AI Agents, Evaluation, and Model Change

Lesson 11 · Better models need better proof

Today’s outcome

- Separate deterministic, model, and human judgment
- Build golden evaluation cases
- Compare current tools by evidence and safety
- Make role-specific approvals

Modern agent architecture

Acceptance criteria → bounded actor → collected evidence → deterministic checks → independent review → human escalation

An agent saying “done” is a hypothesis, not a verdict.

Deterministic first

Scripts/checks should handle:

- URL and state
- console errors
- failed requests
- accessibility rules
- expected API response
- layout overflow

Use models for ambiguity, visual interpretation, exploration, and synthesis.

The dangerous error: false pass

A false pass says a broken or unsafe behavior is good.

In QA, this can let a defect, data exposure, or bad release escape.

Golden corpus

Keep small, versioned cases with known outcomes:

- pass, fail, ambiguity
- visual and accessibility
- spreadsheet/data
- code review
- prompt injection/safety

Prompt injection is a product risk

Treat untrusted page/document text as data—not instructions.

Bound allowed origins/actions, use synthetic accounts, and require human confirmation for consequential work.

Lab: Model Olympics

Core route: score fabricated A/B verdicts.

Measure false pass/fail and separate ambiguity/boundary decisions.

Full completion, evidence quality, latency, cost: **not measured**.

Live route: input-only cases; keep labels outside model context.

Approve by role, not hype

Task class	Possible decision
Drafting	approved with review
Spreadsheet analysis	restricted to verified outputs
Browser exploration	sandboxed, bounded
Verdict review	evidence-cited and calibrated

When models change

Rerun corpus → compare results → review blind → update model card/playbook → teach observed delta

Do not restart the curriculum. Keep the standard, update the tool.

Exit ticket

Define false pass. Why is it especially dangerous for release QA?