



Retail Hive workshop | Session summary | 15 October 2026

Delivering Change Safely: Testing, Governance and Legacy Integration in an AI-Driven World

Session goals

What we set out to explore, building on our earlier Retail Hive and RetailJam round tables.

1

Treat AI as a given

Move past whether to use AI, and focus on what it means for testing, governance and integration.

2

Look beyond faster development

If development is quicker, or bought in from suppliers and platforms, what does that mean for the rest of the lifecycle?

3

Make it yours

Apply the thinking to your own AI maturity and tech stack, then take it back to your organisation.

Three things protect you, in this order: know where your risk and opportunity live, clear accountability, and proportionate testing.

What we heard: quality is harder to locate

Three themes that came up again and again across the room.

The volume is the problem

The process is not broken. When anyone can build something in minutes, you cannot look at everything.

Fuzzy outputs, fixed processes

The biggest risk raised: non-deterministic outputs feeding deterministic workflows, often without anyone noticing.

Finance is under scrutiny

Auditors are now asking whether AI has touched the financial data. Would you know if an agent posted a journal?

“It works” is not the same as “we understand it”.

The Graded Test Approach

Test in proportion to how much it matters.

G0	G1	G2	G3	G4
Personal utility	Shared team tool	Internal dependency	Customer-facing	Mission critical
If it breaks "I shrug"	If it breaks "Someone pings me"	If it breaks "Work stops"	If it breaks "Customers notice"	If it breaks "We lose money or trust"
Release posture Ad-hoc, local	Release posture Internal, flagged	Release posture Guarded, rollback ready	Release posture Controlled rollout, rollback ready	Release posture Fix-forward, formal sign-off

Moving up a grade should be intentional, with test evidence, logging, security checks and risk acceptance. The right level of testing is not the most testing: it is the most appropriate for where the work sits.

Deterministic and non-deterministic

Deterministic

The same input gives the same output.

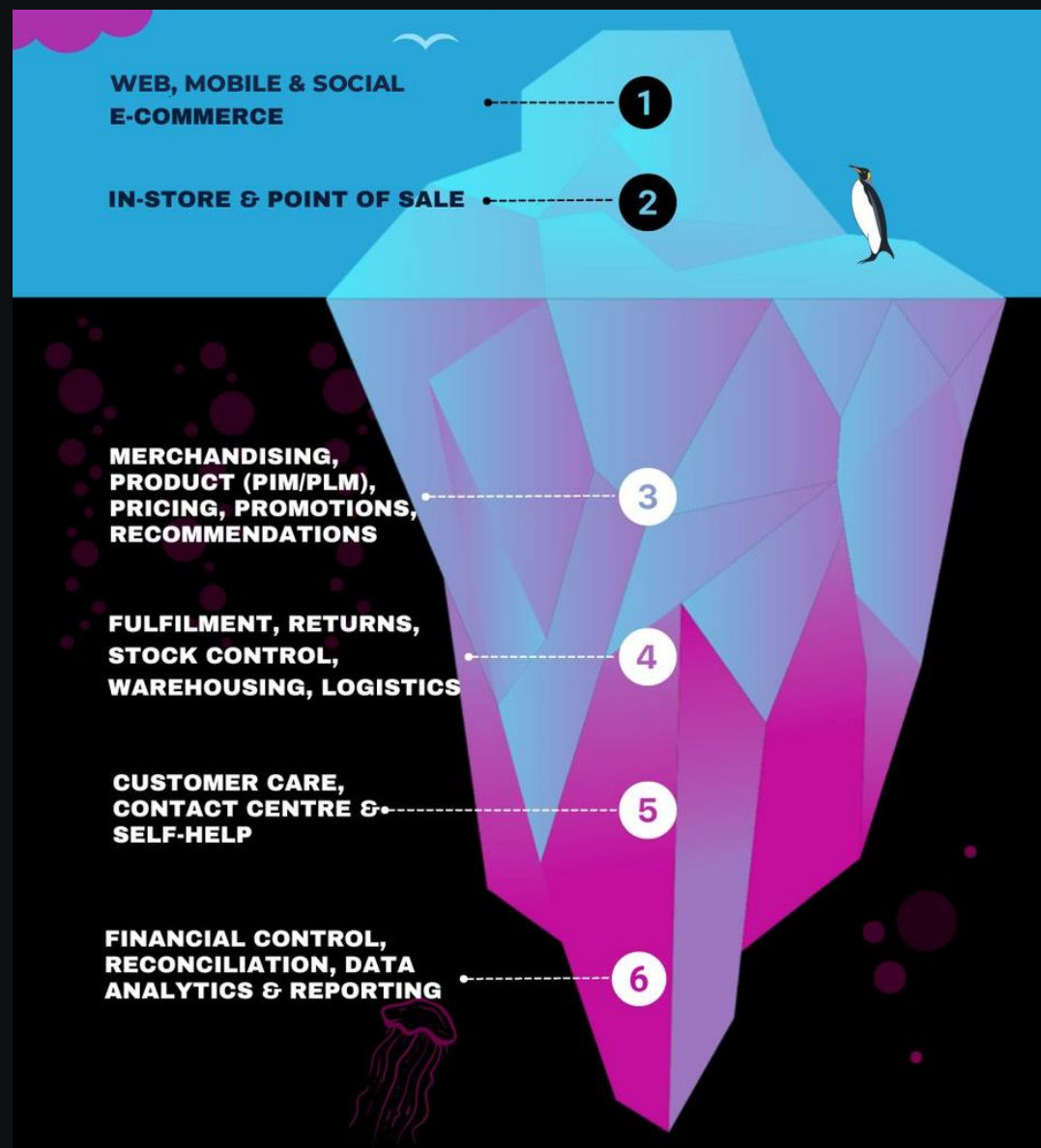
- Scripted tests that assert on a known result
- Pass or fail, so you know where you stand
- Still essential for business processes and end-to-end flows

Non-deterministic, or probabilistic

The same input does not always give the same output.

- Search moving from “a pair of trousers” to “an outfit for a wedding”
- The right result is a matter of judgement: good enough, not pass or fail
- Applies to analytics and reporting as well as search

What we heard: the risk is greatest where non-deterministic results feed processes that expect the same answer every time.



Worksheet 1 | What we heard

Your AI maturity

- 1 Control varies by layer**
 Web and mobile are largely yours; social is not. Shaping AI inputs can make a system more predictable over time.
- 2 Look for the knock-on effect**
 AI in one layer can change the risk in another, and the integrations between layers are where problems surface.
- 3 Finance is under scrutiny**
 Auditors are asking whether AI has touched financial data, and agents in other layers can reach it.

Risk and opportunity here are about the solution itself, not about delivery.

Where AI sits across the stack

Common themes from the room on how far AI is built into each layer today.

Where adoption stands

Most established

Web, mobile and social, and customer care: search, recommendations, product content and chat.

Growing

Merchandising, pricing and fulfilment: automating manual processes, freight and warehouse planning.

Earliest

In-store and finance: mostly thinking or piloting, with reconciliation and response agents emerging.

In use is not the same as confident

Customer-facing AI is widely in use, yet confidence varies. It is lowest where the AI comes built into third-party platforms.

The opportunity sits in finance

Margin, cost management, forecasting and financial control were seen as the big prizes, in one of the least mature layers.

Governance is the top risk

Governance came up most often, alongside cost, duplicated effort and a lack of repeatable approaches.

In-store AI is mostly colleague-facing so far: knowledge management, tasking and scheduling.

Use cases from the room

Real examples shared on the day, from above and below the waterline.

Customer-facing

AI-generated marketing creative replacing costly location shoots.
Agentic contact centre journeys replacing standard IVR.

Fulfilment and stock

Robots running nightly stocktakes, photographing every location and flagging damaged or leaning pallets.

Data and back office

An agent interrogating master data after an ERP migration. A bot chasing purchase approvals daily until they are signed off.

Worth asking of each one: if the AI in that layer broke, who would notice, and where would it sit on the G0 to G4 scale?

Worksheet 2: Typical path to live

What we heard about getting change live when AI is in the mix.

Governance Not a stage: roles, accountability and oversight across all areas, human or machine.

Backlog and requirements



Build or buy



Test



Deploy and run

Volume is the constraint

- Anyone can build something in minutes, so demand rises
- You cannot look at everything: be deliberate about where people focus

Triage by type of change

- Fast-track low-risk cosmetic changes
- Slow down financial and regulatory change, with a human in the loop for now

Delivery models and trust

- Heavily sliced delivery models sit at odds with an AI-enabled lifecycle
- Does quality hold as work passes between agents?

Across the room, AI is most established in build and least in deploy and run. Testing is lagging behind build, risk rises towards test and deploy, and exit criteria for AI-assisted change remain an open question.

Where will we be in five years?

How the room sees tools, people and skills changing across the path to live.

1

Ops focus on outcomes

Business users should not need to understand APIs or platform internals to get the outcome they want.

2

Translation layers

Agents and AI integration tooling map plain-language requirements to the systems underneath.

3

An AI-enabled lifecycle

Requirements, acceptance criteria and tests generated end to end, with people reviewing. Waterfall may be worth another look for well-scoped work.

4

Deciding what risk to accept

Automation covers functional checks, so UAT focuses on acceptance. The question becomes what risk we accept at deployment.

One group's headline: an owned, AI-enabled delivery factory, with people leading change management.

When coding gets cheaper, the cost does not disappear: it moves to validation, release and change management.

Keep the basics, then add

Keep the basics

Deterministic: it asserts on known outcomes.

- Clear requirements, in one place
- Test cases and automation
- A good defect process
- A process that is clear before you automate it

Add for AI

Non-deterministic: results are not always the same twice.

- More exploratory testing
- UAT that handles the edge cases
- Fuzzy matching: good enough
- Closer product owner, tester and developer links
- Test environments that can recreate the model
- Production business rules fed back into test

AI and automation make a process faster. They do not make an unclear process clear.

Close

Questions to take back

1. If your auditor asked tomorrow which of your processes AI has touched, how confident would you be in the answer?
2. Where is a non-deterministic output feeding a process that assumes the same answer every time?
3. Which of your changes sit at G0 or G1, and which at G3 or G4? Is your test effort split the same way?
4. If change volume doubled next year, which gate on your path to live would break first?
5. Who reviews what an agent writes, and who owns it: you or your supplier?
6. Are the business rules created in production coming back into your test environments?
7. Is your delivery model set up for the AI-enabled lifecycle you want, or the one you had five years ago?
8. What level of risk are you prepared to accept at deployment, and who signs that off?
9. Is your testing keeping pace with the AI in your build?
10. Where AI is in use, have you defined the exit criteria for each stage?
11. Is the AI you use but do not control, built into third-party platforms, graded and tested?

“It works” is not the same as “we understand it”.

Your cheat sheet from the session has space for your own questions.



Thank you

Steve Dennis steve.dennis@wearespike.co.uk

Darryl Kennedy darryl.kennedy@wearespike.co.uk

wearespike.co.uk