

This is an example of a report you will get with Intuiva.

# LEAN Project Plan: Your Lean Project

---

## Lean Operating Brief: Multi-User Feature Validation for Productivity SaaS

### Executive Summary:

This project validates whether adding multi-user functionality to a productivity SaaS product will drive meaningful business outcomes (conversion, retention, revenue). The core risk is building a feature that bloats the product without moving key metrics. Immediate focus: quantify demand, segment users, and identify the minimum viable collaboration needed to test impact. Expected outcome: a go/no-go decision within 4 weeks, backed by data.

### --• 1 Business Thesis

Core Thesis: A subset of power users will pay more or churn less if the product enables team collaboration, but only if the implementation aligns with their specific workflows (not generic "sharing").

### Why Now?

- Existing users are signaling unmet needs (feature requests, churn comments).
- Productivity tools are shifting toward collaboration-as-a-differentiator (e.g., Notion, ClickUp).

### Revenue Mechanism:

- Upsell path: charge per additional user or tiered team plans.
- Retention lever: reduce churn from users outgrowing solo functionality.

Note: A "business thesis" isn't a guess; it's a testable claim about how this creates value. If this isn't true, the project fails.

### --• 2 Hypotheses & Assumptions

#### Critical Path Assumptions (Validate First):

1. At least 20% of active users would adopt/pay for multi-user features.
2. Collaboration needs cluster around 12 workflows (e.g., shared editing, not permissions).
3. Lack of this feature is causing measurable churn (check exit surveys).
4. Users will not tolerate significant UX complexity to gain collaboration.

#### Secondary Hypotheses:

- Teams would pay 1.52x current pricing for this functionality.
- This feature could reduce churn by 15% among power users.

Why This Matters: Every hour spent coding before validating 13 risks wasted effort. Start here.

### --• 3 Target Customer & POV

#### Ideal Customer:

- Who: Solo professionals (consultants, creators) now working with 23 contractors/clients.
- Current Workaround: Manual exports, duplicate files, or using Google Docs alongside your tool.
- Trigger to Pay: Avoids switching costs to a "team-friendly" competitor (e.g., Notion).

Note: "Teams" is too broad. Focus on users already paying you who are hitting collaboration limits.

## --• 4 Chosen Strategy & Learning Plan

Strategy: Validation First

- Why: High risk of building the wrong thing (collaboration means 100 different things).
- Key Unknowns: Demand level, use cases, willingness to pay.
- Learning Milestones:
  1. Demand quantified (survey + usage data).
  2. Use cases prioritized (interviews).
  3. MVP impact on conversion/retention (smoke test).

Why This Strategy: Until you know what "collaboration" means to users, any build is gambling.

## --• 5 Milestone Roadmap (Learning-Gated)

Phase 1: Demand Quantification (35 days)

- Goal: Prove that 20% of users care enough to change behavior.
- Actions:
  - Segment power users (top 20% by usage).
  - Survey: "Would you use/pay for [specific use case]?" (avoid vague "collaboration").
  - Analyze churn reasons for these users.
- Gate: Move forward only if 20% signal intent.

Phase 2: Use Case Deep Dive (1 week)

- Goal: Identify the 12 collaboration jobs-to-be-done.
- Actions:
  - Interview 10 power users who requested the feature.
  - Ask: "Walk me through your last 3 times you needed this."
- Deliverable: Feature prioritization (e.g., "Shared editing > permissions").

Phase 3: Smoke Test (2 weeks)

- Goal: Test if a fake "Coming Soon" feature boosts conversions.
- Actions:
  - Add a teaser to pricing page: "Team plans launching Q3."
  - Track clicks/signups from users with team-like domains.
- Gate: Proceed only if conversion lift 10%.

Phase 4: MVP Build (3 weeks, conditional)

- Goal: Launch the smallest version that tests retention impact.
- Constraints:
  - No complex permissions.
  - Support only the top use case from Phase 2.
- Avoid: Building for edge cases (admin controls, audit logs).

Likely Mistake: Assuming "multi-user" means full enterprise features. Start with shared editing only.

## --• 6 Risks & Constraints

### Critical Risks:

1. False Positive Demand: Users say they want it but won't use it.
  - Mitigation: Smoke test before building.
2. Feature Bloat: Collaboration adds UX complexity.
  - Mitigation: Isolate MVP to a separate "team mode."

### Ruthless Constraints:

- Do Not: Build permissions or admin tools until proven needed.
  - Avoid: Spending >1 week on surveys; action beats perfection.
- 7 Leverage & Unfair Advantage
- Existing Users: You have a captive audience to interview/test.
  - Usage Data: You know exactly who's hitting solo limits.
  - Speed: Competitors move slowly; you can ship a focused MVP fast.
- 8 Success Metrics

1. Conversion Lift: 10% increase in signups from team teaser.
2. Retention: 15% reduction in churn for power users post-MVP.
3. Revenue: 30% of users opt for paid team plans.

Vanity Metric to Ignore: "Number of teams created" (unless they pay).

### --• 9 First 10-Minute Action

Task: Pull last 50 support tickets mentioning "team" or "collaboration." Tag them by use case (editing, sharing, etc.).

Why: Within 10 minutes, you'll see patterns; likely 80% of requests cluster around 12 needs.

### --• 10 Strategic Navigation Questions

1. "What's the smallest version of this that would test our business thesis?"
2. "Which users would pay for this today if it existed?"
3. "What's the fastest way to disprove this idea?"

Revisit these weekly to avoid scope creep.

--•

### Tactical Stack:

1. Hotjar (track clicks on "team plan" teaser).
2. SavvyCal (schedule user interviews fast).
3. Linear (build MVP with strict feature flags).

This is a live document. Revisit Phase 1 data before proceeding.