How AI Adapts When the World Changes

The world changes.
The model inside doesn't learn from it.

SmartPlay: an LLM-powered system

Qazybek Beken

LLMday San Francisco / October 1, 2026

After you ship, the model stops learning

Mistakes

It can repeat last week's mistake.

The model doesn't learn from what happened.

Fixes

A retrain can break what used to work.

New patterns can push out old skills.

Scores

A score can't show whether it got better.

Improving and unlucky can look like standing still.

Nothing learns unless something around the model changes.

?

Can it get better with experience, if the model never changes?

And if it does, how would you know?

A C in math. Now what?

A good student, all term. Then, suddenly:

B

Their usual grade

C

This test

Panic says: a tutor, every day.
That's retraining because the score fell.

A calm teacher asks five questions first.

What to look at instead of the score

1 Change
What is different now?
Four kinds of change

2 Evidence
How sure are we?
One bad test, or a pattern?

3 Location
Where should it live?
Context, memory, parameters, rules

Can it get better with experience, if the model never changes?
A student, then SmartPlay

4 Retention
What must keep working?
Learning without forgetting

5 Evaluation
Does it help later?
The next tests decide

In the real world
The same five questions
SmartPlay · three businesses

Change comes in four kinds

What changed? Often a mix of four kinds.

Inputs
Material they haven't seen.
Fractions, for the first time.

Mix of outcomes
More of the hard questions.
Twice the word problems on this test.

Meaning
The same answer now scores differently.
A new teacher gives marks for the working, not just the answer.

What you can see
The records are filtered.
They missed the easy quiz, so only the hard test counts.

Name the kind of change before choosing a fix.

Field terms: covariate shift, label shift, concept shift and selection bias.

One C, or a pattern?

One C

A warning.
Maybe fractions.
Maybe one bad night.

Cs on three fraction tests

A pattern.
Now it's real.
Worth a fix.

A warning is not a diagnosis.

Before you change anything, check the next two tests and the homework.

Fine homework but bad tests? Maybe the problem is the clock, not fractions.

Where should the fix live?

Context

A rule sheet
on their desk

Memory

A notebook of
their mistakes

Parameters

Drills with a tutor
(retraining)

Rules

A new timetable
for homework

Learning fractions without losing multiplication

Three weeks of only fractions, and multiplication slips.

Replay: mix old problems into the new homework.

New homework

Fractions,
every day.

Replay

A few old
multiplication
problems in each set.

Test both

Fractions and
multiplication,
on every test.

A fix that breaks what worked isn't a fix.

In machine learning: forgetting, and replay.

Did it help? The next two tests decide

Before and after the notebook

Before

Fractions: C, C, C.
Multiplication: A.

After

Fractions: B, B+.
Multiplication: still A.
Two extra hours a week.

Better, and nothing worse. Keep the notebook.

If it didn't help, drop it and try something else.

Five questions, one road

Skip 1 and 2: a tutor for one bad night. Skip 4 and 5: the times tables are gone.

1 Change — What changed? Fractions, for the first time.
Not a new test format. Not a sick day.

2 Evidence — How sure are we? One test. Check the next two.
One grade can mean one bad night.

3 Location — Where do we fix it? A rule sheet, a notebook of mistakes.
Context and memory first. The daily tutor, like retraining, comes last.

4 Retention — What must keep working? Multiplication.
Mix a few old problems into the homework.

5 Evaluation — Did it help? Fractions better, multiplication no worse.
Worth the hours? Keep the notebook. If not, drop it.

From “something went wrong” to a change we keep, or throw out.

1–5

SmartPlay

The same five questions. Now the student is a language model.

The squad and the starting eleven

£100m

initial virtual budget

15 players

Choose 11 starters.
Keep 4 substitutes on the bench.

Captain: double points

Each gameweek: transfers, captain, maybe a chip.

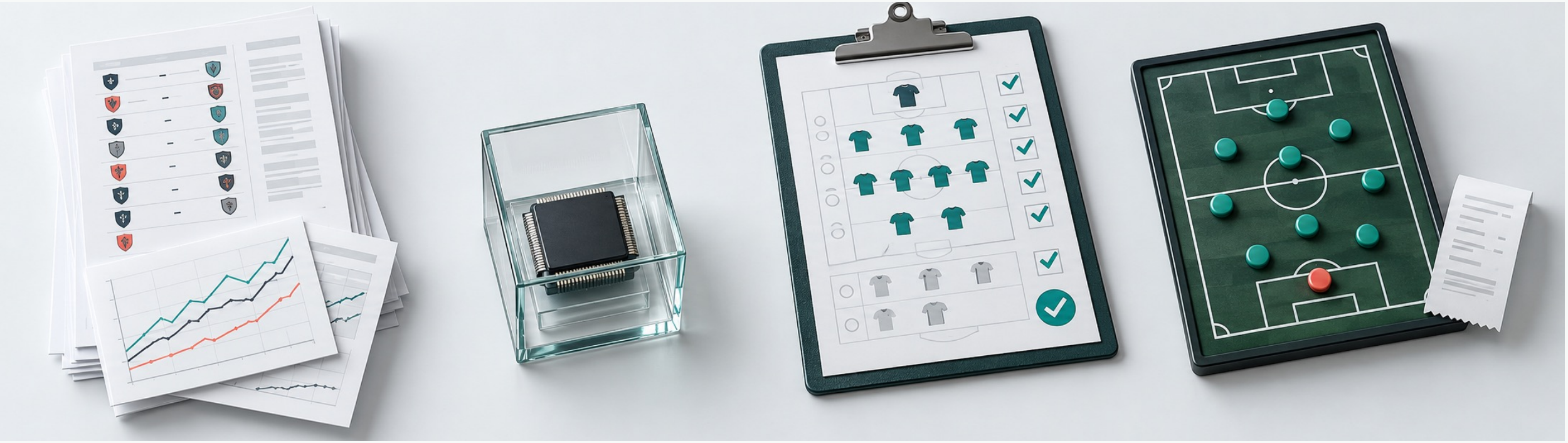
Same road, new names

Everything the student had, SmartPlay has.

In school	In SmartPlay
The student	The LLM, never retrained
A test	A gameweek: one decision, then the real results
The grade	The points
A rule sheet on the desk	Today's evidence, in the prompt (context)
A notebook of mistakes	Notes from earlier reviews (memory)
Drills with a tutor	Retraining the playing-time forecast (parameters)
A new timetable	Checks every plan must pass (rules)

The same four places in both: context, memory, parameters and rules.

SmartPlay's decision pipeline



Evidence

Dated news, forecasts and legal plans

LLM reasoner

One call, no tools: picks a legal plan

Validator

Checks the plan against the rules

Executor

Makes the moves, checks the result

Has it learned anything?

The first five gameweeks, fetched 21 September 2026

299

SmartPlay's team

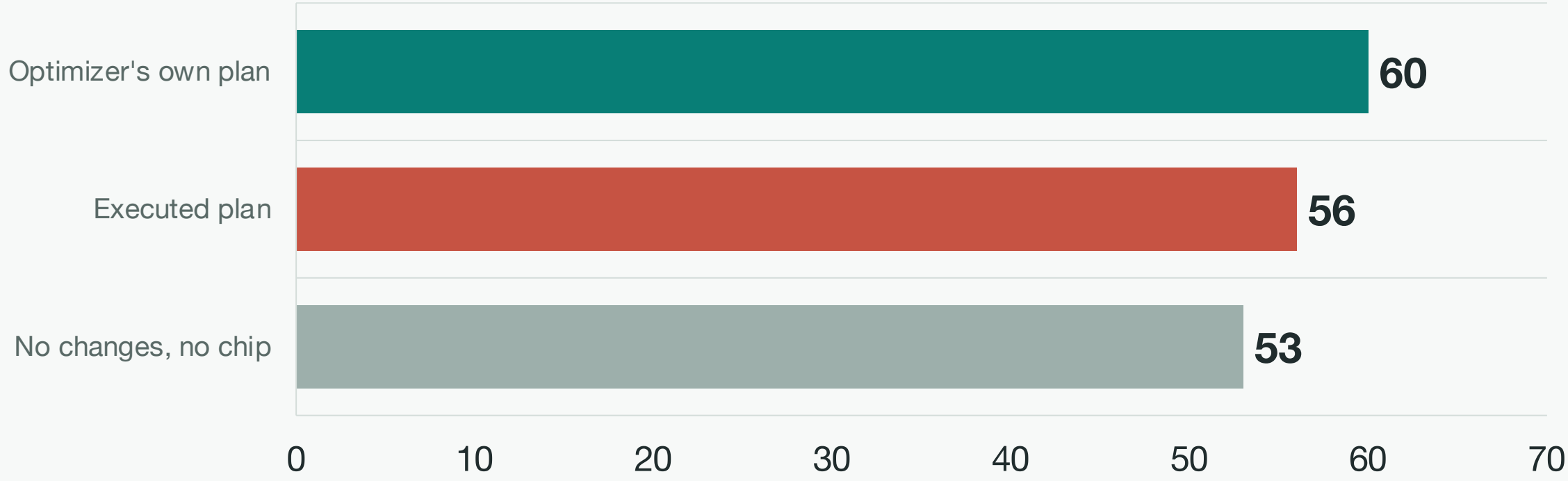
299

Average manager

The score cannot tell you.
Two of those rounds had no decision at all.

In gameweeks 4 and 5 no decision was made. The squad simply rolled over.

One gameweek, three saved plans



The 60-point plan was the optimizer's own. The executed plan differed from it only in the captain. Both use Bench Boost. The 53-point plan keeps it.

This comparison does not establish a season-long advantage.

SmartPlay's notebook of mistakes

What two reviews showed

The same gap, twice:
a transfer never compared
with keeping the squad.

The note

Report the missing check.
Keep the source reviews.
Avoid guessing the cause.

The model doesn't change. Its notebook does.

Research: Reflexion, Shinn et al. (2023). SmartPlay's rule is simpler and deterministic.

The memory rule is built in. It switches on only when the test says it helps.

Learning new patterns without losing old skills

Retraining the playing-time predictor

Replay brings older examples into training.

New examples

Recent player data and actual minutes.

Replay

Selected earlier examples, with their context.

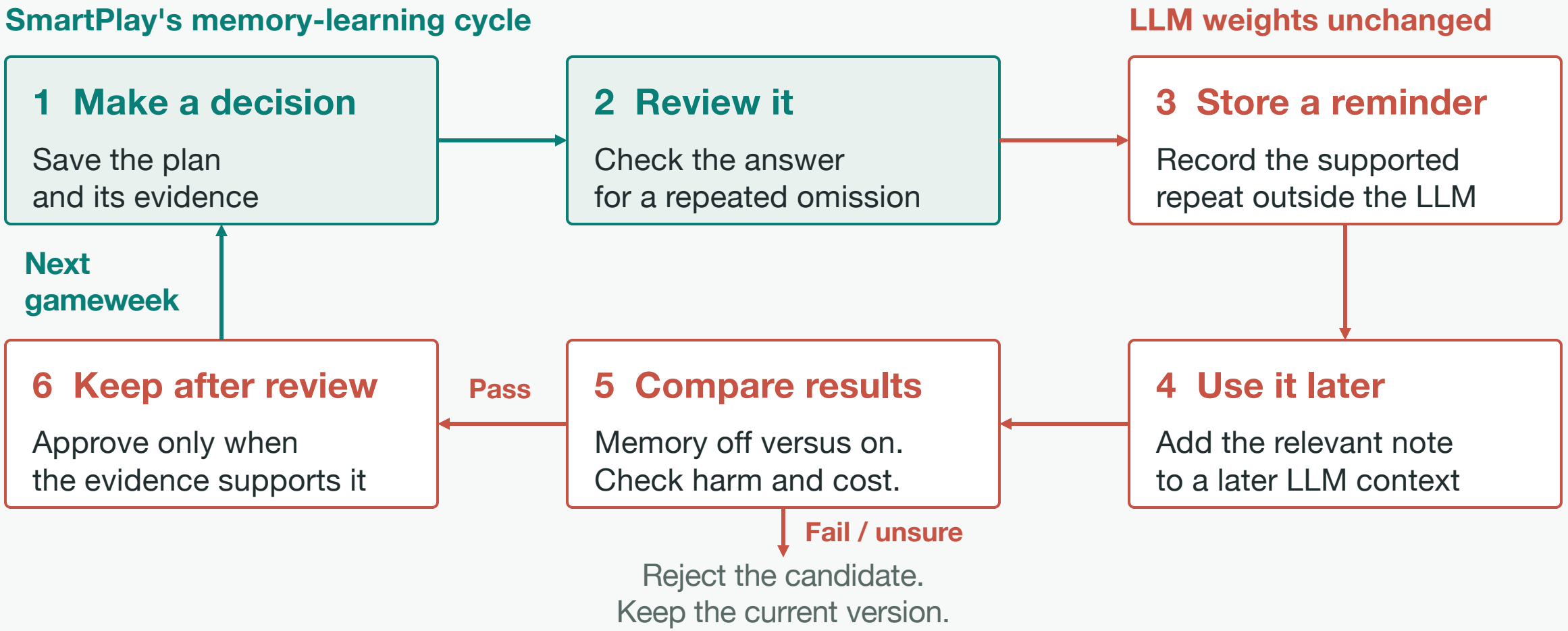
Retention test

Later forecasts and held-out earlier cases.

Replay changes training. Prompt memory changes the input.

Research: Rolnick et al. (2019), NeurIPS.

How SmartPlay learns



Every step of this cycle is built into SmartPlay.

Why we expect SmartPlay to improve

Reasons for the expectation

Evidence	Every gameweek adds a dated, scored record.
Memory	A mistake seen in two reviews becomes a note for later decisions.
Forecast	Retrained on new results, with older examples replayed.

Only changes that pass the test are kept.

Built into SmartPlay: process memory, forecast retraining and the approval test.

The world changed overnight. The maps kept up.

1 Change Lockdowns cut traffic by up to 50%, worldwide.

2 Evidence Live traffic stopped matching the usual weeks.

3 Location Which past the model trusts: the last two to four weeks.

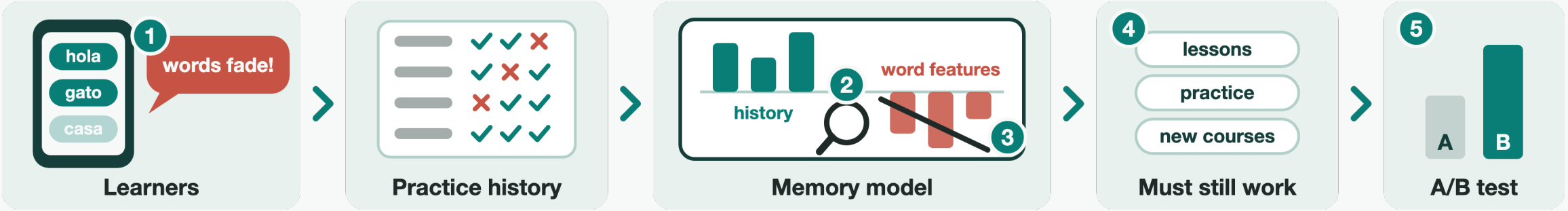
4 Retention Older weeks kept, just weighted less.

5 Evaluation Every real arrival checks the prediction.

Arrival times accurate on over 97% of trips.

Source: Google, “Google Maps 101: How AI helps predict traffic and determine routes,” 3 September 2020.

Learners spoke up. The fix passed a fair test.



1 Change Learners said some words faded fast, despite practice.

2 Evidence Traced to the model: some word features had overfit.

3 Location The model itself: drop those features, keep practice history.

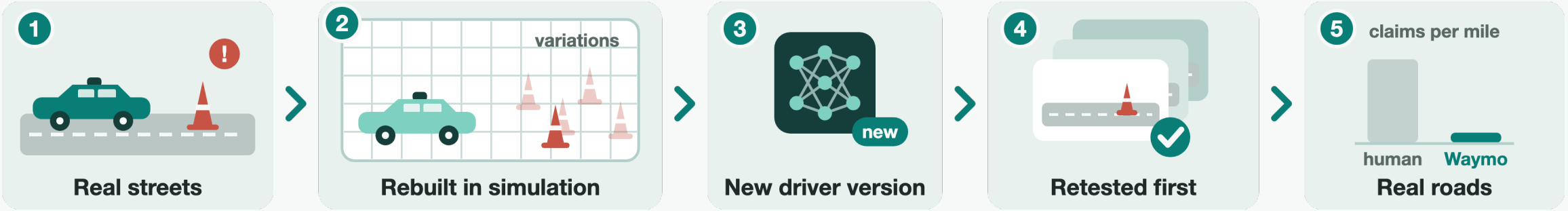
4 Retention Lessons and practice, even in new courses with no data.

5 Evaluation A fair test: 3.3 million learners, split at random.

Next-day return up 12%. The fix went to every learner.

Source: B. Settles and B. Meeder (Duolingo), “A Trainable Spaced Repetition Model for Language Learning,” ACL 2016.

Every hard moment on the road becomes a test



1 Change Rare events and new streets the cars haven't met.

2 Evidence Each hard moment rebuilt in simulation, in many variations.

3 Location The driver itself: new models, trained mostly in simulation.

4 Retention Every change goes through rigorous tests before it ships.

5 Evaluation About 20 million simulated miles a day, then real roads.

Swiss Re: 88% fewer property-damage and 92% fewer injury claims than humans.

Sources: Waymo blog, 28 April 2020 and 6 July 2021; Waymo and Swiss Re, 19 December 2024.

And in the systems you build

Agents that read logs, tickets and runbooks, then act

The model may

Propose a fix.
Summarise an incident.
Draft a runbook step.

Only code and people may

Restart production.
Change a limit.
Promote a lesson into policy.

A log line can inform an action. It cannot authorize it.

One incident is a record. Two are a lesson. A green deploy is a claim. The readback is the evidence.

What stops the same mistake next week?

The opening snapshot: five gameweeks

299

SmartPlay's team

299

Average manager

Progress needs evidence from later decisions.

Next claim to test: earlier experience reduces repeated mistakes.

Questions

What could your system
learn from its next decision?

Qazybek Beken