

THE WORKSHOP

# Assumption Mapping

A pattern for surfacing the beliefs hiding underneath a plan, plotting them on an evidence/impact grid, and deciding which ones to test before you commit. Facilitator's playbook, failure modes, and what to do with the grid afterwards.

2026-05-15

[barkingiguana.com/writing/the-workshop-assumption-mapping/](https://barkingiguana.com/writing/the-workshop-assumption-mapping/)

A facilitator's playbook -- print, annotate, and keep on the table.

# Contents

Intent	3
Also Known As	3
Motivation	3
Applicability	3
Participants	4
Structure	5
Collaborations	5
Facilitator Playbook	6
Steering When It Goes Sideways	9
Consequences	10
Worked Example	10
Outputs & Follow-up	10
Related Patterns	11

---

*This is the pattern. If you want the story of Assumption Mapping in action, read [Assumption Mapping: Testing What You Believe](#) – the Greenbox team discovering that the plan they’d been confidently building was resting on three untested beliefs, one of which was about to cost them the quarter. This post is the reference you keep open the morning of.*

---

## Intent

Surface the beliefs hiding underneath a plan, plot them by how much evidence supports them and how badly the plan fails if they’re wrong, and pick a short list of assumptions to test before committing resources.

---

## Also Known As

Sometimes called the **risk/evidence grid**, the **assumptions grid**, or (informally) the **trench-coat test**. A close cousin is **hypothesis mapping** (same shape, different labels). Popularised by David Bland as part of the *Testing Business Ideas* canon, building on earlier work by Giff Constable, Tom Chi, and the broader Lean Startup community. The 2x2 layout of evidence against impact is the artefact most people mean when they say “assumptions workshop.”

---

## Motivation

A team spends six weeks building a pause-and-resume flow for subscriptions. The flow ships. Adoption is low. The team investigates and discovers that subscribers don’t want to pause – they want to skip a week. Pause is a feature the product owner imagined subscribers needed, based on a conversation with two subscribers, one of whom was actually describing a skip. The team built the wrong thing, beautifully, for six weeks.

The assumption that “subscribers want to pause” was never identified as an assumption. It was treated as a fact. Because nobody had named it as a belief, nobody thought to test it. Because nobody tested it, the whole six-week build rested on a guess that cost two weeks of user research to validate.

This is the universal shape of the failure. Every plan is a stack of beliefs. Some of the beliefs are tested and solid; some are tested and wrong; some are untested and dangerous; and some are untested and cheap to recover from. A team that can’t see the difference treats all the beliefs the same way, which means they treat the dangerous untested ones like the solid tested ones, and they find out too late.

Assumption Mapping exists to make the beliefs visible and to separate them by how much damage they do if wrong. The grid is the forcing function: you can’t pretend an untested belief is solid when you’re looking at a note in the top-left quadrant of a whiteboard everyone is standing in front of.

---

## Applicability

### Use when:

- You’re about to commit significant effort to a new product, feature, or initiative

- The team is confident and you suspect the confidence is resting on beliefs that haven't been checked
- You've just finished an Impact Map, Story Map, or Business Model Canvas and want to stress-test the underlying beliefs
- A decision feels high-stakes and you haven't separated the reversible assumptions from the irreversible ones
- An initiative has stalled and you want to know whether to continue or pivot

**Don't use when:**

- The work is small and low-risk enough that the session costs more than the work itself
- You've already validated the key assumptions through recent user research or experiments
- The team can't yet articulate what they're building – run Impact Mapping or Story Mapping first
- There's no actual decision on the table – Assumption Mapping is a pre-commitment tool, not a general awareness-raising exercise

**Inputs.** Something concrete to test assumptions about. An Impact Map, a Story Map, a Business Model Canvas, or a one-page product brief. The plan is what makes the assumptions findable – without a plan, the session produces generic worries instead of specific beliefs.

---

## Participants

**Facilitator.** Runs the clock, moderates placement debates on the grid, intervenes when "evidence" drifts into "opinion."

**Product owner or initiative lead.** Mandatory. They made most of the assumptions, consciously or not, and they'll be the one deciding which tests to fund.

**Developers.** At least one, ideally two. They'll catch the technical assumptions the business-side people don't know to question – integration feasibility, scale limits, data availability.

**Designers and researchers.** They'll catch the user-behaviour assumptions and, critically, they'll know which of the "we know subscribers want X" claims have actually been researched and which are folklore.

**Business stakeholders.** Someone who can talk about pricing, margin, market, and competitive assumptions. Without them, the grid is thin on the commercial side, which is often where the dangerous assumptions live.

**Operations / SRE.** For technical initiatives – migrations, platform rewrites, reliability projects – ops carries the assumptions about production behaviour that the feature team doesn't know. "*We assume we can cut over with no more than five minutes of downtime*" is a foundational assumption on a migration, and only the on-call engineer knows what it would actually take to test.

**Group size:** 4–6. Assumption Mapping is a debate room. Fewer than four and you lose productive disagreement; more than six and the placement arguments on the grid take longer than the session.

**Who to leave out:**

- **People who weren't involved in making the plan.** They don't hold the assumptions. Their presence produces abstract concerns instead of the specific beliefs you're trying to surface.

- **Large stakeholder groups.** If seven people need to weigh in, run a pre-session with them to agree the assumption list, then run the mapping session with the smaller group.
- **Observers.** Same rule as the other workshops: observers warp the room.

## Structure

Phase	Duration	Materials	Key question
Orient on the plan	10 min	Plan artefact visible	"What are we testing the assumptions of?"
Generate assumptions	20 min	Yellow notes, silent	"What has to be true for this plan to work?"
Share and cluster	15 min	Wall space	"Which of these are the same belief?"
Plot on the grid	25 min	2x2 grid	"How much evidence? What breaks if we're wrong?"
Prioritise testing	10 min	Dot votes or marks	"Which do we test first, and how?"
Wrap-up, owners	10 min	–	"Who owns which test, and by when?"
<b>Total</b>	<b>~90 minutes</b>		

The 2x2 grid has **evidence** on the horizontal axis (left is "no evidence, we're guessing"; right is "strong evidence, we've tested this") and **impact** on the vertical axis (bottom is "low impact if wrong"; top is "high impact if wrong, the whole plan fails"). Quadrants:

- **Top-left: Test these first** – high impact, no evidence. The dangerous ones.
- **Top-right: Monitor** – high impact, but we have evidence. Keep watching.
- **Bottom-left: Test if time allows** – low impact, no evidence. Not urgent.
- **Bottom-right: Known** – low impact, strong evidence. Stop worrying.

The top-left quadrant is the whole point of the session. Everything else is context for it.

## Collaborations

Assumption Mapping alternates between **silent generation** and **open debate**. The shape matters:

- **Generation is silent** because talking first produces groupthink. One confident voice saying "*obviously subscribers want this*" suppresses the three people who would have written assumption notes about it.
- **Sharing is round-the-room** so every person reads their notes aloud, even when several are duplicates. Duplicates are valuable – they tell you which assumptions are shared across the room and which are one person's worry.
- **Plotting is loud** on purpose. The grid placement debate is where the session earns its cost. "*That's low-impact*" / "*No it isn't, if that's wrong the whole plan dies*" is the conversation you came to have.

- **Prioritising is decisive.** The facilitator's job at the end is to force commitment: each top-left assumption gets a test, an owner, and a date, or it doesn't leave the room.

The key rhythm is **write silently, share completely, argue loudly, commit sharply.**

## Facilitator Playbook

### Phase 1 – Orient on the plan (10 minutes)

Put the plan artefact where everyone can see it. The Impact Map, the Story Map, the Canvas, or a printed one-page brief. If there's no artefact, write a one-paragraph description on a flip chart. Then read it aloud:

*"Here's the plan we're stress-testing today. Not whether the plan is right. Whether the beliefs underneath it are true. Our job is to find the assumptions this plan is standing on, plot them, and decide which ones to test before we commit further."*

Then frame the session:

*"The point isn't to debunk the plan. It's to find the parts where we've been treating beliefs as facts. By the end of ninety minutes we'll have a short list of beliefs worth testing in the next week. If the beliefs survive the tests, we commit harder. If they don't, we've saved ourselves a month of building the wrong thing."*

This matters. Teams often arrive defensive. Framing the session as *finding the beliefs* rather than *attacking the plan* gets you the surfacing you need.

#### What to watch for:

- **Defensive framing.** The product owner hears "stress-test the plan" as "attack the plan." Reframe: *"This session exists because we take this plan seriously. We wouldn't bother stress-testing a plan we didn't care about."*
- **No concrete plan.** If the artefact is actually "we want to grow the business," the session cannot run. Schedule Impact Mapping or Canvas first.

### Phase 2 – Generate assumptions (20 minutes)

Hand out sticky notes and markers. Set a timer for fifteen minutes. Give the one instruction:

*"Write silently. One assumption per note. Use the framing 'We believe that...' or 'We assume that...' – for example, 'We believe subscribers want to pause their box when they go on holiday.' Or 'We assume we can hire a second developer by June.' Don't hold back. Half-formed beliefs are exactly what we're here for. I'd rather you write thirty notes and we throw ten away than write ten and miss twenty."*

Prompt with categories if the room gets stuck:

*"User beliefs: what do we assume subscribers want, or how we assume they'll behave? Technical beliefs: what do we assume we can build, integrate with, or scale to? Business beliefs: pricing, margins, costs, churn, suppliers. Team beliefs: who we'll hire, what the team can learn, how fast we can move. Market beliefs: competitors, regulations, timing."*

Silent writing for fifteen minutes. No talking. You're looking for 15–30 assumptions from a 4–6 person room. Fewer than 15 and people are being cautious; more than 40 and you have a clustering problem in phase 3.

### What to watch for:

- **Assumptions framed as facts.** *"Subscribers want a weekly delivery."* Someone writes that as a statement of truth. Challenge at the share: *"How do we know that? Have we asked? Who? When?"*
- **Too few assumptions.** Push at the ten-minute mark: *"What about pricing? Timing? Team capacity? Competitors? Regulations? Failure modes? What assumption would embarrass us most if it turned out to be wrong?"*
- **Risks dressed as assumptions.** *"The API might be slow."* That's a risk. The assumption is *"We assume the API is fast enough for our load."* Reframe as you share.
- **Someone not writing.** They may be overthinking or stuck. Quiet prompt: *"What's the thing you're most worried about in this plan? Write that down. It counts."*
- **Deployment and reliability assumptions.** For technical plans, the silent writing should produce notes like *"We assume we can cut over in a five-minute maintenance window," "We assume our canary is sensitive enough to catch regressions," "We assume we can roll back the migration cleanly if it fails."* These are foundational and often unwritten.

### Phase 3 – Share and cluster (15 minutes)

Go round the room. Each person reads their assumptions aloud, one at a time, and places them on a blank section of the wall – not the grid yet. As notes go up, cluster similar assumptions physically together.

*"As you read yours, if one of mine feels like the same belief, say so – we'll stack them. If it's close but distinct, we keep both."*

Clustering is a light touch, not a merge. *"Subscribers will pay our headline price"* and *"Our pricing is competitive"* are related but test differently – keep both. *"Subscribers want weekly delivery"* and *"Subscribers prefer weekly over fortnightly"* are the same belief – stack them.

Remove exact duplicates. Resist the urge to rewrite notes for clarity – the exact wording often carries the specific concern that made someone write it.

### What to watch for:

- **Dismissing assumptions too quickly.** Someone says *"oh, we know that's true"* about an untested belief. Challenge: *"What evidence? If the answer is 'it's obvious,' that's not evidence."*
- **Long debates about wording.** Pick one phrasing and move on. The placement on the grid matters more than the exact text.
- **Clustering too aggressively.** If you merge too many assumptions, you lose nuance. Keep clusters small – two or three notes maximum per cluster.
- **The "we already know" trap.** The team dismisses half the assumptions as known. For each dismissed one, ask: *"If I asked the CEO the same question, would they give the same answer? What about a new team member?"* If the answer isn't confidently yes, it's not as known as it feels.

## Phase 4 – Plot on the grid (25 minutes)

Move to the 2x2 grid. Take each assumption (or cluster) and place it on the grid. For each one, the team debates:

*“How much evidence do we actually have for this belief? Not ‘it feels true’ – what concrete evidence? User research? Past experiments? Existing data? Or are we guessing?”*

*“If we’re wrong about this, what happens? Do we adjust a feature, or does the plan fall apart?”*

Place the note where the debate settles. Exact position on the grid doesn’t matter; *quadrant* matters.

This phase produces the most valuable conversations in the session. Disagreement is productive – it reveals different levels of confidence across the team. When two people disagree about whether an assumption is high or low impact, they’re disagreeing about what the plan actually is.

### What to watch for:

- **Everything in the top-left.** If every assumption lands in high-impact-no-evidence, the team is either being dramatic or the plan really is that risky. Look for assumptions that can move right with minimal testing, and look for assumptions that are actually lower-impact than they feel.
- **Nothing in the top-left.** If nothing is high-impact-untested, the team is overconfident. Challenge the top-right items: *“Is that really evidence, or is that a strong opinion?”* Push assumptions left until the team flinches.
- **Arguing about exact placement.** *“Is it at 60% or 70% on the evidence axis?”* Interrupt: *“The grid isn’t precise. Which quadrant? Pick.”*
- **Silent placement.** If people are placing notes without discussion, slow down: *“Why does that belong in the top-right? What’s our evidence? Let me hear it.”*
- **The compound assumption.** *“We assume subscribers want to pause, and that they’ll pay more for the feature, and that we can build it in two weeks.”* That’s three assumptions. Split them – each one plots differently.

## Phase 5 – Prioritise testing (10 minutes)

Focus on the top-left quadrant. These are your dangerous assumptions: high impact, low evidence. The ones that could sink the plan.

For each assumption in the top-left, briefly discuss:

*“How could we test this cheaply and quickly? Not a full build. A landing page, a prototype, a handful of interviews, a manual version of the feature. What’s the cheapest thing we could do in the next week that would tell us something?”*

*“Who owns running the test? When do we want the answer?”*

*“What result would change the plan?”*

If you have dot stickers, give each person three dots and vote on which top-left assumptions to test first. The ones with the most dots are the immediate priorities.

---

## What to watch for:

- **Tests that are really full builds.** *“We’ll test whether subscribers want it by building it.”* That’s not a test, that’s the commitment you’re trying to avoid. Push for smaller experiments: interviews, landing pages, manual concierge versions, prototypes, five-person usability studies.
- **No owner.** Every assumption in the top-left needs a person and a date by the end of the session. *“We should test this”* without an owner means it won’t happen.
- **Cherry-picking.** The team picks the interesting tests and skips the boring but important ones. Hold firm: *“The dot vote selects the order, not a different set. We work through the top-left systematically.”*
- **Tests too big to start this week.** If the proposed test is a two-month research project, it’s not an experiment, it’s another commitment. Push: *“What’s the smallest slice of that research we could run this week?”*

---

## Steering When It Goes Sideways

**The optimist.** Someone insists nothing is risky because *“it’s going to work.”* *Recovery:* Anchor to evidence: *“I’m not asking whether you believe it’ll work. I’m asking what evidence we have. Those are different questions.”* *Stop if:* They can’t engage with the evidence question. They’re not participating in the session, they’re performing confidence.

**The pessimist.** Someone puts everything in the top-left. *Recovery:* Calibrate: *“If this assumption is wrong, what specifically breaks? Does the plan fail, or do we just adjust?”* Force them to articulate the failure mode for each one. *Stop if:* The plan really is as fragile as they think. That’s a finding – escalate it rather than finishing the mapping.

**The tangent.** The team starts solving a problem they’ve found instead of finishing the map. *Recovery:* Time-box: *“Great catch. Capture the test you’d run, put it next to the note, keep plotting. We’ll prioritise solutions after we see the full grid.”* *Stop if:* The tangent reveals the whole plan is wrong. Pause the session and escalate.

**The too-many-assumptions problem.** The wall has thirty-five notes and the grid is becoming unreadable. *Recovery:* Pre-plot prioritise: dot-vote on the fifteen most important assumptions to plot. The rest go into a holding area for the next session or for asynchronous review. *Stop if:* The team can’t agree which fifteen matter most. That’s its own finding – the plan has no spine yet.

**The “we already know” trap.** The team dismisses most assumptions as known. *Recovery:* Challenge each “known” with a specific test: *“If I asked a new hire the same question tomorrow, would they give the same answer? If I asked three different customers?”* Most “known” assumptions fail this test. *Stop if:* The team won’t engage with the challenge. They’re overconfident and the session won’t persuade them; the findings will come from production.

**The political no-go assumption.** Someone writes an assumption that implicitly challenges a decision made above the team’s level. *Recovery:* Plot it honestly. Note it as “owned by leadership” and flag it for escalation rather than testing within the team. *Stop if:* Plotting the assumption will cause a political crisis the session can’t contain. Take the note privately to the product owner and handle it offline.

## Consequences

### Benefits

- Hidden beliefs made visible and explicit
- A short list of cheap experiments that de-risk the plan within a week
- Decisions to commit made with clear eyes: “we know what we don’t know”
- An artefact – the grid – that can be revisited as tests come in and assumptions move right or get invalidated
- A team that starts treating “we believe” and “we know” as different statements

### Costs

- 6–9 person-hours per session with 4–6 people
- The follow-up work of actually running the tests – without that, the session is just a wall of colourful worries
- Discomfort: the session is designed to make confident people uncertain, and that is hard on teams that reward confidence
- Recurring cost: the grid needs to be run before any significant commitment, not just once

### Failure modes

- The grid gets produced and then ignored because the team commits anyway
- Tests are scoped so large they become builds, defeating the point
- The session becomes a generic worry exercise instead of focused assumption-testing
- The team treats “we all agree this is true” as evidence, when agreement is not the same as evidence
- One person dominates placement and the grid reflects their risk appetite, not the team’s

### Stop signals

- The plan isn’t concrete enough for assumptions to attach to
- The room is performing confidence and refusing to engage with the evidence question
- The top-left quadrant is empty after twenty minutes – that’s not safety, that’s denial

Stopping and fixing the plan is not failure. Plotting assumptions about a plan that doesn’t exist is.

---

## Worked Example

See [Assumption Mapping: Testing What You Believe](#) for the Greenbox team’s first session – including the moment an assumption that felt obvious turned out to be a guess, and the one-week experiment that saved a month of wrong work.

---

## Outputs & Follow-up

### Facilitator’s close-out (same day)

- Photograph the grid with all notes placed. Make sure each note is readable and the quadrants are clear.

- Transcribe the top-left assumptions into a shared document with: the assumption, the proposed test, the owner, the due date, and the result that would change the plan.
- Send the photos and the top-left list to all participants and to whoever else needs to see it.

### The product owner's week

This is where the pattern earns its cost, and the work is mostly the product owner's. The grid is worthless without the follow-up.

- **Fund the tests.** Each top-left test needs time, possibly budget, possibly access to users. The product owner's first job is to make sure the tests actually run next week, not next month.
- **Run the tests fast.** Days, not weeks. If a test is taking more than a week, it's too elaborate – shrink it. An imperfect answer now is worth more than a perfect answer in a month.
- **Share early results.** Even preliminary findings matter. An assumption that's clearly wrong is worth knowing before the next planning session.
- **Update the grid.** As test results come in, move assumptions from left to right on the grid (evidence accumulating) or kill them entirely (invalidated). The grid is a living artefact.
- **Use the grid to gate commitments.** Before any significant hire, contract, or build decision, the product owner checks: are we betting on something in the top-left that we haven't tested yet? If yes, the commitment waits.
- **Escalate irreversible assumptions.** Some assumptions in the top-left can't be tested by the team – they depend on leadership decisions or external factors. Walk them explicitly to the people who can answer them.

### Ongoing

- Re-run the grid when the plan changes significantly. New impacts, new deliverables, new team members – each changes the assumption set.
- Keep the photographed grid visible where planning happens. It's the reminder that the team is betting on beliefs, not facts.
- Build the language into daily conversation. *"Is that a belief or a known?"* becomes a useful question in standups, reviews, and planning.

## Related Patterns

- **Impact Mapping** – every impact on an Impact Map is an assumption about actor behaviour. Run Assumption Mapping on an Impact Map and the whole middle column becomes testable.
- **Business Model Canvas** – a Canvas is nine boxes of assumptions wearing a trench coat. Assumption Mapping is the natural follow-up, especially on Revenue Streams and Cost Structure.
- **User Story Mapping** – the release-1 slice of a Story Map rests on assumptions about what users actually need. Running Assumption Mapping on the slice tells you which tasks to validate before building.
- **Wardley Mapping** – Wardley Mapping surfaces assumptions about component evolution and competitive position that Assumption Mapping can then test.

- **Threat Modelling** – Threat Modelling surfaces security assumptions (*“we assume the auth token can’t be forged”*) that belong on the grid the same way product assumptions do.

## About this playbook

This playbook is part of *The Workshop*, a reference series of facilitator playbooks published at [barkingiguana.com](https://barkingiguana.com). The canonical, up-to-date version lives at [barkingiguana.com/writing/the-workshop-assumption-mapping/](https://barkingiguana.com/writing/the-workshop-assumption-mapping/).

*These posts are LLM-aided. Backbone, original writing, and structure by Craig. Research and editing by Craig + LLM. Proof-reading by Craig.*