DDD OUTBACK 2026

I Didn’t Hire a Team, I Cloned Myself

Lessons from a year of AI-assisted development

Gert Jansen van Rensburg · Software Consultant

THE SURPRISE

AI didn’t make me a faster typist. It made me responsible for more work at once.

The role shifted from implementation toward orchestration. One year of it, on enterprise Azure and React delivery, across three time zones.

WHAT YOU WILL GET

This is a field report, not a tooling demo

- How the workflow evolved from snippets to specialised agents
- What genuinely improved delivery, and what I stopped doing
- Why the bottleneck moved into verification and human systems
- Where engineering judgment became more valuable

ACT 1 · EVOLUTION

The workflow evolved through small experiments

AT FIRST

Snippets

One clear problem.
One answer.
Manual integration.

THEN

Specifications

The agent asked questions.
We agreed the plan before code.

LATER

Parallel agents

Implementation, review and testing at once.
Then they collided over ports, data and context.

TODAY

Orchestration

I manage context, queues, quality gates and decisions.

THE FIRST HARD LESSON

Fast code didn’t preserve the vision

4

abandoned repositories

A weekend reminders app became two weeks of inconsistent UI, half-built features and repeated restarts.

The tools could execute. They couldn’t hold the product vision without structure.

THE BREAKTHROUGH

Structure beat tooling

Guardrails, a shared specification and explicit quality gates kept the work coherent.

Once the workflow became repeatable, changing the agent mattered less than changing the process.

The delivery loop became explicit

01 **Clarify the story**

02 Make a plan

03 Implement

04 Review

05 Simplify

06 **Verify the acceptance criteria**

07 Ask what was missed

The main thread became the team lead

One place holds the story.

The goal, decisions, plan, progress, risks and remaining human judgment stay together.

Architecture inspection

Implementation

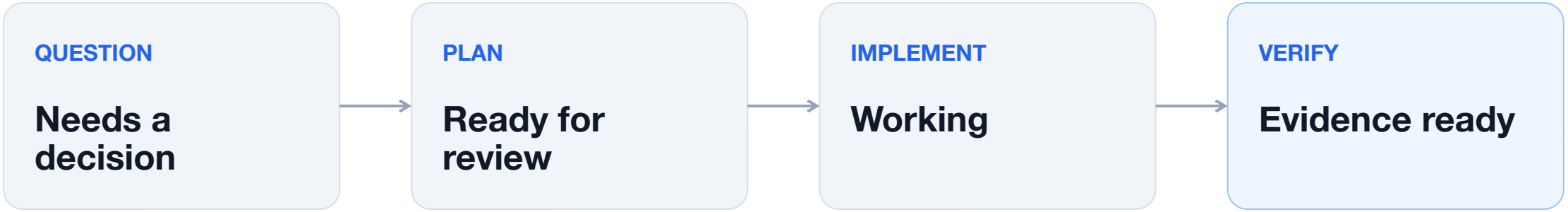
Code review

Tests

Visual verification

THE SHAPE OF THE DAY

It stopped feeling like a loop. It felt like a queue.



My attention moved to whichever item needed judgment next.

My scarce resource became attention

DECIDE	**Option A or B?**	Unblock the next step
REVIEW	**Does the plan fit?**	Prevent expensive drift
VERIFY	**Is the evidence convincing?**	Choose another iteration or finish

A TUESDAY MORNING

What was waiting for me at 9:40am

DECIDE	**Who owns this data?**	Pull in the solution architect
REVIEW	**Integration, fourteen files**	Read the diff, not the summary
VERIFY	**Two overnight checks failed**	Iterate, or hand it to QA

None of it was code I wrote. All of it was mine to answer.

A LOW-RISK HABIT

The best experiments were asymmetric bets

**“If it works, I win time.
If it fails, I’m where I started.”**

Plenty failed. “What did we miss?” invented risks for an hour. A simplification pass rewrote working code. Each cost one evening.

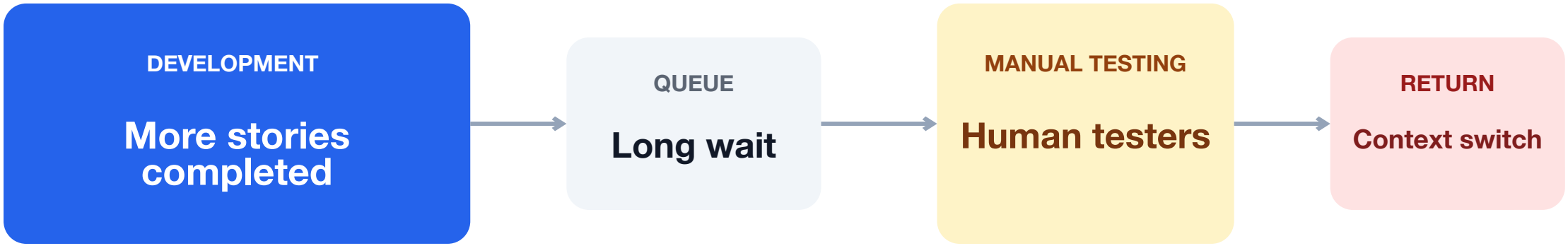
EVENING

Delegate acceptance checks across a large story.

MORNING

Review the evidence, failures and next decisions.

Speed moved the constraint downstream



The system didn’t become infinitely fast. The pressure moved.

WORKING WITH THE REAL CONSTRAINT

We adapted the hand-off, not the accountability

PREPARE A LARGER SLICE

Verify around the epic

Agents assemble evidence across related stories before the hand-off.

KEEP THE HUMAN GATE

Manual verification remains

The team chooses its process and owns the final result.

FROM A CLIENT AI POLICY

What human oversight actually checks

Accuracy	Facts, figures and references are correct
Quality	Meets professional standards, properly referenced
Tone	Suits the intended audience and the context
Bias	Present in the input as well as the output
Cybersecurity	Generated code checked for vulnerabilities

SKILLS STILL MATTER

AI is an amplifier, not just an enabler

WHICH WAY IT AMPLIFIES

It multiplies whatever you bring

IT AMPLIFIES SKILL	System knowledge and judgment compound.
IT AMPLIFIES GAPS	Unclear intent now ships faster, and wronger.
WHAT SPED ME UP MOST	Clear intent, structured stories, actionable acceptance criteria.

The middle got cheaper. The edges got expensive.

MORE VALUABLE	Human intent: architecture and real constraints.
CHEAPER EVERY MONTH	Agent execution: draft, implement, test, review.
MORE VALUABLE	Human judgment: does it belong? Are we done?

Scan for the original post
Damian Maclennan · LinkedIn

WHAT I ABANDONED

A few habits didn’t survive the year

Jumping straight to code	Fast output amplified unclear intent
Letting every agent solve everything	Specialisation reduced noise
Calling it done when tests passed	Green build, wrong outcome
Running “what did we miss?” forever	Judgment still has to end the loop

THE ROLE SHIFT

What I do all day now

Break work apart

Make the next decision small

Feed context

Keep the goal and constraints visible

Compare

Ask for alternatives, not certainty

Review

Inspect code, design and evidence

Test

Exercise the paths that matter

Decide

Own the trade-off and the outcome

WHAT THIS MEANS FOR YOU

This isn’t a story about my tools

IF YOU WRITE CODE	Leverage moves to how clearly you state intent.
IF YOU LEAD A TEAM	Your constraint is about to move. Find it first.
IF YOU TEST	The hand-off changes. Accountability doesn’t.

AI didn’t make me 10× faster.
It gave me a team.
I just didn’t realise
I’d become their manager.

Start with one real task:

Make the plan explicit
Add one independent review
Keep the final decision