

Creating assumptions from a problem, opportunity, or pre-defined solution.

For the moments before a solution is settled, and the moments after one is already on the table. Use this to understand the current state, or to risk-scan a solution before it is tested.

PURPOSE

Initiatives start in two places. The practice is different.

Use this guide when an initiative is still uncertain. Before investing further, understand what must be true. The work depends on where you are starting from.

PATH A

NO DEFINED SOLUTION

Problem or opportunity

An observed pain point, performance gap, customer or team member problem, suspected inefficiency, or strategic opportunity.

PRIMARY QUESTION

What do we believe is true about the current state, and what evidence do we have?

FOCUS ON

- Who is affected
- What is happening today
- Which processes, systems, and behaviours are involved
- Whether the problem is worth solving

PATH B

SOLUTION ALREADY PROPOSED

Pre-defined solution

A proposed tool, a vendor solution, a platform feature, a process change, or a pilot someone wants to run.

PRIMARY QUESTION

What must be true about the current state for this solution test to be meaningful?

FOCUS ON

- What problem the solution assumes exists
- What cause the solution assumes
- What operational, system, and behavioural conditions must be true
- Which assumptions could invalidate the test

READ ONCE. KEEP CLOSE.

Do not jump to solution testing blindly. Do not over-discover when a solution is already defined.

Use the lightest level of current-state discovery needed to avoid testing the wrong thing.

DECISION RULE

If a current-state assumption being wrong would make the solution test **misleading, invalid, unsafe, or wasteful**, test it first or deliberately build it into the test design.

PATH A · DIAGNOSE

PATH B · RISK-SCAN

SHARED · PRIORITISE

SHARED · SYNTHESIZE

SHARED · DECIDE

PATH A

Starting with a problem or opportunity.

Five steps. The goal is to understand the **current state** before solutioning. Diagnose carefully, define the problem honestly, and only then decide what to solve.

- 01 Frame the problem or opportunity
- 02 Create a current-state hypothesis
- 03 Map the current state
- 04 Generate current-state assumptions
- 05 Write in a testable format

STEP 1

Frame the problem or opportunity.

Start with a clear, solution-neutral statement so the team can investigate before deciding what to build.

We have observed **[issue / opportunity]** affecting **[people / process / customer / store / team / system]**, which may be causing **[impact]**.

EXAMPLE

We have observed that store teams spend significant time manually reviewing online order exceptions, which may be causing delays, inconsistent prioritisation, and increased customer escalations.

AVOID (SOLUTION LANGUAGE)

Stores need an automated dashboard.

,

WRITE (NEUTRAL FRAMING)

Stores may lack a clear, consistent way to identify which online order exceptions need attention first.

STEP 2

Create a current-state hypothesis.

We believe **[current-state condition]** is causing **[impact]** because **[reason]**.

STEP 3

Map the current state.

Before creating assumptions, make the current state visible. Pick the artefacts that match the question you are trying to answer.

Current-state journey

What customers, team members, or users experience.

Process map

What happens step by step today.

Service blueprint

Frontstage, backstage, systems, handoffs, support.

Stakeholder map

Who is involved, affected, influential, or dependent.

Systems map

Systems, data flows, tools, reports, integrations, workarounds.

Pain point map

Friction, delay, rework, waste, risk, confusion.

Evidence map

What we know, what we think, what we still need to learn.

FOR RETAIL OPERATIONAL INNOVATION, INCLUDE

Store routines

Support office processes

Customer impacts

System dependencies

Manual workarounds

Policy constraints

Labour impacts

Seasonal / volume variation

Exceptions

Escalation paths

Reporting and measurement

Cross-team ownership

STEP 4

Generate current-state assumptions.

Ask: *what are we assuming is true about the current state?*

Write assumptions as statements, not questions.

QUESTION

Are stores wasting time?

›

ASSUMPTION

Store teams spend significant time manually identifying which exceptions to resolve first.

QUESTION

Is the system data poor?

›

ASSUMPTION

Exception data is incomplete or unreliable enough to affect prioritisation decisions.

QUESTION

Do customers care?

›

ASSUMPTION

Customers are negatively affected when exception resolution is delayed beyond the same trading day.

A USEFUL CURRENT-STATE ASSUMPTION IS

Specific to a stakeholder, process, system, or behaviour. Testable.
Connected to the current-state hypothesis. Important enough that being wrong would change the problem definition.

DIAGNOSTIC LENSES

Five lenses for understanding the current state.

These are not solution-validation dimensions. They are diagnostic lenses for seeing the problem from different angles before solutioning.

01

Need and pain significance

Who is affected, and does the problem matter enough to solve?

02

Operational reality

How does the work actually happen today?

03

Systems and data reality

Which systems, data, tools, reports, and workarounds shape the issue?

04

Problem materiality

What value, cost, risk, or opportunity is at stake?

05

Behavioural and change context

What behaviours, routines, incentives, and organisational conditions sustain the current state?

Need and pain significance.

Core question. Who is affected, and does the problem matter enough to solve?

Use this lens to understand

Who experiences the issue

What pain, friction, delay, confusion, or unmet need exists

How often it occurs

How severe it is

Whether people are working around it

Whether the issue matters to stores, customers, teams, or support functions

Example assumptions

Store teams experience exception handling as a recurring source of friction.

Customers are negatively affected when exceptions are not resolved the same day.

Team members have created workarounds because the current process does not meet their needs.

The issue is important enough that teams would prioritise fixing it over other pain points.

Operational reality.

Core question. How does the work actually happen today?

Use this lens to understand

The real process, not just the documented process

Who does what

Where handoffs occur

Where delays, rework, duplication, or unclear ownership exist

How the process varies by store, format, region, volume, or trading conditions

What happens when things go wrong

Example assumptions

The documented process does not match how stores actually manage exceptions.

Ownership of unresolved issues is unclear between store and support teams.

The current process creates duplicate handling or rework.

High-volume stores experience the problem differently from low-volume stores.

Systems and data reality.

Core question. What systems, data, tools, reports, and workarounds shape the issue?

Use this lens to understand

Which systems are involved

What data is available, missing, late, duplicated, or not trusted

Where manual workarounds exist

What reports or tools people rely on

Whether system constraints are contributing to the problem

What system or data dependencies would matter later

Example assumptions

Store teams need to check multiple systems to understand the status of an issue.

Key data is not available in one place.

Current reports do not support in-day prioritisation.

Data latency contributes to rework or poor decision-making.

Problem materiality.

Core question. What value, cost, risk, or opportunity is at stake?

Use this lens to understand

How frequent the issue is

How large the impact is

Whether the impact is financial, operational, customer, risk, or strategic

Whether the issue is isolated or network-wide

What happens if nothing changes

Whether the problem is worth further discovery or investment

Example assumptions

The current issue creates material labour effort through manual rework.

Delayed resolution contributes to customer contacts, cancellations, refunds, or escalations.

The issue is recurring rather than seasonal or isolated.

The value at stake is large enough to justify further discovery.

Behavioural and change context.

Core question. What behaviours, routines, incentives, and conditions sustain the current state?

Use this lens to understand

What people actually do today

Which behaviours are habitual or reinforced

What workarounds people rely on

What previous changes affect trust, fatigue, or readiness

Who would need to change behaviour if the problem were addressed

What leadership, training, or reinforcement might matter later

Example assumptions

Current workarounds persist because they are faster than the official process.

Store leaders would need to reinforce any future change in this area.

Team members may resist changes that feel like extra admin.

Existing routines leave limited space for proactive issue management.

STEP 5

Write current-state assumptions in a testable format.

Specific. Testable. Connected to the current-state hypothesis. Clear about the stakeholder, process, system, or behaviour involved.

We assume that **[stakeholder / process / system / behaviour]** currently **[does something / experiences something / causes something]**, which contributes to **[impact]**.

EXAMPLE

We assume **store teams** currently **spend material time manually checking multiple systems**, which contributes to **delayed exception resolution**.

EXAMPLE

We assume **unclear ownership between store and support teams** contributes to **unresolved exceptions remaining open longer than necessary**.

EXAMPLE

We assume **current reporting does not support in-day prioritisation**, which contributes to **reactive rather than proactive issue management**.

EXAMPLE

We assume **the problem is concentrated in high-volume stores**, which affects **the scale and shape of the opportunity**.

PATH B

Starting with a pre-defined solution.

Do not restart full discovery. Run a **current-state risk scan**. Check whether the solution test is built on risky, untested beliefs about the current state.

- 01 Describe the pre-defined solution
- 02 Articulate the solution hypothesis
- 03 Run a current-state risk scan
- 04 Identify solution and current-state assumptions
- 05 Decide what to test before, during, or after

STEP 1

Describe the pre-defined solution.

Make the solution explicit so the team can stress-test the assumptions it relies on.

We are testing **[solution]** to help **[users / teams / customers]** achieve **[outcome]** by **[how it works]**.

EXAMPLE

We are testing a new stockroom tool to help store team members locate, manage, and complete stockroom tasks more efficiently by providing clearer task guidance and stockroom visibility.

STEP 2

Articulate the solution hypothesis.

If we **[introduce solution]**, then **[users / teams]** will **[change behaviour]**, resulting in **[measurable benefit]**.

If we introduce the stockroom tool, then team members will complete stockroom tasks faster and with fewer errors, resulting in improved stock availability and reduced rework.

STEP 3

Run a current-state risk scan.

Do not assume the solution is targeting the right problem. Ask: *what must be true about the current state for this solution test to be meaningful?*

CURRENT-STATE ASSUMPTION (STOCKROOM TOOL EXAMPLE)	WHY IT MATTERS
Stockroom tasks are delayed because team members lack visibility or task guidance.	If the real issue is capacity, layout, or competing priorities, the tool may not solve the problem.
Team members can access and use a device during the stockroom workflow.	If not, adoption failure may reflect workflow constraints, not poor desirability.
Stockroom processes are similar enough across pilot stores for one test design to be meaningful.	If not, pilot results may not generalise.
Data in the tool reflects stockroom reality accurately enough to act on.	If not, users may lose trust quickly.
Store leaders will allow the tool to change task routines.	If not, the test may fail because behaviours are not reinforced.

STEP 4

Identify two types of assumptions.

CURRENT-STATE ASSUMPTION

What must be true about today's problem, process, systems, or behaviours?

Stockroom delays are mainly caused by lack of visibility, not labour capacity.

SOLUTION ASSUMPTION

What must be true about the proposed solution working?

The tool gives team members clearer task guidance than the current process.

STEP 5

Decide what to test before, during, or after the solution test.

Test a current-state assumption before the solution test if being wrong would make the solution test **invalid, misleading, unsafe, or wasteful**.

BEFORE

Test current state first

- Problem cause is unclear
- Solution may target a symptom
- Process variation could invalidate the pilot
- Data quality could destroy user trust
- The pilot is costly or disruptive
- Results would be hard to interpret without baseline

DURING

Build it into the pilot

- Risk is manageable
- Observation can be built into the pilot
- The solution test is the best way to reveal behaviour
- Baseline, usage, and outcome data can be captured together
- Failure would still produce clear learning

AFTER

Defer until scale

- Assumption is low risk
- Does not affect pilot validity
- Only matters for scaling
- Can be handled in later operational design

PRE-PILOT READINESS CHECK

What problem · what cause · what behaviour · what operational conditions · what system or data conditions · which high-risk assumptions · which must be tested before, during, or after the pilot.

CONFIDENCE RATING

Rate evidence honestly. Apply the trigger question.

Each assumption starts as **Assumed**. The rating describes strength of evidence, not the team's optimism. If we only have what people said, the highest rating is **Indicative**.

L1 **Assumed**

Belief only. Hypothesis or expert opinion. Every new assumption starts here.

L2 **Indicative**

Signals suggest it may be true. Interviews, surveys, desk analysis. *What people say is always Indicative at best.*

L3 **Demonstrated**

Observed behaviour in one real store. First level that counts as behavioural evidence.

L4 **Confirmed at scale**

Decision-grade. Representative multi-store trial, sustained usage, Data and Finance co-designed.

OFF-RAMP AT ANY LEVEL

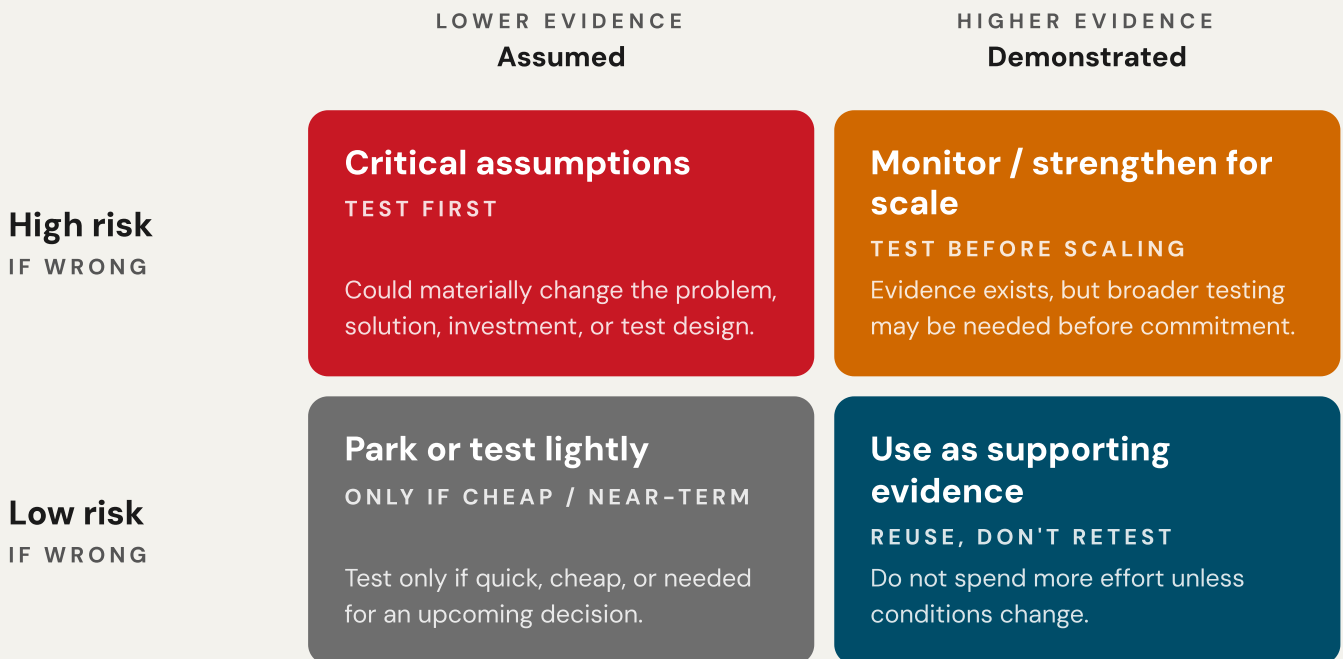
↓ **Disproven**

A fair test contradicts the belief. Report with pride.

PRIORITISE THE RISKIEST

Plot each assumption on a 2×2 to find the critical ones.

High risk if wrong, currently only Assumed. These are **critical assumptions**. They have the greatest potential to change the direction of the work.



WHEN SEVERAL ARE CRITICAL, ASK

What is the next most important assumption that would create significant change if wrong? Test the one most likely to change the problem definition, solution, investment decision, operating model, or test design.

LEARNING ACTIVITIES

Match the activity to the starting point.

PATH A · PROBLEM-LED

Understand the current state to define the problem to solve.

CRITICAL ASSUMPTION	LENS	LEARNING ACTIVITY	SUCCESS SIGNAL
Teams spend material time manually checking multiple systems	OPERATIONAL	Observe exception handling in one store	Repeated manual checking observed in real workflow
Current reporting does not support in-day prioritisation	SYSTEMS / DATA	Report and system walkthrough	Gaps between decision need and available data
Delayed exceptions contribute to customer escalations	MATERIALITY	Compare exception age with escalation data	Relationship visible in existing data

PATH B · SOLUTION-LED

Make the solution test valid and interpretable.

CRITICAL ASSUMPTION	TYPE	TIMING	LEARNING ACTIVITY
Stockroom delays are caused by lack of visibility, not capacity	CURRENT	Before	Store observation and baseline task analysis
Team members can use a device in the workflow	CURRENT	Before / During	Workflow observation and prototype handling
The tool provides clearer task guidance than current process	SOLUTION	During	Prototype trial in real conditions
Value case holds across store types	SCALE	Before scale	Representative multi-store trial

AFTER LEARNING

Synthesize the evidence. Decide by path.

Do not just update the confidence rating. Synthesize what the evidence means for the current state, solution, hypothesis, and next decision.

PATH A · PROBLEM-LED

SYNTHESIS QUESTIONS

- Does this support or challenge our understanding of the problem?
- Does this change who we think is affected?
- Does this change where the problem sits (process, system, data, ownership, behaviour)?
- Does this change the opportunity size or materiality?
- Does this reveal new systems, dependencies, or constraints?
- Does this create new critical assumptions?
- Are we ready to define the problem to solve?

DECISION OPTIONS

Continue discovery

Problem still unclear or critical assumptions untested.

Define problem

Evidence is strong enough to describe stakeholders, boundary, and impact.

Explore solutions

Problem is clear enough to generate and compare directions.

Stop

Problem is not material or worth addressing.

PATH B · SOLUTION-LED

SYNTHESIS QUESTIONS

- Does this support or challenge the current-state assumptions behind the solution?
- Does this support or challenge the solution itself?
- Does this change our mind about the solution hypothesis?
- Does this change the pilot or test design?
- Does this introduce new data, constraints, or operational realities?
- Does this create new critical assumptions?
- Should we proceed, pivot, stop, learn more, or scale?

DECISION OPTIONS

Proceed to test

Current-state risks understood enough for a meaningful test.

Adjust test design

Solution worth testing, but design must reflect current-state realities.

Pivot solution

Evidence supports the problem but challenges the solution.

Stop · Learn more · Scale

Standard off-ramps when evidence is contradictory, thin, or decision-grade.

COMMON MISTAKES TO AVOID

Nine ways teams get this wrong.

Mistake 01**Treating every initiative as if it starts the same way**

Problem-led needs diagnosis. Solution-led needs a risk scan. Do not use the wrong practice.

Mistake 02**Jumping to a solution too early**

Weak: Stores need a dashboard. **Better:** Stores may lack a clear way to prioritise exceptions during the trading day.

Mistake 03**Blindly testing a pre-defined solution**

Before piloting, ask: what current-state assumptions could make this test misleading if wrong?

Mistake 04**Treating the presenting issue as the real problem**

"The report is bad" may really mean unclear ownership, poor workflow timing, or distrust of data.

Mistake 05**Only interviewing stakeholders**

Interviews are useful, but Indicative at best. To reach Demonstrated, observe behaviour in realistic operating conditions.

Mistake 06**Mapping the ideal process instead of the real process**

Do not map what should happen. Map workarounds, skipped steps, duplicate entry, rechecking, informal handoffs.

Mistake 07**Assuming the problem is the same across all stores**

Check variation by format, volume, location, staffing model, capability, trading pattern, and seasonal pressure.

Mistake 08**Testing too many critical assumptions at once**

Focus on the assumption most likely to change what the team does next.

Mistake 09**Treating Disproven as failure**

Disproven means a belief was tested and a downstream mistake was avoided. Report with pride.

SIMPLE WORKING TEMPLATE

One row per assumption.

Use this template to track assumptions through either path. Include starting point, assumption type, and lens or area so the prioritisation is honest.

PATH	TYPE	LENS / AREA	ASSUMPTION	LEVEL	RISK	CRITICAL?	TIMING	NEXT STEP
A	Current-state	Need and pain	e.g. teams treat exceptions as friction.	INDICATIVE	HIGH	YES	n/a	Observe in one store.
A	Current-state	Systems / data	...	ASSUMED	HIGH	YES	n/a	...
B	Current-state	Op / system risk	...	ASSUMED	HIGH	YES	Before	...
B	Solution	Adoption	...	ASSUMED	HIGH	YES	During	...
B	Solution	Viability	...	DEMO	HIGH	NO	After	...

One-line rule. If the initiative starts with a problem, use assumptions to understand the current state before defining the problem to solve. If it starts with a pre-defined solution, run a current-state risk scan to identify any Assumed, high-risk beliefs that could invalidate or reshape the solution test.