Innovation at the Core

A Practical Playbook for Corporate Innovation

Fabio Oliveira

Contents

Section 1: Introduction

Who This Playbook Is For

You lead innovation inside a large organisation. You might be a Head of Innovation, a VP of Product, a Design Director, or a transformation lead. Your title varies. Your problem does not.

You are responsible for making new things work inside a system designed to resist them.

You have teams running experiments. You have stakeholders asking for evidence. You have cycles that start with optimism and end with a slide deck that proves nothing. You have been handed frameworks before. Most of them sit in a shared drive somewhere, untouched after the workshop that introduced them.

This playbook is different. It is not a framework deck. It is a set of working tools, built from practice, designed to be deployed on Monday morning.

It is also for aspiring innovation leaders who want to understand what good practice looks like before they inherit a team. For L&D professionals building innovation capability. For consultants who need practical tools, not another methodology diagram.

If you have ever sat through an end-of-cycle showcase wondering why the team spent six weeks building something nobody tested, this playbook is for you.

What Is Inside

This playbook contains two things.

First, a strategic guide. The document you are reading now. It covers why corporate innovation fails, what an evidence-based practice looks like, and how seven specific tools connect to form a coherent system. It draws on twenty years of leading innovation and design inside world-leading corporations. The examples are real. The failures are real. The solutions were forged in the reality of store operations, boardroom politics, and teams that resist process.

Second, seven companion templates. Each is a standalone PDF designed for immediate use. Print them. Pin them to a wall. Fill them in with a marker. They work without the guide, but they work better with it.

The seven templates:

1. **Innovation Canvas.** An 8-step visual learning journey that becomes the backbone of every initiative.
2. **Quality Gate.** A 6-question self-assessment that stops below-the-line work from reaching decision-makers.
3. **Assumption Map.** A 2x2 matrix that forces teams to test what matters, not what is comfortable.
4. **Synthesis Ritual.** Six questions that turn experiments into learning. The team's weekly status update and learning artefact in one document.
5. **Confidence Dashboard.** A tracker across seven dimensions that makes the quality of learning visible to sponsors.
6. **Open Studio Format.** A run sheet for end-of-cycle showcases that produce decisions, not applause.
7. **Crit Session Format.** A peer feedback structure built around one question: "What would make the evidence stronger?"

How to Use This Playbook

Read the guide first. It gives you the strategic context behind every template: why it exists, what failure mode it prevents, and how it connects to the others.

Then deploy the templates with your team. Start with the Innovation Canvas (Template 1) and the Synthesis Ritual (Template 4). These two establish the learning discipline. Add the others as your team builds fluency.

Do not try to deploy all seven at once. That is the kind of process overload that makes teams abandon tools. Start with two. Add one per cycle. Within three cycles, the full system will feel natural.

> *Built from 20 years leading innovation and design inside world-leading corporations. These templates were forged in the reality of store operations, boardroom politics, and teams that resist process. Not theory. Not a framework deck. Working tools that have driven measurable outcomes.*

Section 2: The Problem

Why Corporate Innovation Fails

Large organisations are good at many things. Scaling operations. Managing risk. Executing repeatable processes. These are genuine strengths that generate billions in value.

Innovation is not one of those things.

The structures that make large organisations efficient are the same structures that make them bad at learning under uncertainty. Approval hierarchies extend time-to-learning. Performance systems that reward predictable outcomes punish experimentation. Resource allocation demands certainty before evidence exists.

The result is a pattern so common it has a name: innovation theatre.

Innovation Theatre vs. Evidence-Based Learning

Innovation theatre looks like innovation. Teams are formed. Workshops happen. Prototypes get built. Showcases fill calendars. Executive sponsors nod approvingly. Reports circulate with optimistic projections.

But nothing changes.

The prototypes were never tested with real users. The showcases presented achievements instead of evidence. The projections were built on assumptions nobody bothered to validate. Six weeks of activity produced a slide deck, not learning.

Innovation theatre survives because it satisfies the organisational need to feel innovative without requiring the discomfort of genuine uncertainty. Everyone gets to say “we’re innovating” without confronting the possibility that the idea might not work.

Evidence-based learning looks different. It starts with a hypothesis. It identifies the assumptions embedded in that hypothesis. It tests the riskiest assumptions first, not the easiest ones. It produces evidence that either builds confidence or challenges the hypothesis. And it synthesises that evidence into a clear recommendation: continue, pivot, or stop.

The difference between theatre and learning is not effort. Teams doing theatre work just as hard. The difference is discipline. Specifically, the discipline to:

- **Name your assumptions before you build.** If you cannot articulate what you believe and why, you are not innovating. You are guessing.
- **Test what is risky, not what is comfortable.** Teams naturally gravitate toward experiments that confirm what they already believe. The assumption map (Template 3) prevents this.
- **Close the loop.** Every experiment must produce a synthesis. What did you learn? What changed? What is next? Without synthesis, experiments are just activities.
- **Accept negative results.** An invalidated assumption is not a failure. It is evidence that prevents a larger failure downstream. The most valuable learning often comes from discovering what does not work.

Why Frameworks Alone Do Not Work

The corporate world is not short of innovation frameworks. Design thinking. Lean startup. Jobs to be done. Agile. Shape Up. Double diamond. Each contains genuine insight. None of them, by themselves, solve the problem.

Here is why.

Frameworks describe a way of thinking. They do not give teams something to do on Tuesday morning. A team that has been through a design thinking workshop knows they should “empathise with users.” They do not know how to extract testable assumptions from that empathy, how to prioritise which assumptions to test first, or how to synthesise findings into a confidence assessment that a sponsor can act on.

The gap between framework and practice is where most innovation programmes die. Teams understand the theory. They lack the tools to execute it under the constraints of corporate life: limited time, competing priorities, stakeholders who want answers, not questions.

This playbook bridges that gap. Each template addresses a specific point where theory breaks down in practice.

Consider “build-test-learn,” the mantra borrowed from lean startup methodology. In theory, teams build something small, test it with users, and learn from the results. In practice, “build” becomes “build a full prototype,” “test” becomes “show it to friendly colleagues,” and “learn” becomes “confirm what we already believed.”

The Innovation Canvas (Template 1) prevents this by forcing teams to articulate what they believe and what they need to learn before they build anything. The Assumption Map (Template 3) prevents comfortable testing by making risk-based prioritisation visible. The Synthesis Ritual (Template 4) prevents superficial learning by demanding specific answers to specific questions.

Frameworks are necessary. They provide shared language and mental models. But they are not sufficient. Teams need tools, not slideshows.

The Cost of Bad Innovation Process

Bad innovation process is not free. It carries real costs that organisations rarely measure.

Wasted cycles. A team that spends six weeks building a prototype without testing its core assumptions has wasted six weeks. If the prototype fails at showcase, the waste compounds. The rebuild takes another cycle. The total cost is not one cycle. It is two, plus the opportunity cost of what the team could have learned in that time.

Stakeholder trust erosion. Every showcase that presents achievements instead of evidence erodes the sponsor's confidence in the innovation programme. After two or three cycles of "we built this amazing thing" followed by "but we don't know if it works," sponsors stop investing attention. Attention is the scarcest resource in corporate innovation. Once lost, it is extraordinarily difficult to recover.

Team morale decay. Teams that work hard but produce no validated learning lose motivation. The work feels pointless because it is pointless. Talented people leave. The remaining team becomes risk-averse, choosing safe experiments that produce comfortable results. The innovation programme becomes a jobs programme, keeping people busy without creating value.

Organisational antibodies. Failed innovation programmes create institutional scar tissue. The next leader who tries to run a genuine innovation practice inherits the residual cynicism. "We tried innovation. It didn't work." What actually happened is that the organisation tried innovation theatre. But the distinction is lost on the people who lived through it.

At one large organisation, the innovation team spent three consecutive cycles building prototypes that were never tested with real users. Each showcase presented polished demos. Each sponsor meeting ended with enthusiastic approval. Nine months later, when the first prototype reached a pilot, it failed within two weeks. Store teams would not use it. The core assumption, that frontline workers would adopt a new digital tool during their busiest period, had never been tested.

The financial cost was significant. But the real damage was to the innovation programme's credibility. It took eighteen months to rebuild trust with the executive sponsor. Eighteen months of reduced funding, reduced scope, and reduced ambition.

All of it was preventable. A single round of assumption extraction and testing in cycle one would have surfaced the adoption risk. A confidence dashboard would have made the untested assumption visible to the sponsor before nine months of investment.

The tools in this playbook exist to prevent exactly this kind of failure.

Section 3: The Framework

An Operating System for Learning Under Uncertainty

Large-scale transformation is a learning problem. Every strategic initiative represents a hypothesis about how the organisation can evolve. Innovation's role is to derisk those hypotheses before the organisation commits at scale.

This requires three things from every team:

- **A clear learning practice.** Teams know what they are testing, how to design experiments, what counts as evidence, and how to synthesise findings into actionable insight.
- **Disciplined assumption extraction and prioritisation.** Teams identify assumptions, map them by risk, and prioritise ruthlessly. Otherwise they drift toward comfortable experiments rather than critical ones.
- **Honesty in confidence assessment.** Teams assess confidence against evidence, not optimism. A visible confidence dashboard makes this possible.

The framework described in this section provides the structure. The seven templates provide the tools. Together, they form an operating system that turns innovation from a set of activities into a series of evidence-based decisions.

The Innovation Narrative: 10 Chapters

Every initiative tells a story. The Innovation Narrative is the structured form of that story: a complete, evidence-based account of whether a strategic bet should be scaled, pivoted, or stopped.

The narrative has ten chapters. Seven of those chapters carry a **confidence dimension**, a lens where evidence is tracked from Assumed through to Validated or Invalidated. The chapters are the story. The dimensions are the scoreboard.

Three chapters do not carry a confidence dimension because they serve a different purpose. The **hypothesis** (Chapter 3) and the **solution design** (Chapter 4) are living artefacts that evolve in response to evidence generated in the measured dimensions. The **recommendation** (Chapter 10) synthesises all seven dimensions into a verdict.

The 10 Chapters

Framing (set first, sequentially):

#	Narrative Chapter	Confidence Dimension	What You Are Tracking
1	Strategic Context	Strategic Fit	Is this the right bet? What strategic assumption does it test?
2	What We Understand About Today	Current State	Do we deeply understand the problem space and constraints?
3	What We Believe (Hypothesis)	*Evolves*	The testable belief. Not measured directly. It sharpens as evidence accumulates.

Evidence (driven by assumption priority, not chapter order):

#	Narrative Chapter	Confidence Dimension	What You Are Tracking
4	What We Are Proposing	*Evolves*	The solution design. Not measured directly. It changes in response to evidence.
5	Desirability	Desirability	Will users and frontline teams want this?
6	Operational Feasibility	Op. Feasibility	Can this work in the existing operating system?
7	Technical Feasibility	Tech. Feasibility	Can we build and sustain this at scale?
8	Viability	Viability	Does the business case hold?
9	Adoption Readiness	Adoption	Is there a real path to handoff and scale?

Assessment:

#	Narrative Chapter	Confidence Dimension	What You Are Tracking
10	What Remains and Recommendation	*Synthesises*	The verdict. Draws on all seven dimensions to recommend: continue, pivot, or stop.

The narrative is not a waterfall. Chapters 1 through 3 are set in the first week. After that, evidence work is driven by assumption priority, not chapter sequence. A team might spend an entire cycle focused on Operational Feasibility (Chapter 6) because that is where the riskiest assumptions sit. That is correct behaviour, not a gap.

> *A team with uneven evidence across dimensions is doing it right, if they can explain why. The assumption map justifies the focus. Even, shallow evidence everywhere suggests a checklist mentality rather than testing what matters.*

The 7 Confidence Dimensions

Each dimension represents a distinct lens through which the initiative must build evidence. Together, they provide a complete picture of readiness.

Dimension	One-Line Description
Strategic Fit	Is this the right bet for the organisation's strategic direction?
Current State	Do we understand the problem space from evidence, not assumption?
Desirability	Will the people who need to use this actually want to?
Operational Feasibility	Can this function within the existing operating environment?
Technical Feasibility	Can we build and sustain the technology at scale?
Viability	Does the business case hold when tested against real data?
Adoption	Is there a credible path from pilot to full-scale deployment?

No single dimension is sufficient. An initiative with strong Desirability but untested Operational Feasibility is a prototype, not a solution. An initiative with strong Viability but Assumed Desirability is a spreadsheet, not an innovation.

The Confidence Dashboard (Template 5) tracks all seven dimensions in a single view, making the quality of learning visible to both the team and the sponsor.

The Evidence Scale

Confidence is measured by the strength of evidence, not by how the team feels. The scale is anchored to assumption coverage and evidence strength.

Level	What This Means	What It Looks Like
Assumed	Beliefs based on opinion, experience, or analogy. Critical assumptions untested.	“We believe frontline teams will adopt this because it saves time.” No frontline teams consulted.
Explored	Tested through discovery-level experiments. Evidence from what people say or models suggest. Not yet observed in context.	“We have interviewed six team leaders. Simulation suggests it works. Not yet observed in a real operating environment.”
Validated	Tested through experiments producing behavioural or operational evidence. Observed what people actually do in real conditions.	“We ran the process in three locations for a week. Completed within the target window in 85% of shifts.”
Invalidated	Experimentation discovered evidence that fundamentally compromises the hypothesis or solution.	“Frontline staff would not use the solution for more than one shift. The core workflow assumption does not hold.”

The distinction between Explored and Validated is critical. Explored means people told you it would work, or a model suggests it should. Validated means you watched it work in real conditions with real people. Many innovation programmes stall at Explored and treat it as proof. It is not.

Invalidated is not a failure state. It is the most valuable evidence an innovation team can produce. Discovering that a core assumption is wrong after two weeks of testing costs a fraction of discovering it after twelve months of development.

> *A team at Assumed across multiple dimensions after several learning cycles has a practice problem, not a progress problem.*

Learning Cycles: The Team's Heartbeat

The innovation practice operates at two scales. The **learning cycle** is the team's heartbeat: a weekly or fortnightly rhythm. The **six-week cycle** is the larger arc that accumulates learning into a coherence check and decision point.

Each learning cycle moves through five steps.

Orient

Review the current state. Update the assumption map. What has changed, internally and externally, that affects your assumptions? New competitor moves, regulatory shifts, technology changes, or internal organisational decisions can all reshape the assumption landscape.

Extract

Pull the next most critical assumptions to test. These are the highest-risk assumptions not yet adequately addressed. They may span any confidence dimension. The discipline is selecting by risk, not by convenience.

Prioritise

Rank by risk. Highest uncertainty multiplied by highest consequence comes first. The assumption map (Template 3) makes this visible. The project lead makes the final call if the team cannot quickly align. The cost of stalling exceeds the cost of testing in a slightly suboptimal order.

Experiment

Run the experiment to produce evidence. Not opinions. Not friendly feedback. Real signals that move assumptions along the evidence scale. The experiment type should match the assumption being tested.

Common experiment types include:

- **Functional prototypes** built rapidly and tested with real users in real contexts.
- **Process simulations** modelling proposed changes against real operational data.

- **Discovery interviews** with structured guides, synthesised for patterns.
- **Shadow shifts** observing and participating in current processes.
- **Concierge tests** delivering the proposed value manually to real users.
- **Multi-location comparisons** running the same test across varied contexts.

Synthesise

Answer six questions (see the Synthesis Ritual, Template 4). This is the team's primary learning artefact and their status update. The same document serves both purposes, eliminating the need for separate reporting.

> *Building to learn is the goal. But you must know what you are building to learn about. Every piece of work should be answerable to a specific assumption.*

Incremental Confidence, Not Sequential Phases

This is not a stage-gate process. There is no "discovery phase" followed by a "validation phase" followed by an "implementation phase." Understanding and building happen simultaneously. You build to understand. You prototype to test operational feasibility.

The discipline is about three things:

1. **Always knowing what assumption you are testing.** If you cannot name the assumption, the work is not connected to the learning agenda.
2. **Always prioritising by risk.** The next most important thing to learn is determined by where uncertainty and consequence are highest.
3. **Always closing the loop.** Every experiment produces a synthesis: what did we learn, what changed, what is next?

Sequential phases create a dangerous illusion of progress. A team that has "completed discovery" and moved to "validation" may have explored broadly but validated nothing. The confidence dashboard prevents this illusion by showing exactly where evidence is strong and where it is thin, regardless of which "phase" the team believes it is in.

Kill Criteria

Teams define kill criteria at initiative launch, tied to the confidence dimensions. These are agreed upfront, not decided retrospectively when sunk cost bias is already entrenched.

Kill criteria answer the question: “Under what conditions should this initiative stop?” They give the sponsor a legitimate, evidence-based mechanism to redirect resources. And they give teams permission to fail early.

Example: “If Desirability remains at Assumed after two six-week cycles, the initiative pauses for reassessment.”

Example: “If Operational Feasibility is Invalidated in any pilot location, the initiative pivots or stops before the next cycle.”

Kill criteria are revisited at each open studio (Template 6). The sponsor reviews current evidence against the pre-agreed criteria and makes a decision: continue, pivot, or stop. This is not punitive. It is responsible resource allocation based on evidence.

Without kill criteria, initiatives drift. Teams continue working on solutions that evidence suggests will not succeed, because nobody defined the conditions under which stopping is the right decision. Sunk cost bias takes over. The organisation invests further to justify past investment, not because the evidence supports it.

Trial Readiness Thresholds

Before recommending that an initiative move from innovation into a formal trial, the team must demonstrate sufficient confidence across all seven dimensions.

Dimension	Minimum Level	What This Means
Strategic Fit	Explored+	Connection to the organisation's strategic direction is clear and tested.
Current State	Explored+	The team can describe the current state from evidence, not assumption.
Desirability	Validated	Real user behaviour confirms the solution works.
Op. Feasibility	Validated	Evidence from real operating contexts confirms it can function.
Tech. Feasibility	Explored+	Critical technical assumptions have been tested.
Viability	Explored	No killers identified. Sizing based on observed data.
Adoption	Explored+	Change tactics have been prototyped. The future owner is engaged.

Any dimension still at Assumed is a clear signal the initiative is not ready for trial. A team that requests trial approval with Assumed dimensions is either overconfident or has not used the confidence dashboard honestly.

Desirability and Operational Feasibility require Validated status because these are the dimensions most likely to cause trial failure. A trial that fails because users reject the solution or because it cannot function in the operating environment wastes significant organisational resources and, critically, erodes stakeholder trust for the next initiative.

The thresholds are not bureaucratic hurdles. They are protection. Protection for the team, whose work deserves a fair test. Protection for the sponsor, whose resources deserve evidence-based allocation. Protection for the organisation, whose appetite for innovation depends on outcomes, not promises.

Section 4: The Innovation Canvas

A Visual Learning Journey in Eight Steps

Most innovation teams start with a solution. They build something, show it to someone, and call the feedback “validation.” The Innovation Canvas exists to prevent this. It forces a team to articulate what they believe, what they need to learn, and what evidence they have before they build anything.

The canvas is not a form. It is a thinking tool. The distinction matters. A form asks you to fill in boxes. A thinking tool asks you to confront what you do not know. Teams that treat the canvas as a form produce polished documents with no learning inside them. Teams that treat it as a thinking tool produce messy, honest artefacts that drive real progress.

The canvas has eight steps, arranged in three groups: Framing, Evidence, and Assessment. The layout is a journey. You move from left to right, from “what we believe” to “what we know” to “what we recommend.” By the end of a six-week cycle, the canvas becomes the backbone of every showcase and every decision.

The Eight Steps

Framing (Steps 1-3): Set Once, Sharpen Over Time

Step 1: Strategic Context. What strategic bet does this initiative test? Why now? This step connects the team’s work to the organisation’s direction. Without it, teams drift into interesting problems that nobody asked them to solve. A strong strategic context statement is two to three sentences that a board member would recognise as relevant.

Step 2: Current State. What do we understand about the problem space today, from evidence rather than opinion? This step forces the team to distinguish between what they know and what they assume. A weak current state reads like a hypothesis. A strong current state reads like a research brief, grounded in observation and data.

Step 3: Hypothesis. What do we believe will happen, and how will we know? The hypothesis follows a structured format: “We believe [who] will [action] in [context] because [reason], and we will know when [outcome].” This structure forces specificity. A vague

hypothesis like “we believe this will help customers” is untestable. A specific hypothesis like “we believe store team leaders will complete the check-in process within fifteen minutes because it replaces a manual task they already do” is testable, measurable, and falsifiable.

These three steps are set in the first week. They evolve as evidence accumulates, but they do not change daily. The framing is the anchor. Everything else moves around it.

Evidence (Steps 4-6): The Learning Engine

Step 4: Assumption Map. What assumptions are embedded in our hypothesis, and which ones carry the most risk? This step feeds directly from the Assumption Map template (see Section 6). The team plots their assumptions on a 2x2 matrix of uncertainty and consequence, then tests from the top-right corner: highest uncertainty, highest consequence.

Step 5: Solution Design. What are we proposing? This is the team’s evolving solution, captured as a storyboard or blueprint. It is explicitly labelled as evolving because the solution should change in response to evidence. A solution that looks the same at week six as it did at week one has not learned anything.

Step 6: Weekly Synthesis. What did we learn this cycle? This step captures a timeline of synthesis entries, one per learning cycle. Each entry answers six questions (see the Synthesis Ritual, Section 7). By week six, the team has four to six entries showing the learning journey unfolding over time. This timeline is the most valuable part of the canvas. It shows not just where the team ended up, but how they got there.

Assessment (Steps 7-8): The Verdict

Step 7: Confidence Dashboard. Where is our evidence strong and where is it thin? This step reflects the Confidence Dashboard template (see Section 8), tracking all seven dimensions in a single view. It makes the quality of learning visible to both the team and the sponsor.

Step 8: Recommendation. Continue, pivot, or stop. The recommendation draws on all seven confidence dimensions and the pre-agreed kill criteria. It is not a pitch. It is a verdict based on evidence.

How the Canvas Connects Everything

The canvas is the connective tissue of the entire practice. The assumption map (Step 4) drives what the team tests. The synthesis entries (Step 6) capture what they learn. The confidence dashboard (Step 7) shows where evidence is accumulating. The

recommendation (Step 8) synthesises everything into a decision.

At the end-of-cycle showcase, the canvas provides roughly 80% of the content. Steps 1 through 3 open the frame. Step 4 shows what the team chose to test and why. Step 6 walks through the learning journey. Step 7 shows the confidence picture. Step 8 closes with the recommendation.

> *The most common mistake with the canvas is treating it as a form to fill rather than a thinking tool to use. If the team is not arguing about what goes in a box, they are not using it properly.*

A team working on a frontline productivity initiative used the canvas to discover, in week two, that their core assumption about user adoption was untested. They had spent an entire previous cycle building a prototype without ever asking whether the people who needed to use it would actually adopt it during their busiest period. The canvas made this gap visible. They redirected their effort immediately and had behavioural evidence by week four.

The Canvas as a Living Document

The canvas is not completed once and filed. It is a living document that updates throughout the cycle. Steps 1 through 3 are set early and sharpen over time. Step 4 (the assumption map) changes weekly as assumptions are tested and new ones emerge. Step 5 (the solution design) evolves as evidence shapes the team's understanding. Step 6 (the synthesis timeline) grows with each learning cycle. Step 7 (the confidence dashboard) shifts as dimensions move along the evidence scale.

A canvas that looks the same at week six as it did at week one has not been used. It has been filed. The canvas should be messy by week three. Assumptions should be crossed out and replaced. The hypothesis should have annotations. The solution design should show evidence of multiple iterations. This is what learning looks like on paper.

Teams sometimes resist this messiness. They want a clean document. But a clean canvas at the end of a cycle is a warning sign. It means either the team did not learn anything new, or they rewrote the canvas to look polished rather than honest. Neither outcome serves the practice.

The best canvases are the ones you can trace through time. You can see what the team believed in week one, what evidence they gathered in weeks two through four, and how their understanding evolved by week five. This trace is the learning journey. It is the most valuable output of the entire cycle.

See Template 1 for the printable Innovation Canvas.

Section 5: The Quality Gate

Protecting Everyone's Time

The Quality Gate exists because of a pattern that repeats across every innovation programme: teams submit work that is not ready. The decision-maker catches the gaps. The work goes back for revision. The cycle repeats two or three times before the deliverable reaches an acceptable standard.

This is expensive. Not just in time, but in trust. Every round of revision erodes the sponsor's confidence that the team can deliver. Every round of revision consumes the team's energy on rework instead of learning. After two or three cycles of this, the sponsor stops engaging deeply. They skim instead of reading. They nod instead of challenging. The feedback loop degrades, and both sides lose.

The Quality Gate breaks this pattern by moving the checkpoint before submission. The team self-assesses their deliverable against six criteria. If any criterion is not met, the work does not leave the team. The gate protects the team's time by catching problems early. It protects the sponsor's time by ensuring that what arrives is ready for a real conversation.

The Six Criteria

Each criterion addresses a specific failure mode that causes revision cycles.

#	Question	What It Prevents
1	Does this deliverable have a measurable benefit hypothesis?	Work that cannot be evaluated against outcomes
2	Have you addressed all feedback from the last review?	The same feedback appearing three times
3	Is the core narrative clear: what problem, for whom, how?	Deliverables that need a verbal walkthrough to make sense
4	Have you tested this with at least one real user?	Solutions built on assumptions about user behaviour
5	Can someone outside your team understand this without explanation?	Jargon-laden work that only makes sense to the creators
6	Is this in the correct format or template?	Basic formatting problems consuming review time

How to Introduce the Gate

The gate works only if the team sees it as protection, not policing. The framing matters.

Wrong framing: “You must pass this gate before I will look at your work.” This creates resentment. The team hears: “I don’t trust you.”

Right framing: “This gate protects your time. If you submit work that isn’t ready, it comes back for revision and you lose a week. This catches the gaps before they cost you.” The team hears: “This exists so your effort counts.”

When a team answers “no” to any question, they see a specific instruction for what to fix. The gate does not just reject. It redirects. “No” to question four becomes: “Run your solution past at least one real user. Document what they said.” This is actionable. The team knows exactly what to do next.

The gate sits between the team and the decision-maker. It is not an approval process. It is a self-assessment. The team decides whether they pass. This preserves autonomy while raising the bar.

> *A gate that rejects without redirecting creates frustration. A gate that redirects without rejecting creates learning.*

See Template 2 for the printable Quality Gate checklist.

Section 6: The Assumption Map

Testing What Matters, Not What Is Comfortable

Every hypothesis contains assumptions. Some are trivial. Some are existential. The Assumption Map exists to make the difference visible and to force teams to test the assumptions that matter most, not the ones that are easiest to test.

This is harder than it sounds. Teams naturally gravitate toward comfortable experiments. They test assumptions where the evidence is likely to be positive. They interview friendly colleagues instead of sceptical users. They prototype the features they are confident about instead of the features that carry the most risk. This is human nature, not incompetence. But it produces a dangerous illusion of progress: lots of activity, lots of positive signals, and a core assumption that nobody tested sitting quietly in the background.

The Assumption Map prevents this by making risk the organising principle of all experimentation.

How It Works

The map is a 2x2 matrix with two axes.

X-axis: Uncertainty. How confident are we that this assumption is true? Low uncertainty means we have strong evidence or direct experience. High uncertainty means we are guessing.

Y-axis: Consequence. If this assumption is wrong, how much damage does it cause? Low consequence means the initiative adjusts. High consequence means the initiative fails.

The four quadrants that result:

	Low Uncertainty	High Uncertainty
High Consequence	Plan (we are fairly sure, but the stakes are high, so verify)	Test First (highest priority, test immediately)
Low Consequence	Park (low risk, low uncertainty, address later)	Monitor (uncertain but stakes are low, keep an eye on it)

The discipline is simple: test from the top-right corner first. The assumptions with the highest uncertainty and highest consequence are the ones that could kill the initiative. Test them before you invest further.

Writing Good Assumptions

A well-written assumption is specific, testable, falsifiable, and connected to a confidence dimension.

Weak assumption: “Users will like the new feature.” This is vague. What does “like” mean? Which users? Which feature? There is no way to test this rigorously.

Strong assumption: “Store team leaders in metropolitan locations will complete the daily check-in within fifteen minutes during the morning setup period.” This is specific (who, what, when, where), testable (you can measure it), falsifiable (if they take thirty minutes, the assumption fails), and connected to a confidence dimension (Operational Feasibility).

Quality	Weak Example	Strong Example
Specific	“People will use it”	“80% of targeted users will complete the task within the first week”
Testable	“The business case works”	“Unit economics are positive at 500 transactions per month”
Falsifiable	“Customers want better service”	“Customers will choose the new process over the existing one when given both options”
Connected	“This is feasible”	“The existing network infrastructure supports the data volume required”

The Top-Right First Rule

Teams resist the top-right corner. The assumptions there are the scariest. They are the ones where the team genuinely does not know the answer, and where being wrong has the biggest consequences. Testing them feels risky because a negative result could stop the initiative.

But that is exactly why they must be tested first. If the top-right assumption is wrong, every other experiment the team runs is wasted effort. A team that spends four weeks validating desirability while an untested operational feasibility assumption sits in the top-right corner is

building on sand.

> *If you are testing comfortable assumptions while uncomfortable ones wait, the assumption map is not working. It is decorating.*

One team mapped twelve assumptions for a frontline productivity tool. Nine sat in the lower-left quadrant: low risk, low uncertainty. Three sat in the top-right: high risk, high uncertainty. The team's instinct was to start with the nine easy ones to "build momentum." When forced to confront the top-right first, they discovered in week two that the core workflow assumption did not hold. The initiative pivoted immediately, saving four weeks of wasted effort.

Each assumption on the map should include a reference to which confidence dimension it belongs to. This creates a direct link between the assumption map and the confidence dashboard. When an assumption moves from Assumed to Explored or Validated, the corresponding dimension on the dashboard updates.

Updating the Map

The assumption map is not static. It changes every week.

New assumptions emerge as the team learns. An experiment that validates one assumption often surfaces two more. The team adds these to the map, positions them on the 2x2, and reassesses priority.

Tested assumptions move off the priority list. They do not disappear from the map. They move to a "tested" state with a note about the result: validated, invalidated, or inconclusive. Over time, the map becomes a record of the team's learning journey, showing what they tested, what they found, and how their understanding evolved.

External changes also affect the map. A competitor launches a similar product. A regulation changes. A technology matures faster than expected. These shifts alter the uncertainty and consequence of existing assumptions. A monthly review of the map against external context keeps it current.

The most effective teams spend ten to fifteen minutes at the start of each learning cycle reviewing and updating the map. This is time well spent. It ensures the team is always testing the most important thing, not just the next thing on last week's list.

See Template 3 for the printable Assumption Map.

Section 7: The Synthesis Ritual

Turning Experiments into Learning

Running experiments is not the hard part. Synthesising what you learned is. Teams that run experiments without synthesising them accumulate activities, not insight. They can tell you what they did. They struggle to tell you what it means.

The Synthesis Ritual exists to close this gap. It is a set of six questions that the team answers after every learning cycle. The answers become the team's primary learning artefact and their status update. The same document serves both purposes, eliminating the need for separate reporting.

> *Completeness without synthesis is busy-ness, not progress.*

The Six Questions

Each question builds on the one before it, creating a logical chain from intent to action.

1. What did we set out to learn? This connects the experiment to the assumption map. If the team cannot name the specific assumption they were testing, the experiment was not connected to the learning agenda.

2. What did we do? A factual account of the experiment. Not a justification. Not a sales pitch. What actually happened, who was involved, and how long it took.

3. What did we find? The evidence. Not the interpretation. Not what the team hoped to find. What the data, observations, or user behaviour actually showed. Separate fact from inference.

4. What changed in our confidence? This is where the evidence connects to the confidence dashboard. Did a dimension move from Assumed to Explored? From Explored to Validated? From any level to Invalidated? If nothing changed, the experiment was not designed to produce a meaningful signal.

5. What does this mean for our hypothesis? Does the hypothesis hold? Does it need refinement? Does it need to be abandoned? This question forces the team to confront the implications of their evidence.

6. What is the next most important thing to learn? This feeds directly back into the assumption map. The cycle closes. The team identifies the next highest-risk assumption and designs the next experiment.

Making It a Habit, Not a Chore

The synthesis ritual works when it becomes rhythmic. Weekly or fortnightly, at the same time, in the same format. The first two or three times feel effortful. By the fourth, the team can complete it in thirty minutes.

The key is keeping it honest. A synthesis where every answer is positive is suspicious. Real learning cycles produce mixed signals. Some assumptions hold. Some do not. Some experiments produce ambiguous results that require further testing. A team that reports only good news is not synthesising. They are presenting.

The synthesis ritual also doubles as the team's status update. There is no need for a separate weekly report, a slide deck, or a progress dashboard that someone has to maintain. The six questions tell the sponsor everything they need to know: what the team is testing, what they found, and what they plan to do next. This is more useful than a percentage-complete bar or a traffic-light status chart.

See Template 4 for the printable Synthesis Ritual.

Section 8: The Confidence Dashboard

Making the Quality of Learning Visible

The Confidence Dashboard tracks evidence across seven dimensions in a single view. It answers the question that matters most to sponsors: “How confident are we that this initiative will work, and what is that confidence based on?”

Without a dashboard, confidence is invisible. Teams report progress in terms of activities completed or prototypes built. Sponsors hear “we’ve been busy” and assume things are going well. The dashboard replaces this with a direct view of where evidence is strong, where it is thin, and where it is absent.

The Seven Dimensions in Practice

Each dimension represents a distinct question the initiative must answer.

Dimension	The Question	What Good Evidence Looks Like
Strategic Fit	Is this the right bet?	Alignment to documented strategic priorities, sponsor confirmation
Current State	Do we understand the problem?	Observation data, baseline measurements, user research findings
Desirability	Will people want this?	Behavioural evidence from real users in real contexts
Op. Feasibility	Can it work in the real environment?	Evidence from pilot locations during normal operations
Tech. Feasibility	Can we build and sustain it?	Working technical prototypes, integration tests, load tests
Viability	Does the business case hold?	Unit economics based on observed data, not projections
Adoption	Is there a path to scale?	Change tactics prototyped, future owner engaged, training tested

The Evidence Scale in Practice

For each dimension, the team marks one of four levels.

Assumed means the team has a belief but no evidence. This is the starting point for most dimensions. There is nothing wrong with Assumed at the beginning. There is something wrong with Assumed after several learning cycles.

Explored means the team has tested the assumption through discovery-level experiments. They have data from what people say, models suggest, or simulations predict. This is real progress, but it is not proof. People say many things they do not do. Models predict many things that do not happen.

Validated means the team has observed the thing working in real conditions with real people. Not in a lab. Not in a simulation. Not in a friendly demonstration. In the actual operating environment, during normal conditions, with the actual users.

Invalidated means the team has discovered evidence that fundamentally compromises the hypothesis or solution design for that dimension. This is not failure. This is the most valuable evidence an innovation team can produce.

Trial Readiness

The dashboard reveals whether an initiative is ready for trial. The thresholds are straightforward: Desirability and Operational Feasibility must reach Validated. All other dimensions must reach at least Explored. Any dimension still at Assumed is a clear signal the initiative is not ready.

A team that requests trial approval with Assumed dimensions is either overconfident or has not used the dashboard honestly. Either way, the dashboard makes this visible before resources are committed.

What the Dashboard Reveals About Practice Quality

The dashboard is not just a progress tracker. It is a diagnostic tool for the team’s learning practice.

Even, shallow evidence everywhere suggests a checklist mentality. The team is touching every dimension but going deep on none. They are spreading effort evenly instead of following the assumption map to the highest-risk areas.

Deep evidence in one dimension, Assumed everywhere else is correct if the assumption map justifies it. A team might spend an entire cycle on Operational Feasibility because that is where the existential risks sit. The dashboard should be uneven if the assumption map is uneven.

> *A team at Assumed across multiple dimensions after several learning cycles has a practice problem, not a progress problem.*

Explored everywhere but Validated nowhere suggests the team is stuck at discovery. They are collecting opinions and running models but not testing in real conditions. This is a common plateau that the dashboard makes visible.

The dashboard makes the conversation between the team and the sponsor productive. Instead of “how’s it going?” followed by a narrative answer, the conversation becomes: “I see Operational Feasibility moved to Explored this week. What did you test? What did you find? What’s next?” This is a conversation about evidence, not about feelings.

The Dashboard Over Time

The dashboard tells a different story at different points in the cycle.

Week 1. Most dimensions are at Assumed. This is expected. The team has just framed the initiative. The dashboard establishes the starting point.

Week 3. Some dimensions should have moved. If the team has completed two or three learning cycles and the dashboard still shows Assumed across the board, there is a practice problem. Either the experiments are not producing meaningful evidence, or the team is not connecting evidence to dimensions.

Week 5. The pattern should be clear. Dimensions the team focused on show Explored or Validated. Dimensions they deprioritised show Assumed or Explored. The assumption map explains why the pattern looks the way it does. If the sponsor sees a dimension at Assumed and the assumption map shows that dimension’s assumptions in the bottom-left quadrant (low risk, low uncertainty), that is fine. If a top-right assumption’s dimension is still at Assumed, that is a problem.

Week 6. The dashboard is the centrepiece of the Open Studio. It tells the sponsor everything in one view: where the team has evidence, where they do not, and what the implication is for the recommendation. A dashboard with all seven dimensions at Validated

would be extraordinary. A dashboard with a mix of Validated, Explored, and one Invalidated is more realistic and often more useful. The Invalidated dimension tells the sponsor something crucial about where the initiative cannot go.

Teams should update the dashboard weekly, even when nothing changes. The act of reviewing all seven dimensions forces the team to consider whether their learning is balanced or whether they have tunnel vision on a single dimension.

See Template 5 for the printable Confidence Dashboard.

Section 9: The Open Studio Format

A Showcase That Produces Decisions

The Open Studio is the end-of-cycle showcase. It is where the team presents their learning to the sponsor and to peer teams, and where the sponsor makes a decision: continue, pivot, or stop.

Most showcases fail. Not because the teams are bad, but because the format is wrong. Teams present achievements. They show polished prototypes. They walk through features. The audience nods. The sponsor says “great work.” Everyone leaves feeling good. Nothing was decided. No evidence was examined. No assumptions were challenged.

The Open Studio format prevents this by structuring the showcase around evidence, not achievements.

The Five-Part Structure

The showcase runs for 35 minutes. The timing is deliberate. Shorter than most meeting slots, which forces discipline. Every minute has a purpose.

Part	Duration	What Happens
1. Open with the frame	5 min	Present the strategic context, current state understanding, and hypothesis (canvas steps 1-3). Set the stage for why this work matters.
2. Show the assumption map	5 min	Walk through the assumptions identified, their position on the 2x2, and which ones were prioritised for testing this cycle.
3. Walk through learnings	15 min	The heart of the showcase. Walk through what was tested, what was found, and what changed. This is assumption-led, not feature-led. Each learning connects to a specific assumption.
4. Show the confidence dashboard	5 min	Present the current state of evidence across all seven dimensions. Highlight what moved and what did not.
5. Close with what remains and the recommendation	5 min	What assumptions remain untested? What is the team's recommendation: continue, pivot, or stop? Review against pre-agreed kill criteria.

Who Is in the Room

The Open Studio has two audiences.

The sponsor holds decision rights. They can approve continuation, mandate a pivot, or stop the initiative. The sponsor reviews the evidence against the pre-agreed kill criteria and makes a decision before leaving the room. Not "I'll think about it." Not "let's discuss offline." A decision, based on evidence, in the room.

Peer teams provide feedback. They have been through the same process. They understand the methodology. Their feedback is grounded in practice, not theory. Peer feedback at the Open Studio often surfaces blind spots that the presenting team cannot see because they are too close to the work.

Common Failures

The most common showcase failure is teams presenting achievements instead of evidence. “We built this prototype” is an achievement. “We tested this prototype with six users in three locations over two weeks, and here is what we observed” is evidence. The difference determines whether the sponsor can make a real decision.

One team spent an entire cycle building a prototype they had never tested. The showcase revealed this in front of the sponsor. The prototype was impressive. The demonstration was polished. But when the sponsor asked “what did users think?” the answer was silence. The team had built to impress, not to learn. The sponsor paused the initiative and redirected the team to test the core desirability assumption before building anything further.

Another common failure is avoiding the recommendation. Teams present evidence and then stop, leaving the sponsor to draw their own conclusion. This is abdication. The team is closest to the evidence. They should have a view on what it means. The recommendation does not need to be certain. “Based on the evidence, we recommend continuing with a pivot toward operational feasibility testing” is a recommendation. “We’re not sure” is not.

What the Sponsor Should Do

The Open Studio is not a passive viewing experience for the sponsor. They have a job to do.

Before the showcase, the sponsor should review the kill criteria agreed at the start of the cycle. They should arrive with a clear understanding of what “stop” looks like so they can assess the evidence against those criteria in real time.

During the showcase, the sponsor should ask questions about evidence quality, not about design choices. “How many users did you test with?” matters. “Have you considered making the button blue?” does not. The sponsor’s questions should focus on whether the evidence is strong enough to support the recommendation.

After the showcase, the sponsor makes a decision before leaving the room. Continue means: the initiative proceeds to the next cycle with the team’s proposed focus. Pivot means: the initiative continues but redirects toward a different assumption or approach. Stop means: the initiative ends, the team is reassigned, and the learning is captured for future reference.

The decision should be documented. The reason should be documented. Both feed into the canvas (Step 8) and become part of the institutional record. A sponsor who routinely makes undocumented decisions after showcases creates ambiguity that undermines the entire practice.

> *The purpose of a showcase is not to get applause. It is to get a decision.*

See Template 6 for the printable Open Studio Format run sheet.

Section 10: The Crit Session

Peer Learning, Not Status Reporting

The Crit Session is a fortnightly gathering where two teams present their latest learning to each other. It is not a status update. It is not a rehearsal for the showcase. It is a structured peer feedback session built around one question: “What would make the evidence stronger?”

This question does everything. It keeps the conversation generative rather than evaluative. It prevents the feedback from becoming a critique of the team’s work quality and redirects it toward improving the evidence quality. A team that hears “your evidence would be stronger if you tested in a second location” receives actionable, non-threatening input. A team that hears “your work is not good enough” receives a judgement that produces defensiveness.

The Structure

The session runs for 60 minutes with two presenting teams.

Segment	Duration	What Happens
Team A presents	15 min	Present the latest synthesis: what was tested, what was found, what changed. Assumption-led, not slide-led.
Feedback for Team A	15 min	Peer feedback guided by: “What would make the evidence stronger?”
Team B presents	15 min	Same format as Team A.
Feedback for Team B	15 min	Same feedback format.

Keeping It Generative

The Crit Session requires facilitation, at least for the first few rounds. Without it, feedback drifts toward two failure modes.

The first failure mode is politeness. Teams give vague, positive feedback because they do not want to create conflict. “Looks great, keep going” is comfortable but useless. The facilitator’s role is to push past politeness: “What specific assumption do you think is undertested? Where would you focus next if this were your initiative?”

The second failure mode is defensiveness. The presenting team interprets feedback as criticism and starts justifying their decisions. The facilitator’s role is to reframe: “This is not about whether your work is good. It is about where the evidence could be stronger. Every team has evidence gaps. Finding them now is cheaper than finding them at showcase.”

After three or four sessions, most teams find their rhythm and the facilitator can step back.

Why External Stakeholders Should Not Attend

The Crit Session is a safe space for honest peer feedback. The moment a sponsor, a senior stakeholder, or anyone with decision authority enters the room, the dynamic changes. Teams present to impress rather than to learn. Feedback becomes cautious. The session becomes a performance, not a learning exercise.

Keep the Crit Session peer-to-peer. The Open Studio (Section 9) is where sponsors and decision-makers see the work. The Crit Session is where teams strengthen it.

See Template 7 for the printable Crit Session Format.

Section 11: Putting It Together

How the Seven Templates Connect Across a Six-Week Cycle

The seven templates are not standalone documents. They form an integrated system where each template feeds the next. The Innovation Canvas is the backbone. The other six templates serve specific functions at specific moments in the cycle. Understanding when to use each one, and how they connect, is what separates a coherent innovation practice from a collection of worksheets.

The Weekly Rhythm

A six-week cycle has a natural shape. The first week is framing. The middle weeks are learning. The final week is assessment. Here is how the templates map to that shape.

Week 1: Framing

The cycle begins with the Innovation Canvas. The team completes Steps 1 through 3: Strategic Context, Current State, and Hypothesis. These three steps set the anchor for everything that follows.

In parallel, the team creates the initial Assumption Map. They extract every assumption embedded in their hypothesis, position each one on the 2x2 matrix, and identify the top-right assumptions that must be tested first.

Kill criteria are defined with the sponsor at the beginning of the cycle. Under what conditions should this initiative stop? These criteria are tied to specific confidence dimensions and documented in the canvas.

By the end of Week 1, the team has a framed hypothesis, a prioritised set of assumptions, and a clear plan for what to test first.

Active templates: Innovation Canvas (Steps 1-3), Assumption Map

Weeks 2-4: Learning Cycles

This is where the work happens. Each week (or fortnight, depending on the team's cadence) follows the Orient-Extract-Prioritise-Experiment-Synthesise rhythm described in Section 3.

The Assumption Map is updated as assumptions are tested and new ones emerge. The Synthesis Ritual is completed after each learning cycle, producing a dated entry that captures what was tested, what was found, and what changed.

The Innovation Canvas evolves. The Solution Design (Step 5) changes in response to evidence. The Synthesis entries (Step 6) accumulate. The Confidence Dashboard (Step 7) begins to show movement as dimensions shift from Assumed to Explored.

Fortnightly Crit Sessions occur during this period. Two teams share their latest synthesis and receive peer feedback guided by the question: "What would make the evidence stronger?"

The Quality Gate operates whenever a deliverable needs to go to the sponsor. Before any work product leaves the team, they self-assess against the six criteria. If any criterion is not met, the work stays with the team until it is ready.

Active templates: Assumption Map (updated), Synthesis Ritual (weekly), Innovation Canvas (evolving), Quality Gate (as needed), Crit Session Format (fortnightly)

Week 5: Confidence Review

The team conducts a thorough review of the Confidence Dashboard. Where is evidence strong? Where is it thin? Where is it absent?

This review feeds the final Crit Session of the cycle. Peer teams help identify blind spots before the showcase. The presenting team has one more week to address gaps.

The team also begins preparing the Open Studio, structuring their showcase around the canvas: framing (Steps 1-3), assumption map (Step 4), learnings (Step 6), confidence dashboard (Step 7), and recommendation (Step 8).

Active templates: Confidence Dashboard (review), Crit Session Format, Innovation Canvas (preparation)

Week 6: Open Studio

The cycle culminates in the Open Studio. The team presents their learning journey to the sponsor and peer teams using the five-part structure. The sponsor reviews evidence against kill criteria and makes a decision: continue, pivot, or stop.

The canvas is the presentation. Roughly 80% of the showcase content comes directly from the canvas steps. The remaining 20% is the team's narrative connecting the evidence into a coherent story.

After the showcase, the team updates the Innovation Canvas with the recommendation (Step 8) and any notes from the sponsor's decision. If the initiative continues, the updated canvas and assumption map become the starting point for the next cycle.

Active templates: Open Studio Format, Innovation Canvas (Step 8), Confidence Dashboard (presented)

The Cycle at a Glance

Week	Primary Activity	Active Templates
1	Framing and assumption extraction	Canvas (Steps 1-3), Assumption Map
2	First learning cycle, first experiments	Assumption Map, Synthesis Ritual, Quality Gate
3	Second learning cycle, first Crit Session	Assumption Map, Synthesis Ritual, Crit Session
4	Third learning cycle, evidence deepening	Assumption Map, Synthesis Ritual, Quality Gate
5	Confidence review, final Crit Session, showcase prep	Confidence Dashboard, Crit Session, Canvas
6	Open Studio, sponsor decision, cycle close	Open Studio Format, Canvas (Step 8), Dashboard

How the Templates Feed Each Other

The connections between templates create a self-reinforcing system.

The **Hypothesis** (Canvas Step 3) generates **assumptions** (Assumption Map). Testing assumptions generates **evidence** (Synthesis Ritual). Evidence updates the **confidence levels** (Confidence Dashboard). Confidence gaps drive the **next experiment** (back to Assumption Map). The **Quality Gate** ensures that any work product meets a minimum standard before it reaches the sponsor. The **Crit Session** strengthens evidence through peer feedback. The **Open Studio** synthesises everything into a decision.

This is not a linear flow. It is a cycle within a cycle. The weekly learning loop (Orient, Extract, Prioritise, Experiment, Synthesise) repeats inside the six-week arc. Each repetition adds evidence. Each repetition updates the assumption map. Each repetition moves dimensions on the confidence dashboard.

The system works because every template has a clear purpose and a clear connection to the others. Remove one, and a gap opens. Remove the Synthesis Ritual, and experiments accumulate without learning. Remove the Assumption Map, and teams test comfortable things. Remove the Confidence Dashboard, and sponsors cannot see where evidence is thin. Remove the Quality Gate, and below-the-line work reaches decision-makers. Remove the Crit Session, and teams miss blind spots. Remove the Open Studio, and showcases produce applause instead of decisions.

Deploying the System Incrementally

Do not introduce all seven templates at once. That is the kind of process overload that makes teams abandon tools entirely.

Cycle 1: Start with the foundation. Introduce the Innovation Canvas and the Synthesis Ritual. These two establish the core disciplines: framing the learning journey and closing the loop after each experiment. The canvas gives the team structure. The synthesis ritual gives them rhythm. These two templates alone will improve the team's practice significantly.

Cycle 2: Add prioritisation and quality. Introduce the Assumption Map and the Quality Gate. The assumption map formalises what the team has been doing informally (choosing what to test next) and makes risk-based prioritisation explicit. The Quality Gate catches below-the-line work before it reaches the sponsor and begins breaking the revision cycle.

Cycle 3: Add visibility and feedback. Introduce the Confidence Dashboard, the Crit Session Format, and the Open Studio Format. By now the team has two cycles of practice. They understand the framework. They have generated evidence. The dashboard makes that evidence visible to sponsors. The Crit Session provides peer feedback. The Open Studio structures the end-of-cycle decision.

By the third cycle, the full system is in place. Each template was introduced when the team was ready for it. The cognitive load was distributed across cycles rather than concentrated in week one.

Teams that try to deploy everything at once typically abandon the most unfamiliar tools first. The synthesis ritual goes. Then the assumption map. Within two cycles, they are back to building prototypes and hoping for the best. Incremental deployment prevents this.

The full system, deployed incrementally over two to three cycles, transforms an innovation programme from a set of activities into an evidence-based learning practice.

Section 12: Common Failure Modes

What Goes Wrong and How to Fix It

Even with good tools, innovation teams fail in predictable ways. These failure modes are not random. They are patterns that repeat across organisations, industries, and team sizes. Recognising them early is the difference between a team that learns from mistakes and a team that repeats them.

The five failure modes described here are drawn from real experience across multiple organisations. The names and details have been changed. The patterns are universal.

1. The Build-Test-Learn Trap

What it looks like. The team adopts "build-test-learn" as their mantra. They interpret "build" as "build a full prototype." They interpret "test" as "show it to colleagues." They interpret "learn" as "confirm what we already believed." The result is a polished prototype that has never been tested with a real user in a real context.

Why it happens. "Build-test-learn" is borrowed from lean startup methodology, but it loses critical nuance when transplanted into corporate environments. In a startup, "build" means the smallest possible thing that tests a hypothesis. In a corporate team, "build" defaults to the team's comfort zone, which is usually a medium-fidelity prototype that demonstrates a feature rather than testing an assumption.

The team conflates building with learning. They feel productive because they are making something. The prototype looks good. Stakeholders are impressed. But the prototype was never connected to a specific assumption. It was not designed to produce evidence. It was designed to demonstrate capability.

Which template prevents it. The **Innovation Canvas** (Template 1) prevents this by requiring the team to articulate their hypothesis and extract assumptions before they build anything. The canvas forces the question: "What are you building this to learn?" If the answer is vague, the build is not connected to the learning agenda. The **Assumption Map** (Template 3) then ensures the team prioritises what to test by risk, not by what is most natural to build.

How to spot it early. The team produces impressive artefacts but cannot answer the question: “Which assumption did this test?” They describe what they built rather than what they learned. Their synthesis entries (if they write them) answer question two (“What did we do?”) in great detail but answer question three (“What did we find?”) in vague terms.

2. The Comfortable Experiment Trap

What it looks like. The team runs experiments, but they test assumptions where the evidence is likely to be positive. They interview users who are known to be enthusiastic. They test in locations where conditions are favourable. They measure metrics where improvement is almost guaranteed. Meanwhile, the critical assumptions, the ones where being wrong would kill the initiative, sit untested.

Why it happens. Testing risky assumptions is frightening. A negative result could stop the initiative. Teams that have invested weeks of effort do not want to discover that their core assumption is wrong. So they avoid the question. They test around the edges, accumulating positive evidence on low-risk assumptions while the existential risk remains unexamined.

This is not dishonesty. It is human nature. People seek confirmation. They avoid discomfort. Without a structural mechanism that makes risk-based prioritisation visible, teams will default to comfort.

Which template prevents it. The **Assumption Map** (Template 3) makes this pattern visible. When assumptions are plotted on the 2x2 matrix, it becomes obvious if the team is testing bottom-left assumptions (low risk, low uncertainty) while top-right assumptions (high risk, high uncertainty) remain untouched. The map creates accountability: the team cannot claim progress while the most consequential assumptions sit unaddressed.

How to spot it early. Look at the team’s experiment log. If every experiment produced a positive result, something is wrong. Real innovation testing produces mixed signals. Consistent positive results indicate the team is confirming what they already know, not testing what they do not know. Ask the team to point to the top-right assumption on their map and describe the experiment that tested it. If they cannot, the comfortable experiment trap is active.

3. The Sunk Cost Showcase

What it looks like. The team has spent six weeks on an initiative. Evidence is mixed. Some assumptions have been validated. Others have been invalidated. The honest recommendation would be to pivot or stop. But the team has invested significant effort. The

sponsor has invested attention. Neither side wants to acknowledge that the initiative may not work.

So the team presents achievements instead of evidence. The showcase becomes a highlight reel. The polished prototype gets a standing demonstration. The positive user quotes are front and centre. The invalidated assumptions are mentioned in passing or omitted entirely. The sponsor, seeing polished work, approves continuation. Another cycle begins with the same unresolved problems.

Why it happens. Sunk cost bias is one of the most powerful cognitive biases in organisational decision-making. The more an organisation has invested in something, the harder it is to stop. This is amplified in innovation programmes where the team's identity becomes tied to the initiative. Stopping feels like personal failure, not rational resource allocation.

The showcase format compounds the problem. Traditional showcases are structured around demonstrations. "Look what we built." This framing rewards achievement over evidence. Teams optimise for what gets rewarded.

Which template prevents it. The **Open Studio Format** (Template 6) prevents this by structuring the showcase around evidence, not achievements. The five-part structure requires the team to show the assumption map, walk through learnings assumption by assumption, and present the confidence dashboard. There is no segment for "look what we built." There is only "here is what we tested, here is what we found, here is what we recommend."

The **Confidence Dashboard** (Template 5) also plays a role. When the sponsor can see that two dimensions are Invalidated and three are still at Assumed, the evidence speaks for itself. The team cannot hide behind a polished prototype when the dashboard shows the reality.

Pre-agreed **kill criteria** provide the final mechanism. When the conditions for stopping are defined at the beginning of the cycle, the decision is not emotional. It is a comparison of evidence against criteria. "We agreed that if Desirability remained at Assumed after two cycles, we would pause. Desirability is still at Assumed. The criteria apply."

How to spot it early. The team's showcase preparation focuses on demonstrations rather than evidence. They rehearse the prototype walkthrough but have not reviewed the confidence dashboard. They can describe what they built in detail but cannot articulate what they learned. The language shifts from "we discovered" to "we achieved." Achievement language at a showcase is a reliable signal that sunk cost bias has taken hold.

4. The Feedback Amnesia Loop

What it looks like. The team receives feedback from a review or a Crit Session. They nod, take notes, and go back to work. The next time they submit, the same issues appear. The reviewer gives the same feedback. The cycle repeats. After two or three rounds, the reviewer disengages. The team interprets the silence as approval and submits the same quality of work to the sponsor.

Why it happens. Teams do not ignore feedback deliberately. They absorb it in the moment and then return to their working context, where the pressure to produce output overwhelms the discipline to address feedback. The feedback lives in meeting notes that nobody rereads. It was never converted into a specific action or a change in the artefact.

In some cases, the team disagrees with the feedback but does not say so. They comply superficially, making minor adjustments that do not address the substance. The reviewer sees the minor changes and assumes the deeper feedback was also addressed.

Which template prevents it. The **Quality Gate** (Template 2) breaks this loop with question two: “Have you addressed all feedback from the last review?” This is a binary question. Yes or no. If the answer is no, the work does not leave the team. The gate converts feedback from a thing that was said into a thing that must be addressed before submission.

The **Synthesis Ritual** (Template 4) also helps, specifically through question five: “What does this mean for our hypothesis?” When feedback from a Crit Session reveals a weakness in the evidence, the synthesis forces the team to document what changed. The feedback becomes part of the learning record, not a note in a forgotten document.

How to spot it early. The same feedback appears in consecutive reviews. The reviewer feels like they are repeating themselves. The team submits work that addresses surface-level comments but ignores the structural feedback. When asked “did you address the feedback from the last session?” the team says yes, but the reviewer recognises the same gaps. A quick audit of the previous three reviews against the current deliverable will confirm the pattern.

5. The Missing Hypothesis

What it looks like. The team does excellent research. They conduct user interviews. They observe workflows. They map pain points. They build detailed personas. The research is thorough, well-documented, and genuinely insightful.

But there is no hypothesis. The team has a rich understanding of the problem space and no testable belief about how to solve it. When asked “what do you believe will work and why?” the answer is a summary of the research, not a specific, falsifiable prediction.

The consequence is that the team cannot design experiments. Without a hypothesis, there is nothing to test. The team defaults to building something that “addresses the research findings” rather than testing a specific bet about what will work. They produce beautiful prototypes with no connection to business outcomes. The sponsor sees impressive work that cannot be evaluated because there is no clear success criterion.

Why it happens. Research is comfortable. It feels rigorous. It produces thick, detailed documents that look like progress. The transition from “we understand the problem” to “we believe this specific thing will work” requires a leap that many teams find uncomfortable. Stating a hypothesis means staking a claim. It means being specific enough to be wrong. Research-oriented teams prefer to keep gathering data rather than commit to a position.

This is also a structural problem. Many innovation methodologies emphasise the “empathise” or “discover” phase without making clear that discovery must culminate in a testable belief. Teams get stuck in permanent discovery because nobody defined when discovery ends and experimentation begins.

Which template prevents it. The **Innovation Canvas** (Template 1) prevents this through Step 3: the hypothesis. The structured format, “We believe [who] will [action] in [context] because [reason], and we will know when [outcome],” forces specificity. The team cannot leave Step 3 with a research summary. They must articulate a testable belief.

The canvas also creates a structural transition. Steps 1 and 2 are about understanding (Strategic Context and Current State). Step 3 is the bridge to action (Hypothesis). Step 4 is about testing (Assumption Map). The canvas architecture makes the transition from research to experimentation explicit and unavoidable.

How to spot it early. The team produces extensive research documentation but cannot complete the hypothesis template. They describe the problem space in rich detail but struggle to state what they believe will work. When asked “what is your hypothesis?” they answer with a research summary rather than a testable prediction. Their assumption map is empty because there is no hypothesis to extract assumptions from.

Recognising the Patterns

These five failure modes rarely appear in isolation. The Build-Test-Learn Trap often co-occurs with the Missing Hypothesis. A team that has no clear hypothesis defaults to building because building feels productive. The Comfortable Experiment Trap often leads to the Sunk

Cost Showcase. A team that tests easy assumptions accumulates positive evidence that creates false confidence, which then becomes difficult to abandon at showcase time.

The templates work as a system precisely because the failure modes are interconnected. The Innovation Canvas catches the Missing Hypothesis early. The Assumption Map prevents Comfortable Experiments. The Synthesis Ritual prevents Feedback Amnesia. The Open Studio Format prevents the Sunk Cost Showcase. The Quality Gate provides a checkpoint that catches multiple failure modes before work reaches the sponsor.

The most important thing a leader can do is name these patterns when they appear. “I think we might be in the Comfortable Experiment Trap. Let’s look at the assumption map. Where are the top-right assumptions, and when are we testing them?” Naming the pattern removes the personal dimension. The team is not being criticised. A recognisable pattern is being identified and addressed.

> *The tools in this playbook do not guarantee success. No tool can. What they guarantee is that failure happens early, cheaply, and with learning attached. An innovation team that fails at week two with a clear understanding of why has produced more value than a team that fails at month nine with a polished prototype and no evidence.*