

Our Banking Client: shared QA platform

A payment-release story: 75 offshore QA staff, heavy manual testing and constrained test environments.

75

offshore QA staff

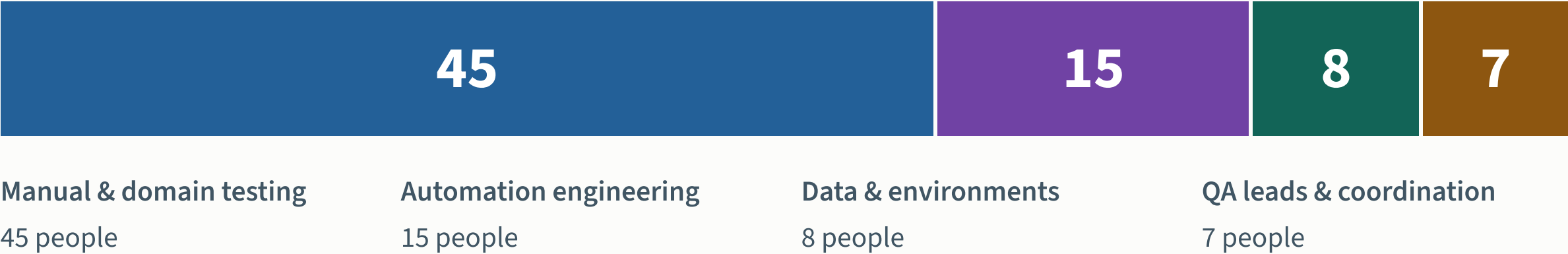
One payment release.

A platform the next team can reuse.

Executive briefing • Strategic vision • Client scenario

The QA organization

75 offshore QA staff across five squads; 45 focus on manual and domain testing. Environments and vendor test slots limit parallel work.



The pilot draws eight QA staff from this population. More engineers alone do not create more vendor test capacity.

QE testing scenario: one timeout, two debits

PAY-142 simulates a CAD 100 transfer accepted before a timeout. The customer retries; an injected duplicate-posting defect tests whether QE catches the second debit.

Customer

Send CAD 100

Payment API

→ Key: pay-142

Provider

→ Accept; response times out

Customer retries

→ Reuse pay-142

Injected test defect

2 journals

Payer CAD 800

Recipient CAD 200

FAIL • duplicate posting

Expected test outcome

1 journal

Payer CAD 900

Recipient CAD 100

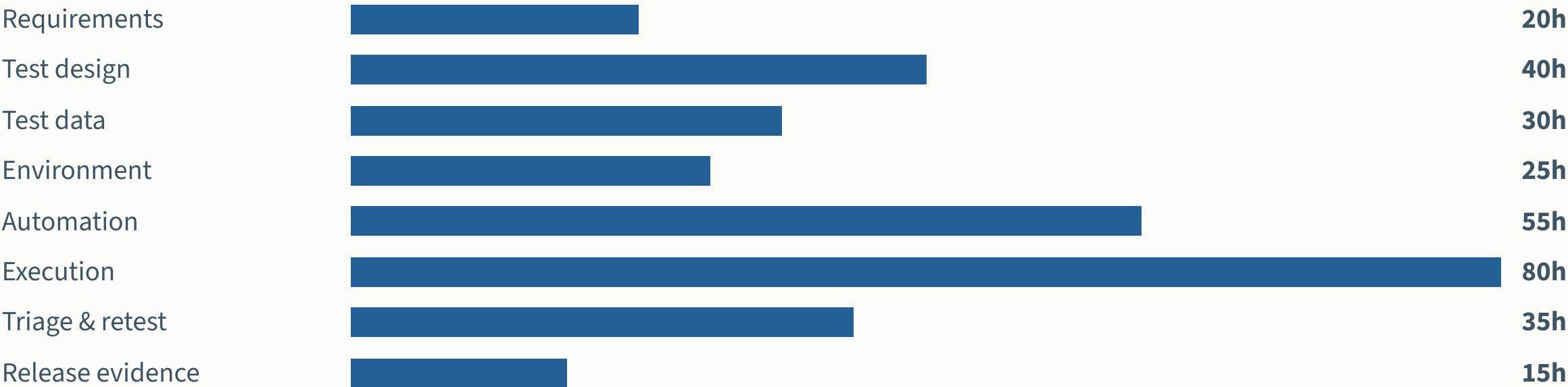
PASS • expected state

QE testing scenario. The injected defect causes the duplicate posting; a timeout alone does not imply two debits. Initial balances: CAD 1,000 and CAD 0. One journal contains one debit and one credit. No fees, FX or other activity.

Expected: one transfer, one balanced journal and one customer-visible result. This is a QE test scenario, not a reported client incident.

Where the release work goes

Modelled hands-on effort for one bounded release test pack. Eight stages add to 300 hours.

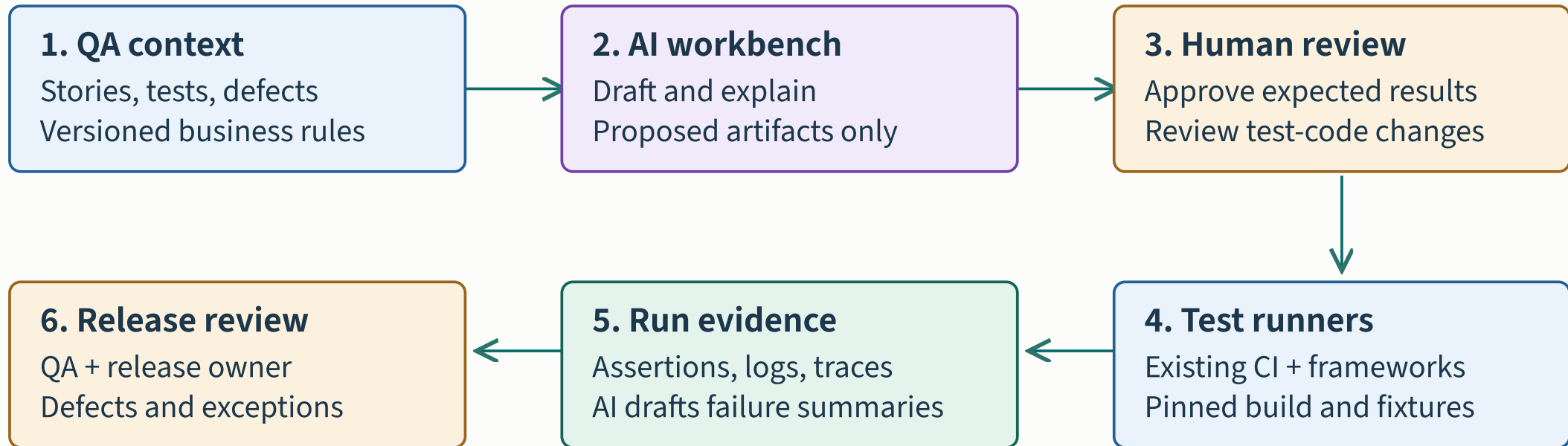


300 person-hours for this pack. Each activity belongs to one stage. Baseline already includes its ordinary review and rework.

The pack spans 10 business days. Waiting time and human effort use different units and must be measured separately.

The platform investment

Common context, fixtures and evidence serve every squad. Application teams retain business ownership.



Shared foundation: synthetic data · dependency stubs · artifact IDs · ownership · model configuration

Existing test frameworks and pipelines execute approved work. AI supports preparation and interpretation.

Before / after: separate modernization from AI

01 / Manual QE

Copy cases and run by hand
Shared data + scarce vendor slots
Reconcile screenshots and logs



02 / Modernized QE

Reviewed automated tests
Isolated data + virtual services
CI retains repeatable run evidence



03 / AI-assisted QE

Draft cases, code and fixtures
Suggest scope, diagnosis and repair
Summarize proof for human review

Establish the baseline

Active work · vendor waits · coverage gaps

Prove repeatability

Environment readiness · reruns · raw evidence

Isolate added AI value

Candidate acceptance · review · rework

Same payment scope. The two target states are proposed; no productivity or quality improvement has been measured.

Delivery maturity changes the first move

The same tool creates different work in a shared manual environment and a repeatable delivery system.

01

Shared & manual

Begin with requirements, test design and evidence drafts. Establish stable fixtures and repeatable runs before expanding generation.

02

Mixed maturity

Pilot payment APIs and one web journey. Add synthetic fixtures, provider stubs and evidence links alongside AI assistance.

03

Repeatable delivery

Extend generation and triage across more application teams. Evaluate test selection in shadow mode before changing required coverage.

These profiles illustrate effort, not adoption approval. Review twelve dependency areas before committing to scope, dates or benefits.

What 78 hours of gross capacity means

Mixed-maturity target: 300 hours becomes 222, including 40 hours of review.



Purple segment = review. Total assisted effort = 222h.

78h	66h	33h	15 packs
gross capacity	after operation	usable capacity	to recover 480h setup
300 – 222	78 – 12	66 × 50%	rounded up, steady assumptions

Conditional capacity arithmetic; adoption prerequisites are unverified. Tool, cloud and vendor charges are excluded. This is not cash ROI or measured AI impact.

Ownership within the offshore model

OWNER	ACCOUNTABILITY
Shared enablement	Own context adapters, templates, fixtures, runner integration and coaching.
Application QA teams	Own business scenarios, expected results, execution and defect disposition.
Product and release owners	Resolve requirement questions and accept the release evidence.
Delivery management	Provide repository access, overlap hours and an explicit backlog for freed capacity.

Reskill domain testers through paired reviews. Agree supplier incentives and reuse ownership before scaling.

Observed time framework for the API pilot

Record actual pilot time from baseline through repeated API runs. Example windows remain subject to readiness.

Observation status: not yet recorded		
Planning window · weeks 1–2 Observe the baseline Record API test preparation, active effort, waits and rework for one comparable pack. Evidence gate: Reviewed API scope + usable baseline Record actual dates and elapsed time; track active person-hours, waiting reasons, review and rework separately.	Planning window · weeks 3–8 Run the API pilot Record setup separately; measure review, execution, triage and retest over repeated packs. Evidence gate: Required assertions pass + comparable run evidence	Planning window · weeks 9–12 Confirm repeatability Measure second-team setup, support effort and repeat runs before choosing the next scope. Evidence gate: Reusable pattern + evidence-based next step

Planning windows are illustrative. Environment, data, virtualization and reviewer readiness determine when each phase can progress.

The evidence needed to expand

MEASURE	EXPANSION EVIDENCE
Quality	All mandatory payment assertions pass; no unexplained skips or weakened checks.
Effort	Include preparation, review, correction, retesting and platform operation.
Delivery	Report elapsed time, waiting reasons and failed-run recovery separately.
Reuse	A second squad runs the pattern and maintains it without the pilot authors.

A faster draft is insufficient. Compare tasks of similar scope and retain conventional work as a reference.

Adoption depends on funded foundations

SCOPE FOLLOWS PROVEN CAPABILITY

Inputs + people + evidence + approved AI access + a measured baseline

DRAFTING

Reviewed scenarios

Approved rules and observable outcomes.



Domain QA accepts the scenario matrix.

Assess test design →

DIAGNOSIS ONLY

Evidence-linked triage

Retained failures, known dispositions and usable diagnostics.



A reviewer checks the diagnosis; no retest runs.

Assess diagnosis →

EXECUTION

Reviewed test changes

Add repeatable environments, CI, fixtures, provider control and reliable runners.

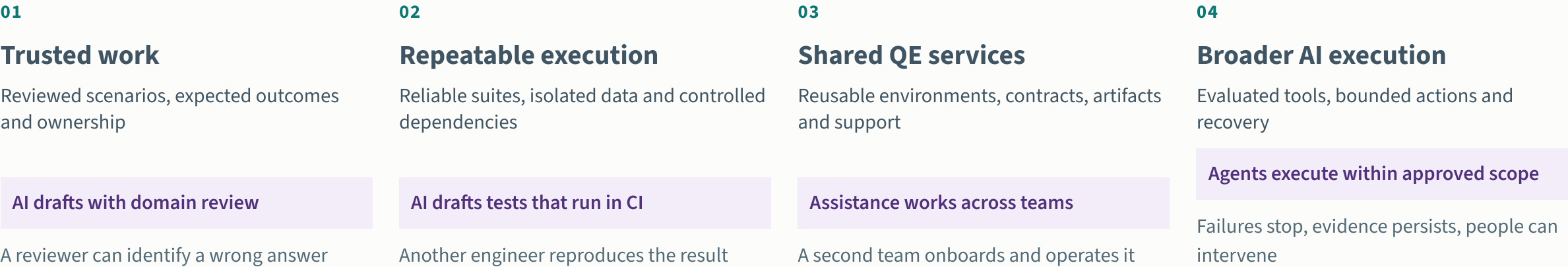


Independent assertions challenge the candidate.

Assess automation →

Our Banking Client has not been assessed. The dependency backlog can change scope, cost and timing; the 480-hour setup allowance is not a client quote.

Modernization enables wider AI adoption



Proposed capability progression. Reviewed assistance can begin while foundations improve. Broader execution requires evidence for its specific scope. [Modernization workstreams and sources](#)

For Our Banking Client, fund repeatable payment tests and shared QE services alongside reviewed AI assistance. Validate this capability roadmap through client discovery.

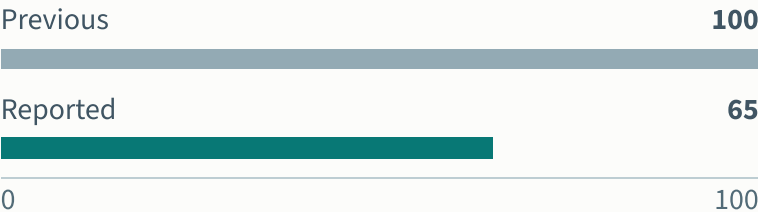
Peer evidence for the payments pilot

Libra Internet Bank

35% less time

Test-creation time

Index: previous time = 100



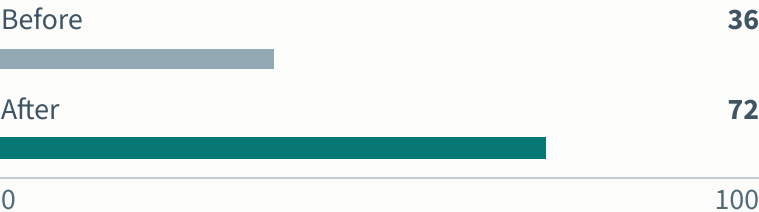
UiPath customer case. Normalized from the reported reduction; review effort is unspecified. [F1](#)

Goldman Sachs

36% → 72%

Unit-test coverage

Share of a selected module (%)



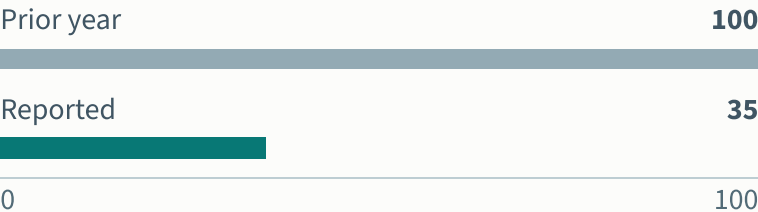
Diffblue customer case. Coverage measures exercised code; it does not establish financial correctness. [F2](#)

Fiserv

65% fewer incidents

Major incidents

Index: previous year = 100



Tricentis modernization case. AI testing was a later pilot; this is not an AI-attributed reduction. [F4](#)

Separate outcomes and denominators. Each panel has its own meaning; no combined savings rate is calculated.

Peer results support a trial. Outcomes for Our Banking Client remain unmeasured.

Our Banking Client: where to begin

Requirements ready

Onboarding design

Draft scenarios in the current test-management workflow.

Proof

Accepted coverage and measured review effort.

Execution repeatable

Payment API tests

Generate candidates in the existing framework and CI pipeline.

Proof

Correct behavior passes; a known fault fails.

Diagnostics available

Failure triage

Draft diagnoses alongside the existing disposition process.

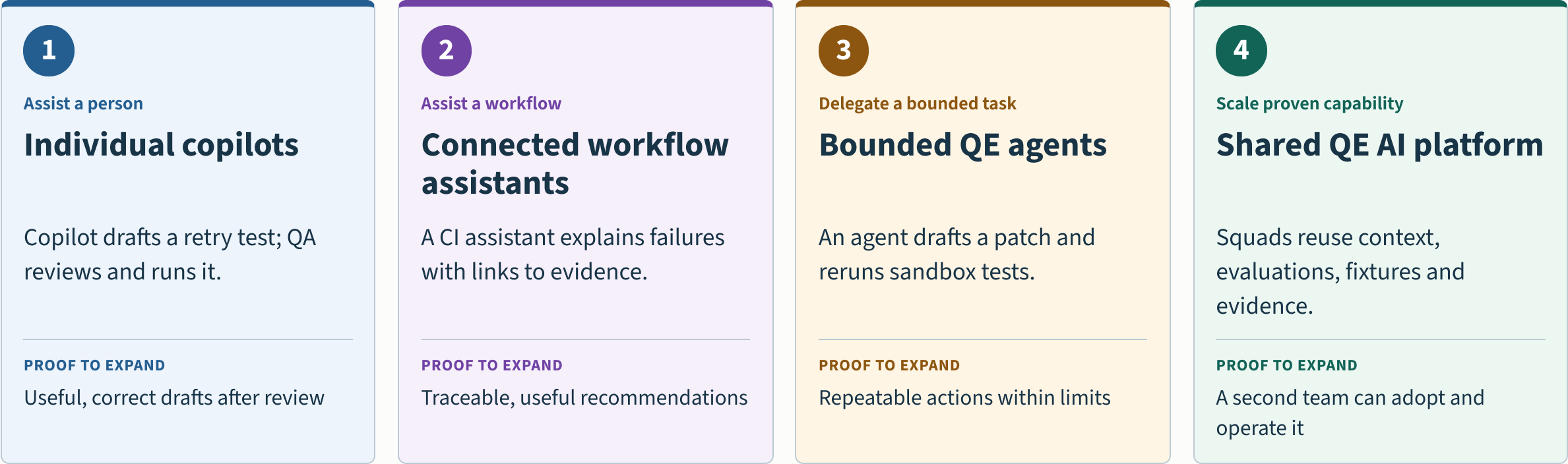
Proof

Faster correct decisions with traceable evidence.

Proposed entry points. Choose one using the client's constraints; [inspect prerequisites](#) and [prepare a trial brief](#).

Choose one entry point, validate its prerequisites and agree how reviewers will judge the result. Wider rollout depends on local evidence.

Four layers of AI-assisted QE

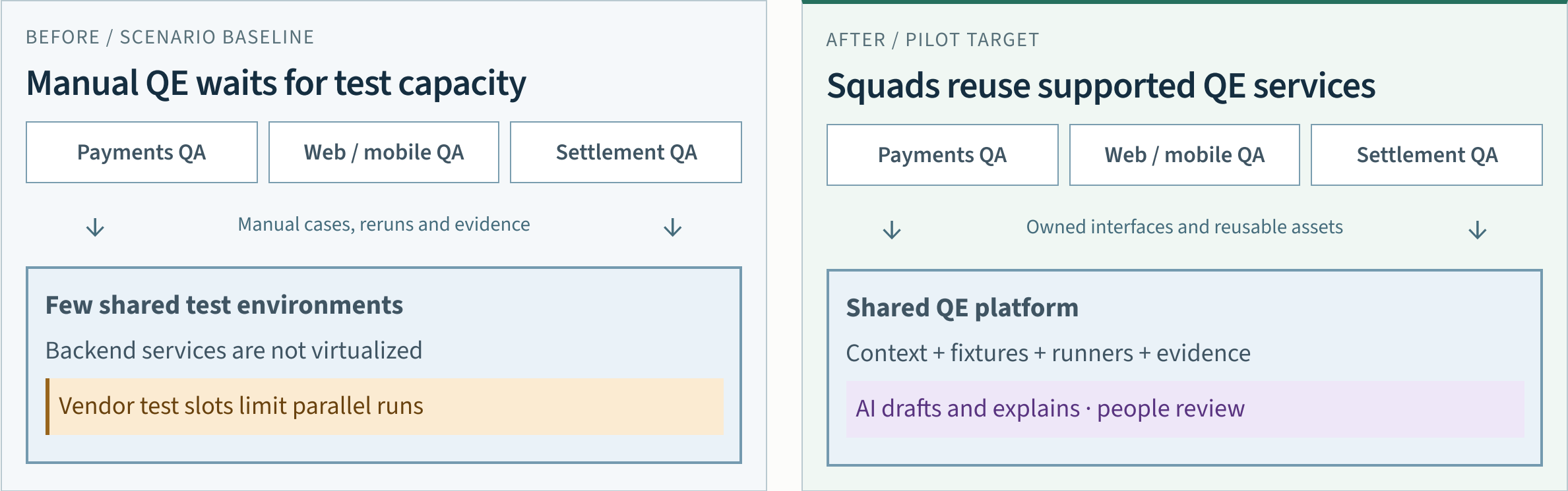


Shared foundations · start on day one Context · data · reliable tests · CI · QE modernization · owners

Layer 4 supports layers 1–3. Scale useful assistance without having to delegate more actions.

Authored roadmap. Expand with validated prerequisites and client evidence.

Before / after: the shared QE architecture



QE modernization: isolated runs and controlled dependencies. **AI assistance:** preparation and interpretation.

Starting point: heavy manual QE and scarce vendor environments. Pilot target: controlled service tests, with real integration still required.

Outcomes beyond hours

01

Catch payment faults

Catch every critical seeded fault

Reviewed fault scenarios and independent balance assertions

02

Cover critical journeys

Exercise every critical scenario

AI drafts edge cases; domain QA approves the scenario matrix

03

Explain each release decision

Complete every decision record

CI captures evidence; AI drafts a source-linked summary

Supporting proof

Trust repeat runs

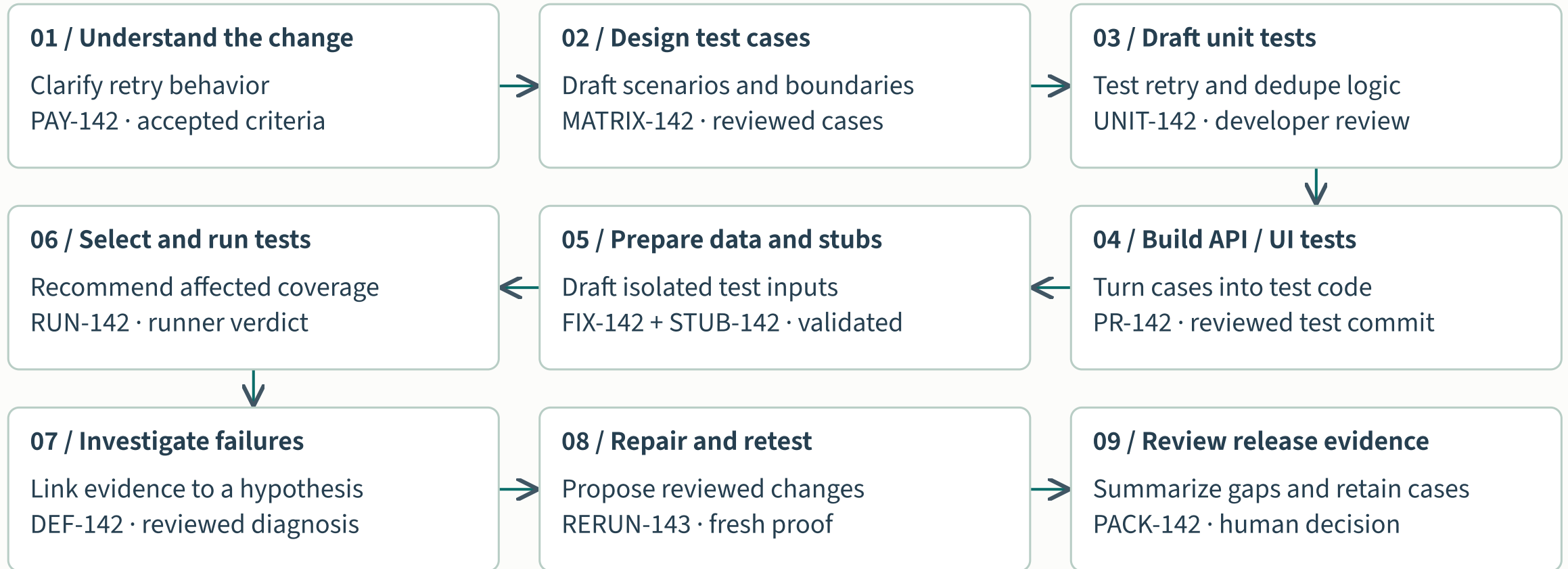
Validate real integrations

Enable the next squad

Baseline: Not recorded · Observed after: Not recorded · Targets proposed for pilot agreement.

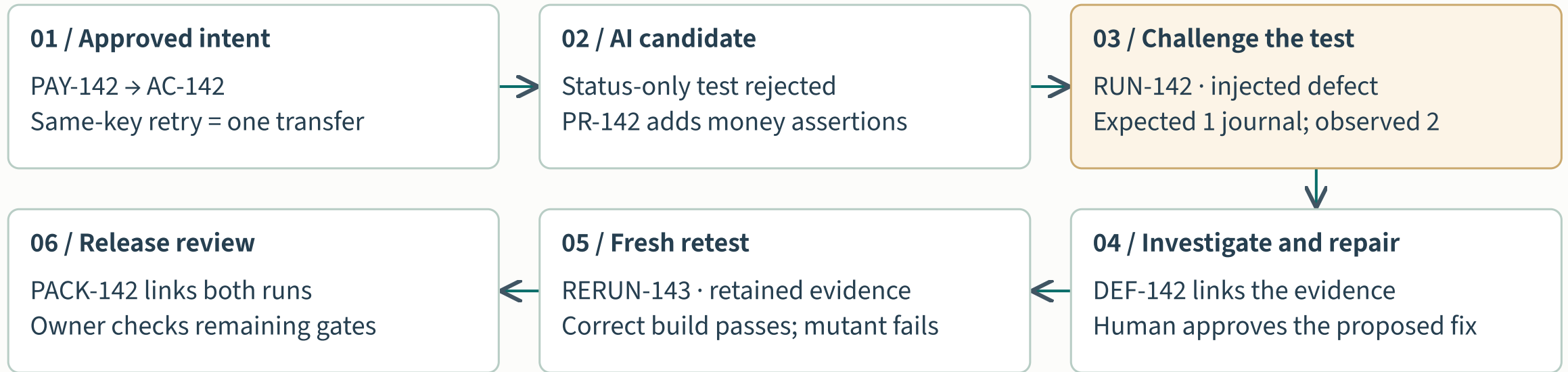
Business aims: fewer escaped payment defects and emergency fixes. These proposed pilot measures do not establish those outcomes.

What AI contributes, from requirement to release



PAY-142 is an authored scenario. Each AI output needs review; test runners execute and people decide.

Follow one payment test through the handoffs



Illustrative artifacts and outcomes. Preserve failed evidence; a passing retest alone does not approve release.

Brainstorm questions

The next decision: assess one payment workflow with QA, development and platform leads.

1. “Across the 75 QA staff, where do people spend time and where are they waiting?”

2. “Which payment journey has stable requirements, observable outcomes and an owner who can join a pilot?”

3. “Could one platform team remove duplicated work across the squads?”

4. “If we return capacity, which backlog or release commitment will use it?”

Agree the bounded pilot and its evidence before promising wider rollout. See the discovery brief for owners, commercial scope and timing.