DATA SATURDAYS HOLLAND

Cognitive Shift: How Agentic AI Changes the Way We Think



Thijs Baan

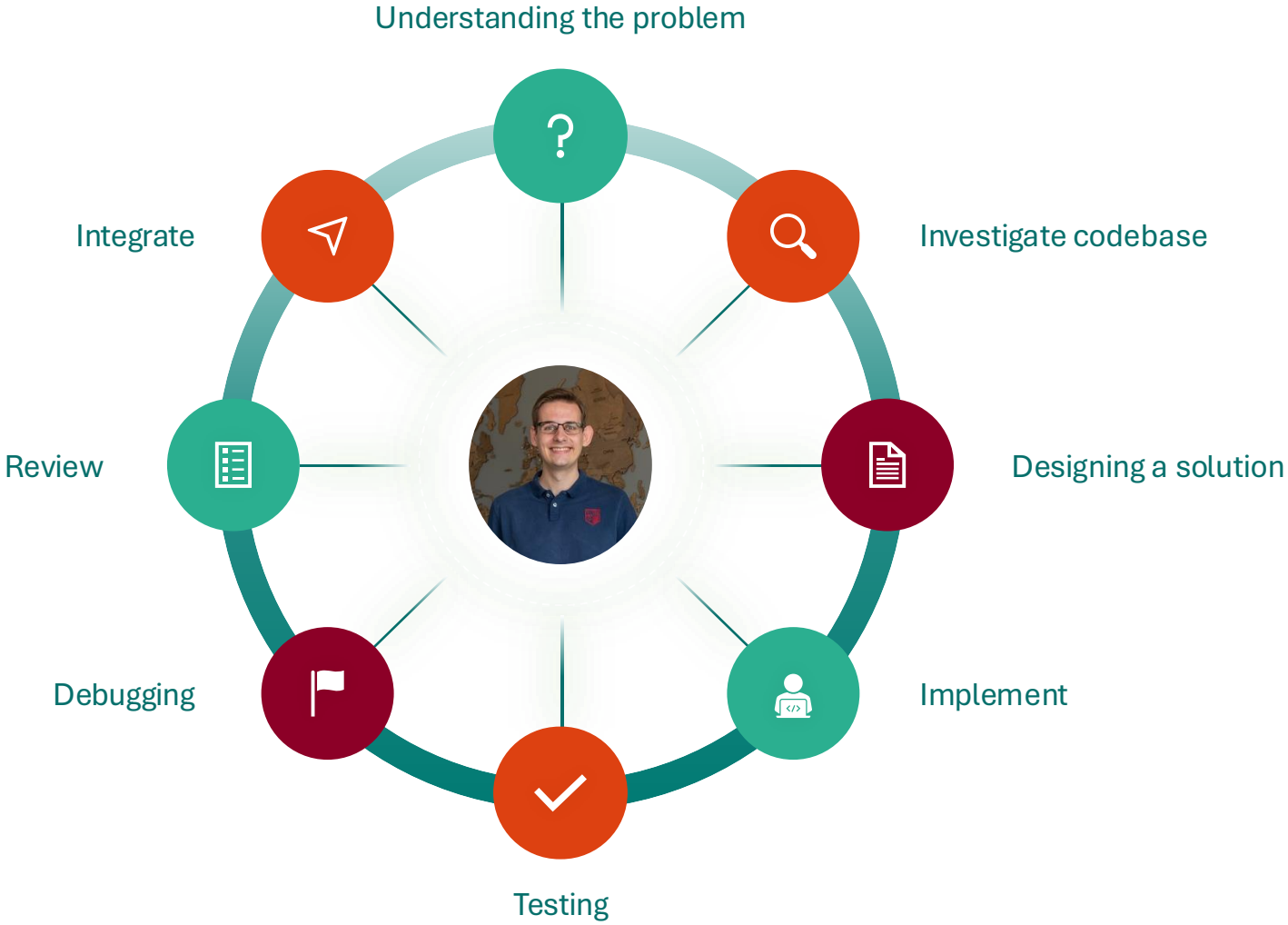
Thijs Baan



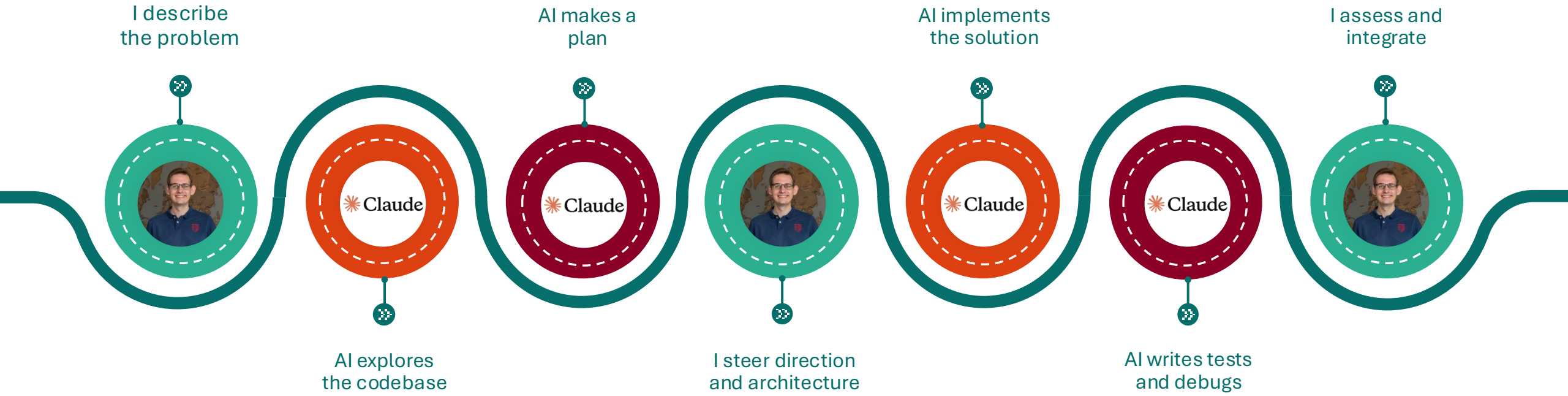
- Ede
- Info Support B.V.

- Lead Engineer
- Applied Psychologist

2021: my development workflow

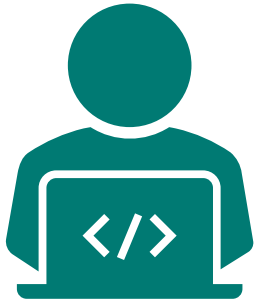


2026: my current AI workflow



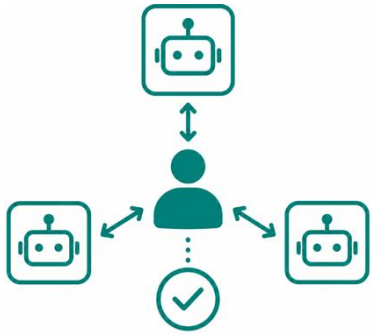
My work is shifting

ACTOR





DIRECTOR



Other thinking



More direct implementation

- Actively explore codebase
- Forming and testing hypotheses
- Go through implementation steps
- Local debugging

Cognitive centre of gravity shifts



More control

- Formulating intentions and frameworks
- Reasoning at the architectural level
- Reconstructing context
- Assessing output
- Adjusting and integrating

DATA SATURDAYS HOLLAND

London



The Knowledge



Admission for London black-cab drivers

- London by heart
- Determine your own routes
- Reproduce routes verbally

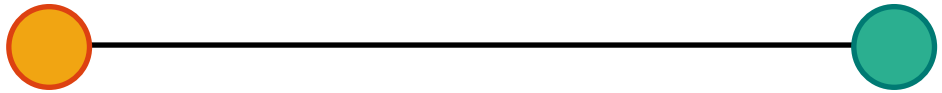
This learning process usually takes several years.

Practicing the same thinking for years



What changed in successful trainees?

Trainees through time



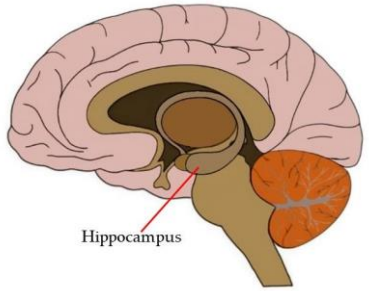
Measurement at the start

Measurement years later

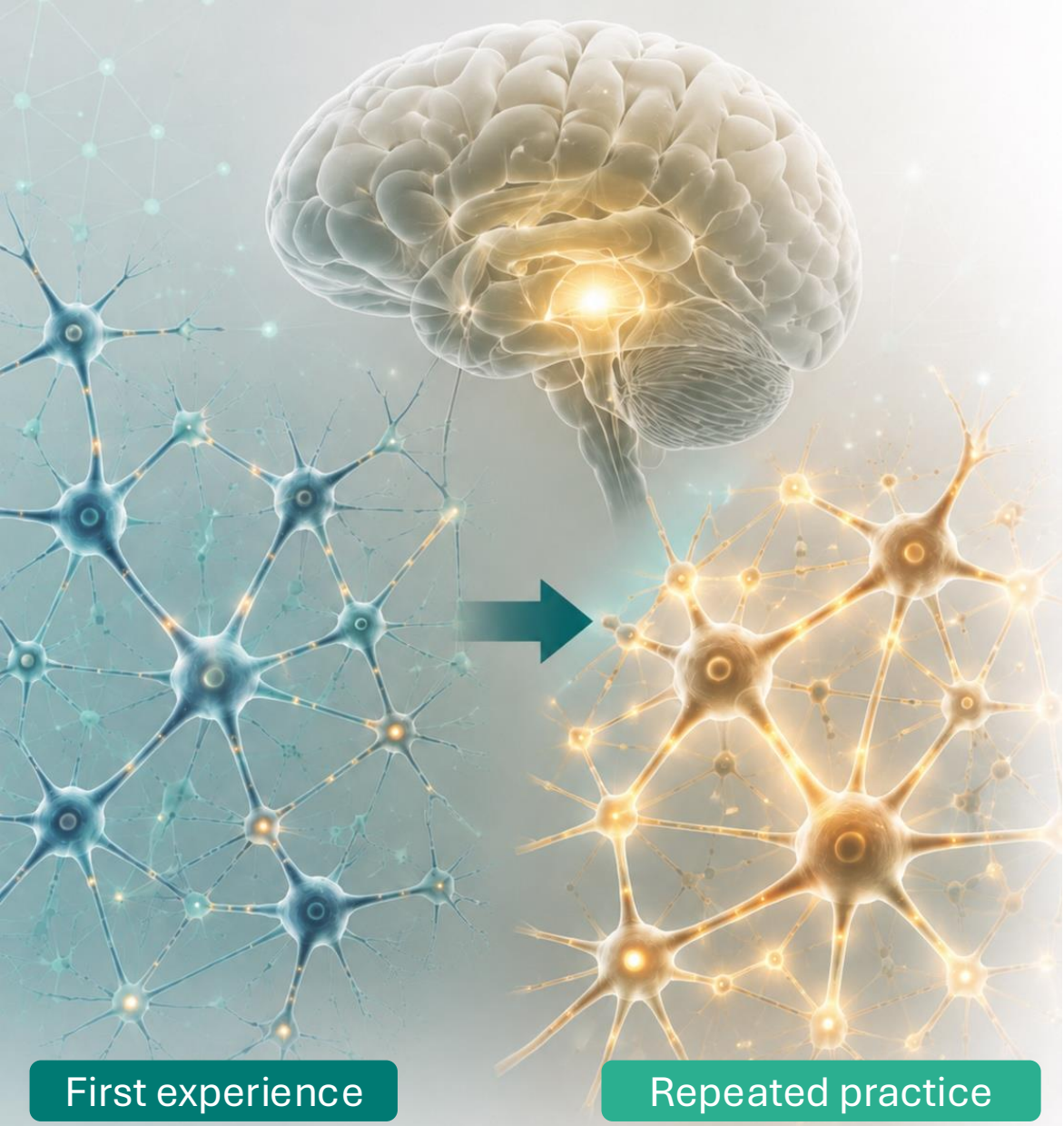
- No clear group differences at the start;
- Successful trainees developed more London-specific navigation knowledge;
- They used London as a cohesive route network.

And on the MRI?

Hippocampus: important for memory and spatial navigation



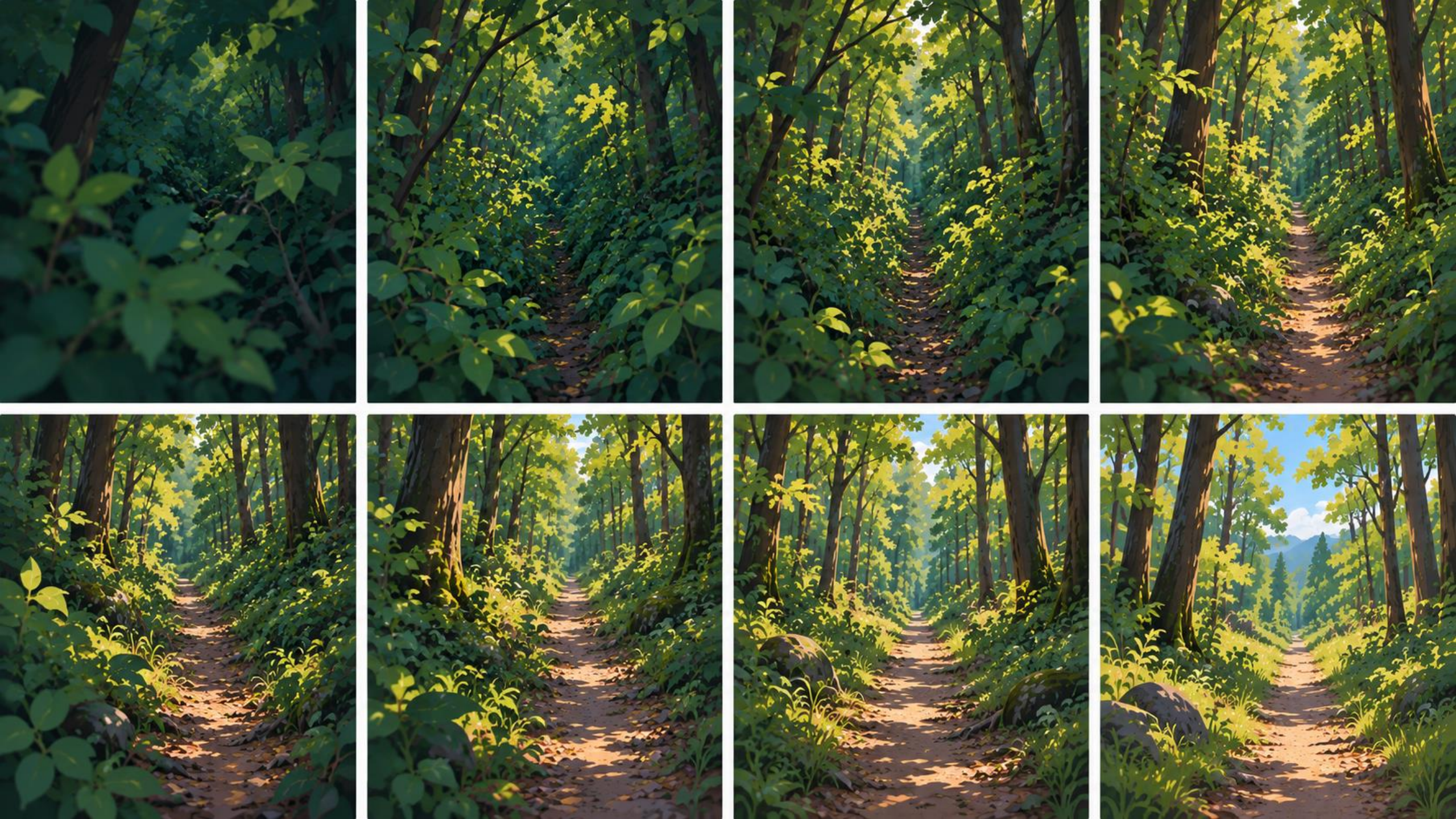
Successful trainees later showed measurable **growth in the hippocampus .**



Practice changes the brain

Neuroplasticity: Repetitive experience and practice can alter the functioning and organisation of neural connections and networks.

What we repeat **matters**

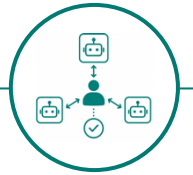


Back to engineering



The Knowledge

Years of specific practice formed specialised navigation expertise.



Agentic Engineering

As my daily thinking changes, so does which skills I practice most often.

DATA SATURDAYS HOLLAND

Outsourcing cognitive work



You already do this all day

Remember

A reminder keeps the intention for me.

Calculate

A calculator performs the operation for me.

Navigate

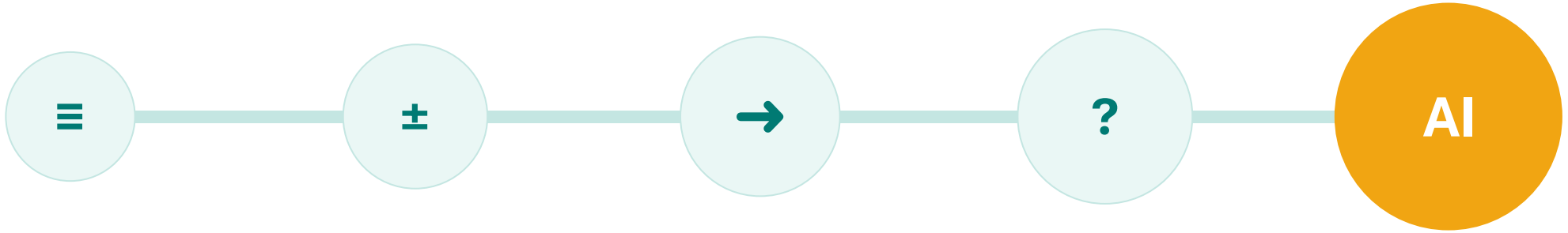
GPS keeps the route outside my head.

Cognitive offloading

Using the environment to change the cognitive demands of a task.

Without a reminder	Remember	→	Monitor	→	Recall	→	Act
With a reminder	Set cue	→	External signal	→	Interpret	→	Act

Agents change the scope



Reminder
One cue

Calculator
One operation

GPS
A route

Search
Retrieve context

Agent
Multi-step route

What happened between prompt and output?

My prompt
"Investigate where this behaviour comes from, make a plan and implement the change."

UNDERSTAND

- Search codebase
- Looking up relevant information
- Investigate the cause

CHOOSE

- Hypothesis forms
- Weighing up options
- Make a plan

EXECUTE

- Change code
- Writing Tests

Result

- Working code
- Tests
- Choices made

So what still needs to remain with me?

FRAMING

Can I tell...

if the problem was framed correctly?

IMPACT

Can I see...

what else this change affects?

EVIDENCE

Can I judge...

whether the evidence is enough?



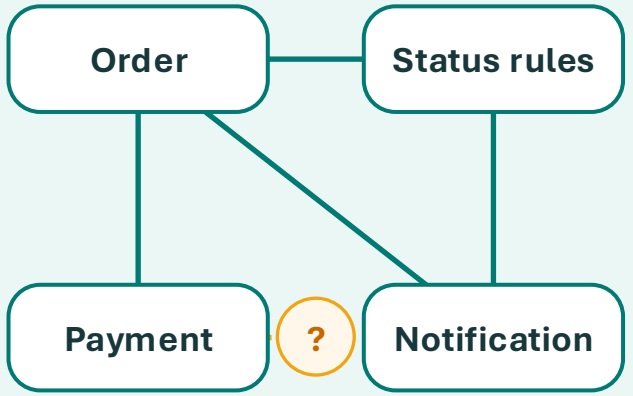
What is a mental model?

THE SYSTEM

OrderService.cs | status enum | retry policy | payment webhook | event schema | logs | API contract | DB migration | UI form | feature flag | unit tests | config | auth middleware | email template | legacy job | README | queue | invoice PDF



MY MENTAL MODEL



relationships · assumptions

Explain
how it works

Predict
what a change does

Challenge
whether it holds up

Building understanding

1. Explore
2. Hypothesis
3. Design
4. Implementation
5. Test

My mental model is growing

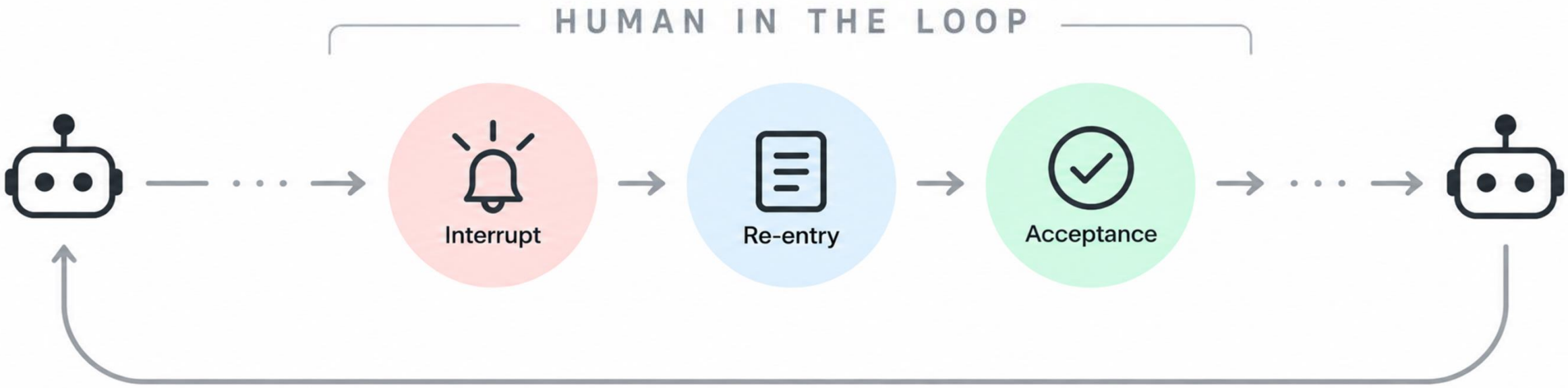
- Texture
- Relationships
- Assumptions
- Consequences

If the agent is more of that route goes through



Output can progress faster than my understanding.

Human thinking in action



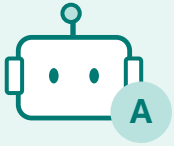
DATA SATURDAYS HOLLAND

Interrupt



Execution can run in parallel

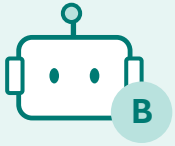
One feature: a new order status. Three workstreams.



Agent A
Order Service

Implements the state transition and domain rules

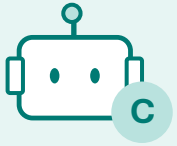
Works independently



Agent B
Payment Service

Updates the corresponding payment flow

Works independently



Agent C
Notification Service

Adds event and notification logic

Works independently

And then suddenly they all need me…

Agent A
Order Service

"Is this state transition acceptable?"

Agent B
Payment Service

"Can I change this contract?"

Agent C
Notification Service

"Is this event boundary correct?"

My attention needed

My attention needed

My attention needed



My attention

Judgement needs an active task context

FOREGROUND

What do I need active to judge this question?

- Goal
- Rules
- Current state
- Criteria

AVAILABLE, NOT ACTIVE

- Other code
- Other agent logs
- Other decisions

Working memory



A limited mental workspace for information that is active and usable right now.

So how limited are we talking?

Let’s try something

1 Watch

A row of digits appears for less than two seconds.

2 Remember

Keep as many as you can in your head. No writing, no phones.

3 Recall

How many can you still reconstruct?



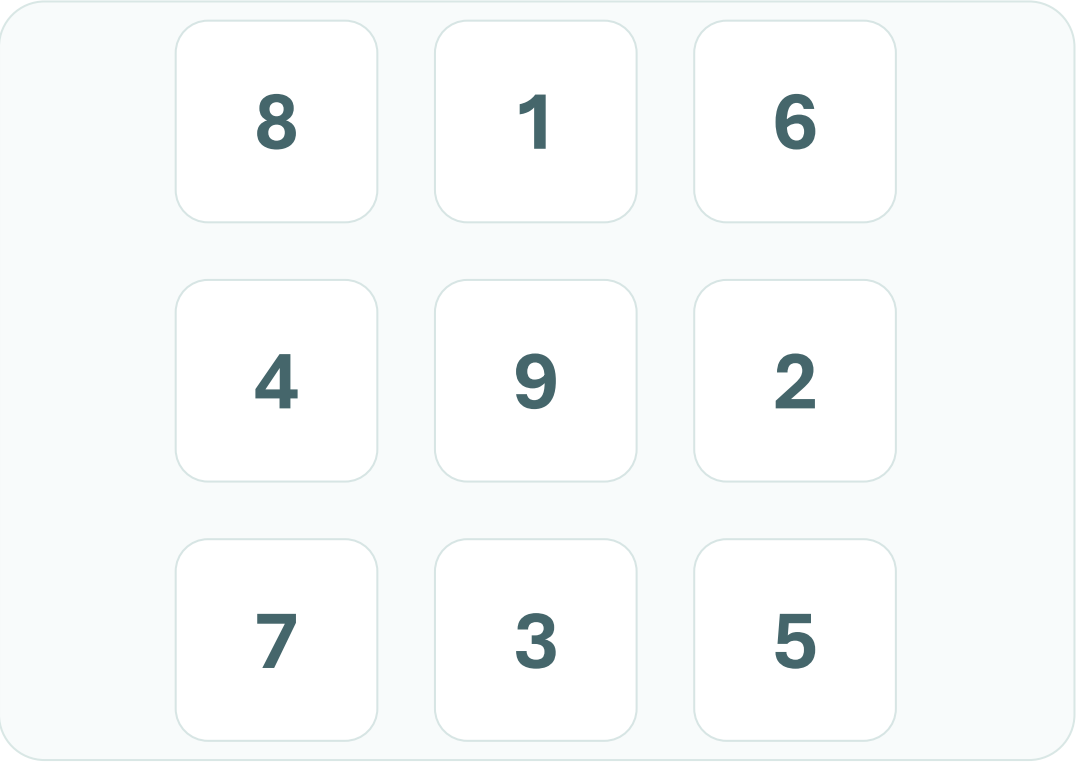
Ready?

8 1 6 4 9 2 7 3 5

What do you remember?

How many can you reconstruct?

Only a few chunks at once



≈ 3–4

Capacity estimates depend on the task, prior knowledge and how information can be organised.

Let’s try that again

Ready?

2026

404

007

And now?

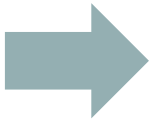
This time there were more digits.

But we don’t work with isolated facts

Separate elements

2 0 2 6 4 0 4 0 0 7

ten separate things to hold



Meaningful units

2026 404 007

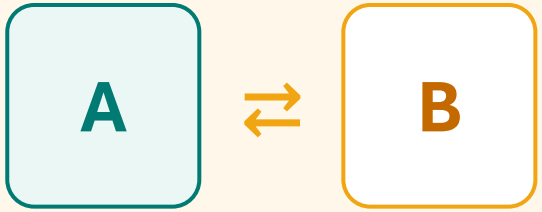
three familiar patterns

Knowledge and structure help us compress complexity.

Back to the agents



Chunking helps me handle complexity **within one context.**



An interrupt can make me activate a **different context.**

Task switching

Task A · Order Service

- Goal A
- Rules & invariants A
- Current state A
- Acceptance criteria A

SWITCH

Let go of A
Activate B

Task B · Payment Service

- Goal B
- Rules & invariants B
- Current state B
- Acceptance criteria B

Switch cost

After a switch, controlled tasks are typically slower and often more error-prone than after repeating the same task.



Parallelism is not the problem

Agent A
Order Service

Implements the new order status

Agent B
Payment Service

Processes the corresponding Payment status

Agent C
Notification Service

Adds event and notification logic

My attention needed

My attention needed

My attention needed



My attention

I have to prevent three agents from needing me at the same time.

Quiet execution, loud exceptions

QUIET EXECUTION

Keep running

- Run tests
- Format / lint
- Search code or docs
- Read issues, PRs and work items (MCP)
- Apply an agreed pattern

No judgement needed yet

LOUD EXCEPTIONS

Interrupt me

- Hard or impossible to reverse
- Large blast radius
- Public API / contract change
- New dependency
- Security or privacy boundary

Human judgement changes the path

Interrupt reduces the number of switches: not better juggling, but less frequent switching.

One loud exception, the rest runs quietly

Agent A
Order Service

Implements the new order status

Works independently

Agent B
Payment Service

Processes the corresponding Payment status

My attention needed

Agent C
Notification Service

Adds event and notification logic

Works independently



My attention

Interrupt: let the execution continue until my judgement really changes something.

Give the agent a safe playing field

Isolation
sandbox / container / temporary environment



Dev containers

Outside the agreed playing field
→ human judgement

Agent works autonomously

Scope
Restricted data, network and tool access



Hooks

Least privilege
only necessary credentials and rights

Recoverability
version control · reversible actions · automated checks

DATA SATURDAYS HOLLAND

Re-Entry



Re-entry existed before AI

Classic

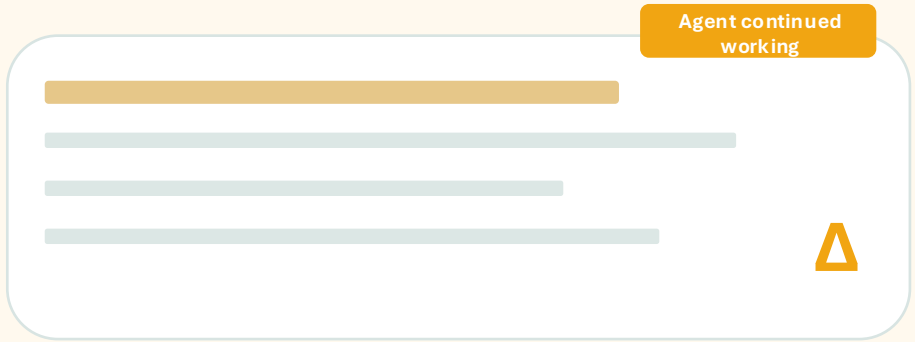
I return to roughly the same task.

I'll be back

Where was I?

Agentic

The task may have continued while I was away.



I'll be back

What changed?

Attention restored ≠ task context restored



Perform task

Purpose and context active

Interruption

Active context fades

Back at the task

My attention is back

Resume task

I know again what is relevant

Task resumption

- Purpose
- Current status
- Relevant context
- Next step

Context back is not yet understanding

Context back

What happened?

- summary
- diff
- Tests & Evidence
- Decisions
- Current status

INTEGRATE

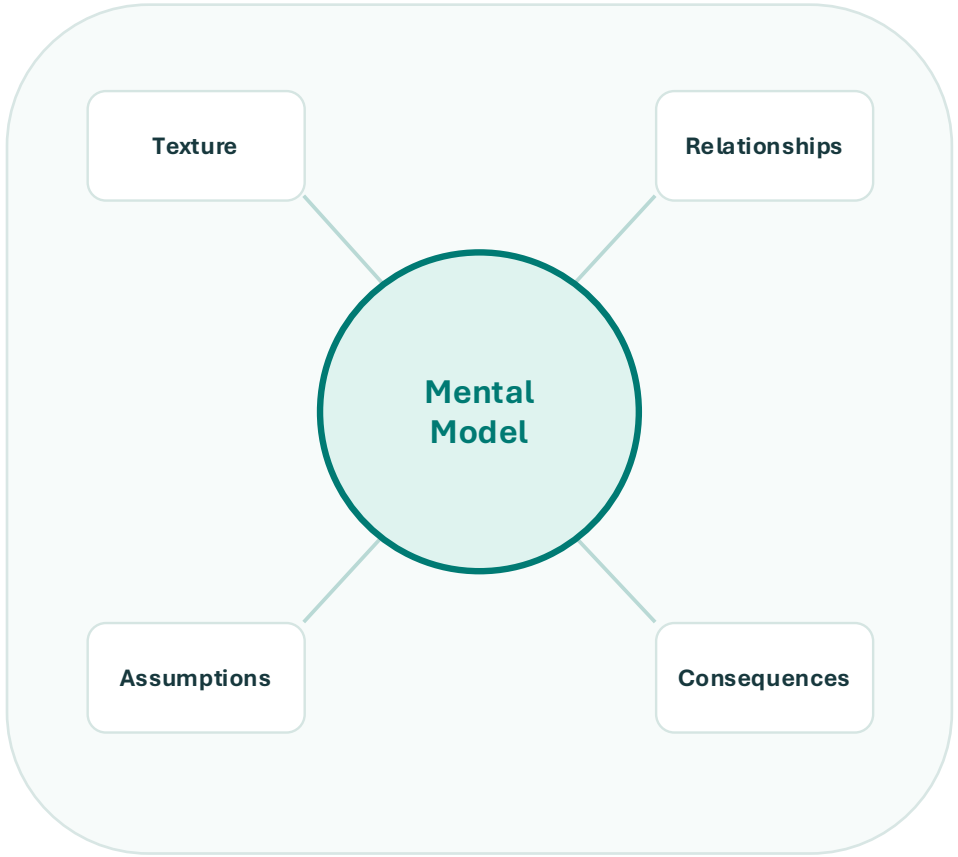
Understanding back

What does that mean?

- Why did this change?
- What does this affect?
- What consequences do I expect?
- What do I need to decide now?

Restoring mental model

Re-entry makes my mental model usable again



Don't reconstruct everything.
Just enough to reason with again.



Re-entry is all about the delta

LAST CHECKPOINT

10:03

I knew enough to release the agent.

When I return, I mainly need to know what has changed since that moment.

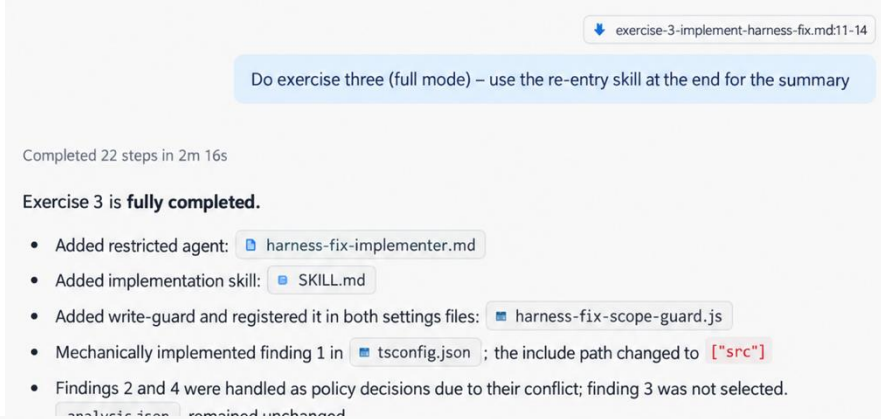
Δ SINCE CHECKPOINT

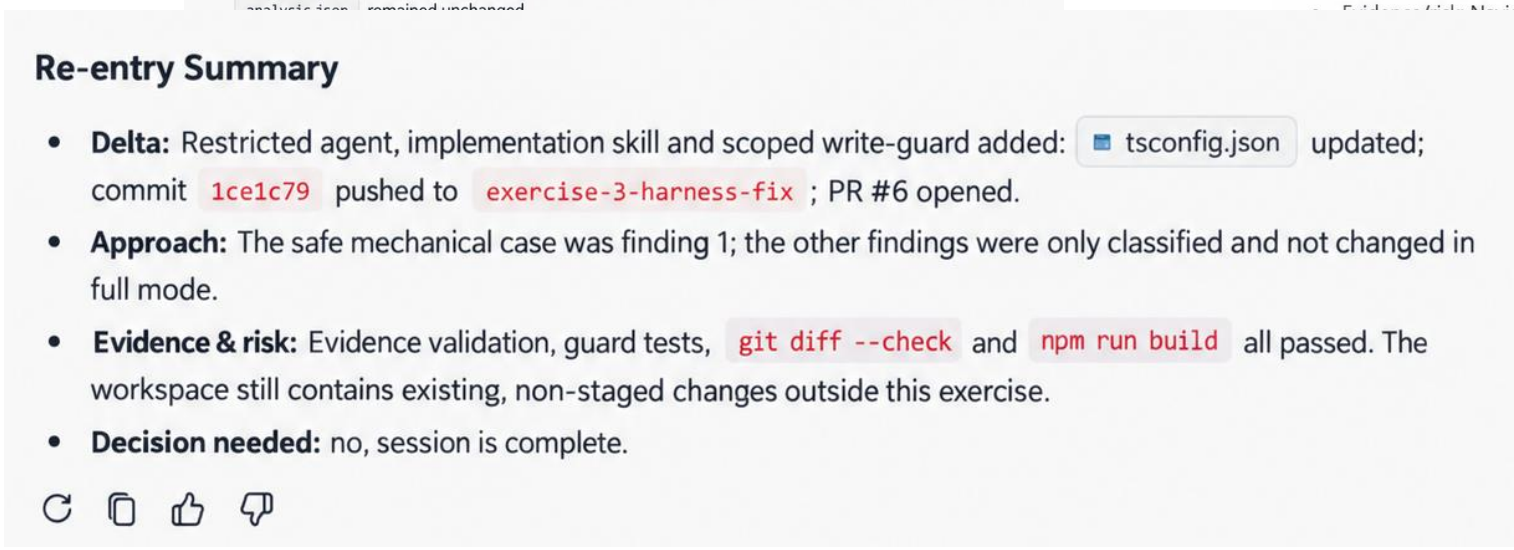
What do I need to update?

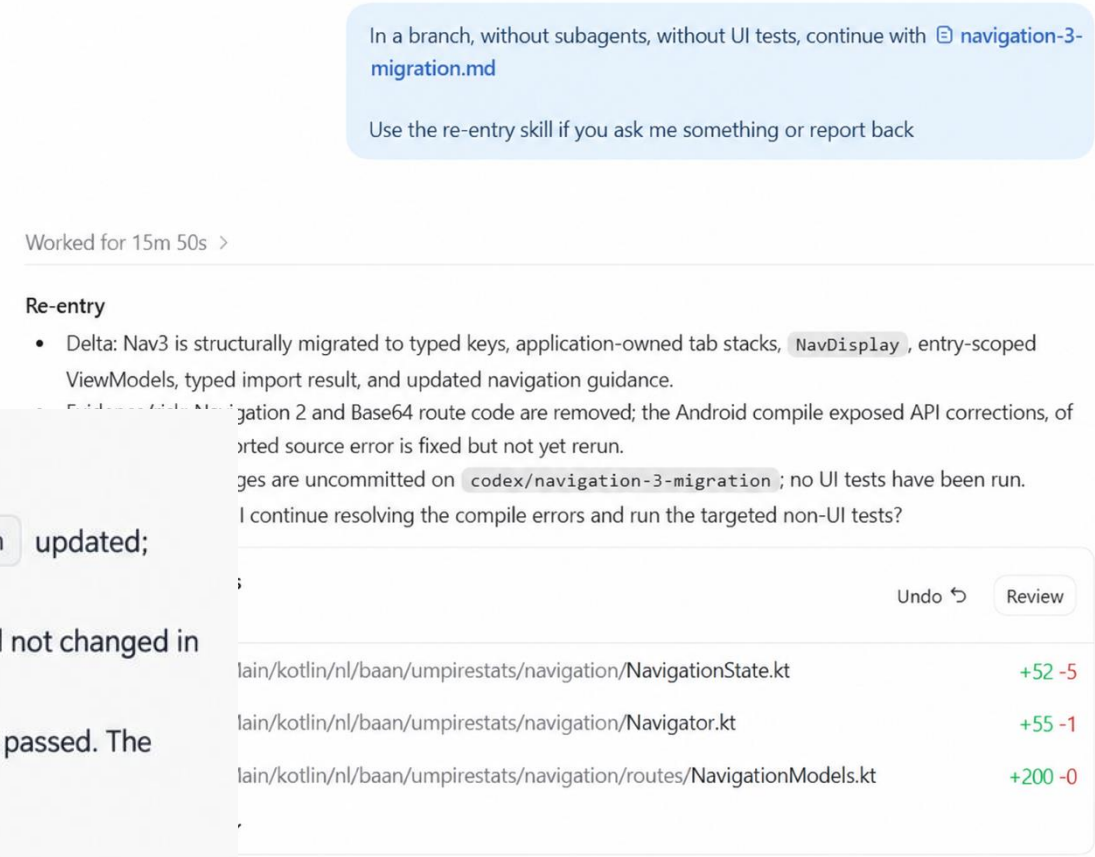
CHANGED	**What is really different?**	code · Structure · Status
ASSUMPTION	**Which assumption changed or proved incorrect?**	What I previously believed to be true
EVIDENCE / RISK	**What do we know now that we didn't know before?**	Tests · observations · new uncertainty
DECISION	**Where is my judgement needed now?**	The next human choice

RE-ENTRY SKILL

Re-entry into my AI session







Re-entry has two time horizons

MINUTES · HOURS · DAYS

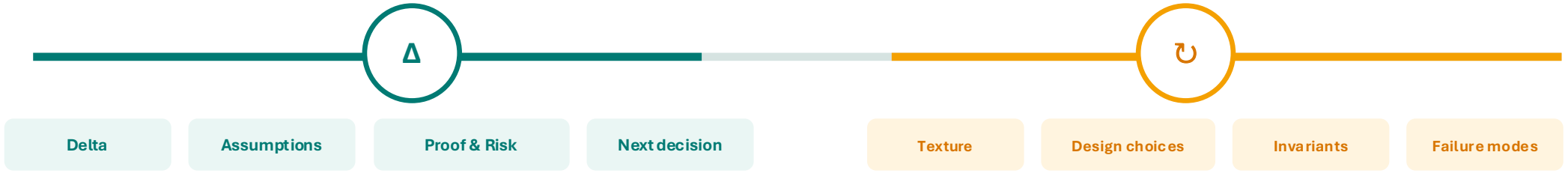
NOW

Back to work

WEEKS · MONTHS · LATER

LATER

Back in the system



Checkpoint

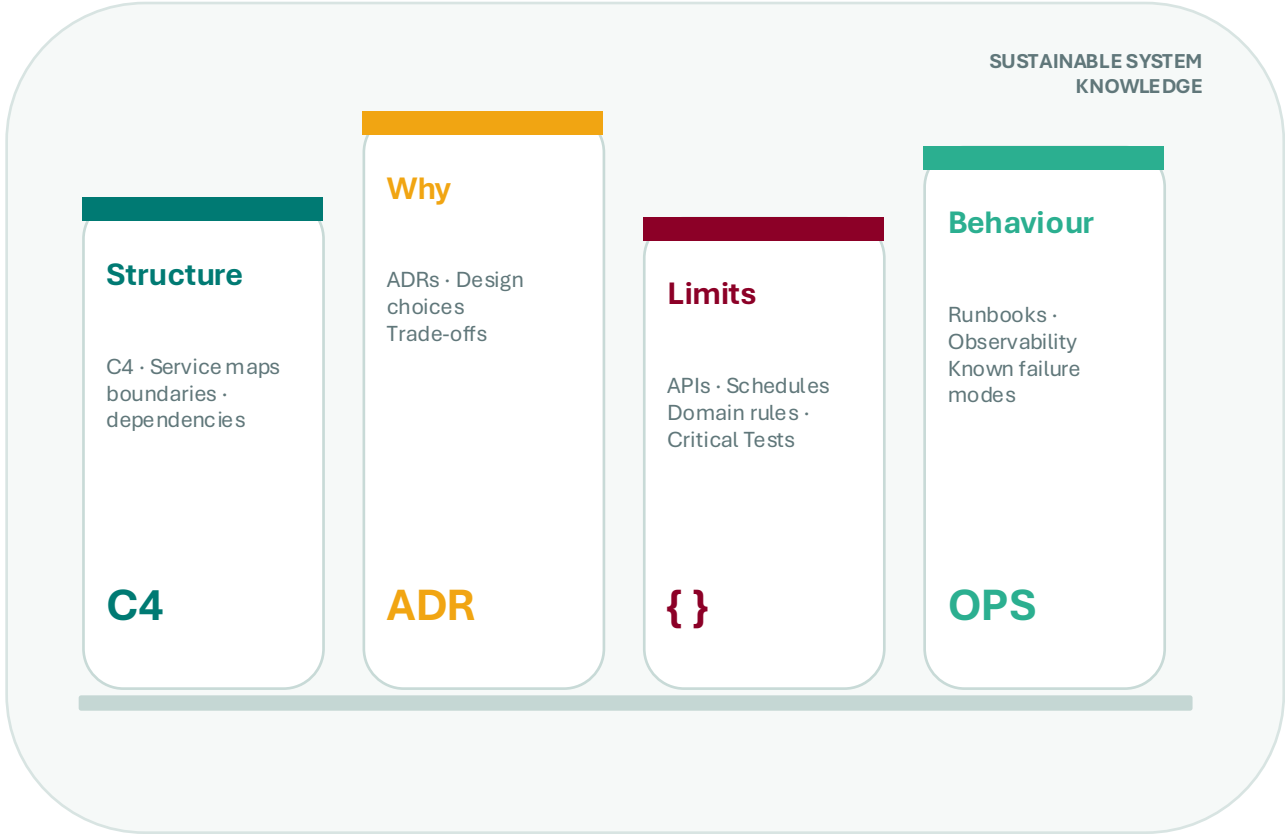
Helps me get back to work.

Sustainable context

Helps me build a usable model later on.

Enable long-term re-entry

Not everything has to stay in my head.
Important knowledge must be retrievable.



DATA SATURDAYS HOLLAND

Acceptance



Execution ready. Decision begins.

AGENTIC EXECUTION

"Done."

- **48 tests green**
- Formatter & Linter Clean
- Implementation summary generated
- No known errors

COGNITIVE SHIFT

HUMAN ACCEPTANCE

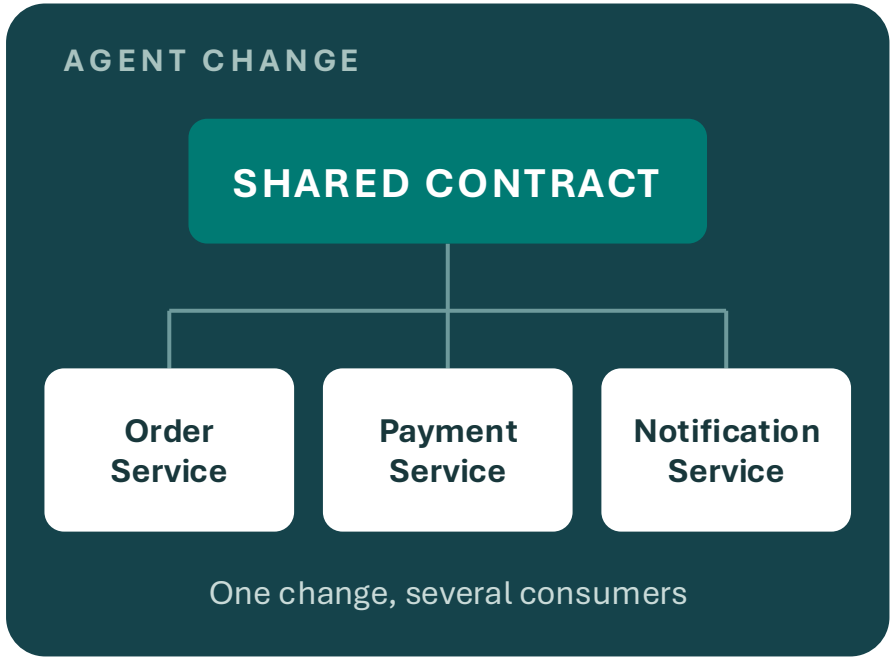
Am I ready for a 'yes'?

MERGE?

I did not build every part of this.

What do I still need to understand?

What do I need to understand?



1. **Who depends on this?**
   Which components or consumers are affected?

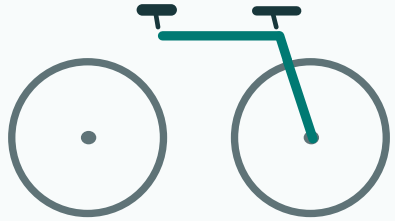
2. **What happens to an older consumer?**
   Can I predict the relevant downstream behaviour?

3. **What must remain true?**
   Which assumption carries this decision?

You know how a bicycle works, right?

BEFORE THE TEST

“Of course I know how a bicycle works.”

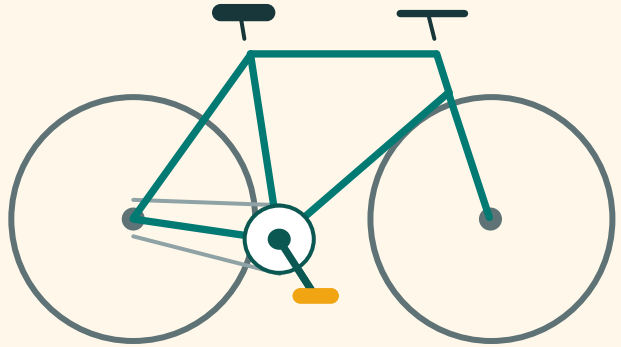


How well do I understand it?

1 2 3 4 5 **6** 7

NOW DRAW IT

Where exactly do the frame and chain go?



Familiarity can hide gaps in our mental model.

Make understanding observable

EXPLAIN

Why does this work?

Without repeating the agent's summary.

SHARED CONTRACT

"Old consumers can still parse the payload, because the new field is optional..."

PREDICT

What happens if an assumption fails?

Name the relevant consequence.

SHARED CONTRACT

"An older consumer ignores the field, so its behaviour stays the same..."

CHALLENGE

What would make me reject it?

Name the condition that changes my decision.

SHARED CONTRACT

"Unsafe if any consumer treats unknown fields as invalid."

How expensive is a wrong 'yes'?

IMPACT

What can break?

From local behaviour to users, data, security and multiple systems.

Wider consequences → more depth

RECOVERABILITY

How easily can we go back?

From simple revert to migration, external contract or side effect that is difficult to reverse.

Harder recovery → more depth

DETECTABILITY

How quickly will we notice?

From a failing test to downstream behaviour, silent corruption, or a production issue.

Later detection → more depth

The more a wrong “yes” costs, the deeper I go.

Decide what matters up front

BEFORE / EARLY IN THE RUN

My frame

- Desired behaviour
- Architecture boundaries
- Invariants
- Main risks
- Expected evidence

FOCUS THE REVIEW

WHEN THE AGENT IS DONE

My questions

- Can I explain the important mechanism?
- Can I predict the risky consequence?
- Can I challenge the critical assumption?
- Does the evidence match my frame?

Before you say yes: explain, predict, challenge.

As deep as a wrong “yes” is expensive.

DATA SATURDAYS HOLLAND

The core



Design for the pressure points

INTERRUPT

UNDER PRESSURE

Execution scales. My attention doesn't.

DESIGN

Minimise interrupts

Quiet routine. Loud exceptions.

→ **A safe playing field**

RE-ENTRY

UNDER PRESSURE

The work continued. My model didn't update.

DESIGN

Make context recoverable

Checkpoint, then restore the delta.

→ **The re-entry skill**

ACCEPTANCE

UNDER PRESSURE

Green signals ≠ a responsible "yes".

DESIGN

Review against a frame

Decide up front what matters.

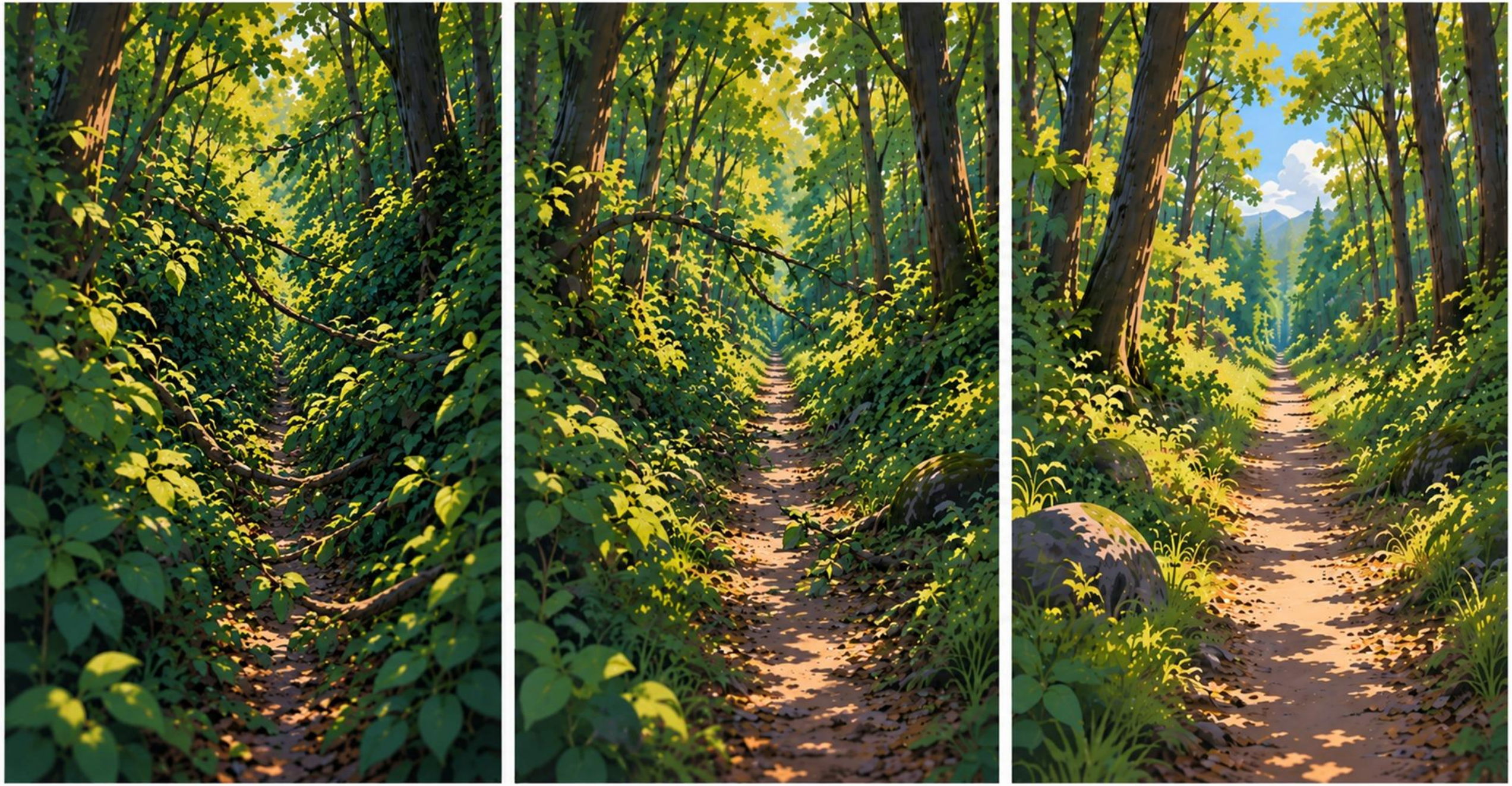
→ **Explain · predict · challenge**

ONE MORE QUESTION

If AI does more of the work...

won't I lose the skill?





DATA SATURDAYS HOLLAND

The Cognitive Shift



Thank you sponsors!

INSPARK

twintos

ilionx

InfoSupport
Solid Innovator

redgate

sopra steria

DATA GIANTS

BRANDER
company

SQL brander.nl
DATA brander.nl
CRM brander.nl
ERP brander.nl

Bull
Master your AI destiny

dbWatch

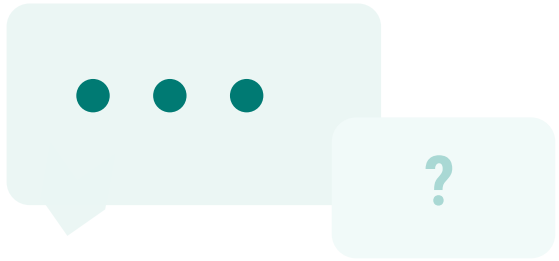
BBTG

Data Saturdays

Questions?

Thank you for your attention!

Do you have any questions later? Feel free to ask later.



More from me



Behavioral Engineer

behavioralengineer.substack.com

LinkedIn

linkedin.com/in/thijs-baan

Resources

Re-Entry Skill

Feedback

Scan to give feedback

Your feedback helps me improve!

Thank you sponsors!

INSPARK

twintos

ilionx

InfoSupport
Solid Innovator

redgate

sopra steria

DATA GIANTS

BRANDER
company

SQL brander.nl
DATA brander.nl
CRM brander.nl
ERP brander.nl

Bull
Master your AI destiny

dbWatch

BBTG

Data Saturdays