVP feature, decide: will you build it from scratch, buy an existing tool, or configure an AI agent? This shapes your timeline and budget.

STEP 3 WRITE THE MVP SPECIFICATION DOCUMENT

3.1 Use this standard structure for the MVP Spec:

SECTION	WHAT TO WRITE
Riskiest Assumption	One sentence: what the MVP is designed to prove or disprove (Complete this row first — all other sections must support it).
Problem Statement	2 sentences: the pain, who has it, why current solutions fail
Core User	One sentence describing the specific persona the MVP serves (from Guide A3 priority segment)
Core Use Case	One paragraph: exactly what the user does with the MVP, step by step
MVP Feature Set	Numbered list of features in scope. Maximum 5–7 features. Each feature states: what it does AND what it tests.
Explicit Out-of-Scope	Numbered list of features that were requested but excluded. At minimum, this list should be as long as the in-scope feature list. State why each is excluded.
Success Metrics	3–5 measurable outcomes: define the threshold that constitutes success vs. failure
Timeline	Week-by-week milestones from start to first user test
Budget	Itemized estimate: design, engineering, tools/APIs, user testing incentives

STEP 4 USE AI TO STRESS-TEST THE SPEC

4.1 Paste your MVP Spec into your AI tool with this prompt:

AI PROMPT

```
"Review this MVP specification. Identify: (1) Features that do not
directly test the stated riskiest assumption, (2) Success metrics
that are too vague to measure, (3) Dependencies or technical risks
not mentioned in the spec, (4) Whether the timeline is realistic for
the scope, (5) What a skeptical investor would ask about this spec."
```

4.2 Address each AI flag before presenting to the executive sponsor.

STEP 5 OBTAIN EXECUTIVE SIGN-OFF

5.1 Present the MVP Spec in a 30-minute executive review. Use 5 slides: Problem, User, MVP Feature Set, Success Metrics, Timeline & Budget.

5.2 Obtain written sign-off (email confirmation is sufficient). This creates a formal record of agreed scope and protects the team from post-approval scope expansion.

TROUBLESHOOTING

ISSUE	LIKELY CAUSE	FIX
Stakeholders keep adding features to scope	Fear that MVP is 'too simple' to show externally	Reframe: 'This is not a product launch. This is an experiment. We measure it, we learn, then we build the full product.'
Engineering says timeline is impossible	Feature set is too large for MVP	Apply the '50% rule': cut the feature set in half. If something breaks, it was needed. If not, it was out of scope.
Executive sponsor won't sign off	Spec does not connect to strategic intent from Phase One	Add a slide showing exactly how the MVP tests the Phase One strategic hypothesis. Create the through-line.
Riskiest assumption is too vague to test	Assumption describes a goal rather than a testable claim	Apply the sentence test: can you measure whether the assumption is true or false within 8 weeks. If not, it is a goal, not an assumption.
MVP passes AI stress test but engineers say it is not buildable	Technical constraints or dependencies were underestimated	Return to Step 2, Exercise 4 (Build / Buy / Configure). Reduce scope or change the delivery approach.

7

VALIDATION STEPS

Can every feature in the spec be traced to the riskiest assumption?

Is the out-of-scope list as long as (or longer than) the in-scope list?

Has the engineering team confirmed the timeline is realistic?

Does the executive sponsor have a written copy of the signed MVP Spec?

8

NEXT STEPS

↘

Once your MVP Specification is signed off, proceed to Guide B2: How to Run Rapid Prototyping and Solution Validation Sessions.

CHECKLIST

FOUNDATION

☐ Riskiest assumption written in one clear sentence: Our venture will fail if it turns out that ___

☐ The riskiest assumption reviewed by the full team — everyone agrees it is the real failure mode

☐ All Section A deliverables reviewed: Customer Insight Report, priority pain point, and recommended solution direction confirmed

☐ Executive sponsor committed and available for the 2-hour MVP Scope Workshop

MVP SCOPE WORKSHOP

☐ User Story Mapping completed: every customer action from first to last mapped on a whiteboard

☐ Must Have / Should Have / Won't Have voting completed for every user story

☐ Only Must Have stories that directly test the riskiest assumption included in MVP scope

☐ Minimum Success Criteria defined in numbers: user count, retention rate, conversion rate, and time period

☐ Build / Buy / Configure decision made for each MVP feature: from scratch, existing tool, or AI agent

MVP SPECIFICATION DOCUMENT

☐ Problem Statement written: 2 sentences covering the pain, who has it, and why current solutions fail

☐ Core User defined: one sentence describing the specific persona the MVP serves

☐ Core Use Case written: one paragraph describing exactly what the user does with the MVP step by step

☐ MVP Feature Set written: maximum 5–7 features, each stating what it does AND what assumption it tests

☐ Explicit Out-of-Scope list written: at least as long as the in-scope list, with reason for each exclusion

☐ Success Metrics defined: 3–5 measurable outcomes with pass/fail threshold for each

☐ Timeline produced: week-by-week milestones from start to first user test

☐ Budget itemized: design, engineering, tools/APIs, and user testing incentives

☐ Riskiest Assumption restated at the bottom of the spec document

REVIEW & SIGN-OFF

☐ AI stress-test completed on the MVP Spec: unsupported claims, vague metrics, missing dependencies identified

☐ All AI-identified flags addressed before executive review

☐ Engineering or build team has reviewed the spec and confirmed the timeline is realistic

☐ Executive sponsor sign-off obtained in writing (email confirmation is sufficient)

☐ Signed MVP Spec distributed to all team members