RESEARCH INTERESTS

Closing the gap between “looks done” and “works”: measurable reliability for agentic and data-centric AI systems

Jiawei Zhang · Engineering lead for an AI data platform and its business-agent runtime · M.S. Computer and Information Science, Marquette University (USA) · Hangzhou, China, willing to relocate

MOTIVATION

Over the past year I led the engineering of an AI data platform, from its first commit to production: multimodal data pipelines, model serving, and an LLM agent that operates the platform through 150+ tools. The hardest problems were rarely about building features. They were about knowing whether a system actually worked.

Again and again, every conventional signal said “done” while the system was not usable. I want to study this gap systematically.

Green CI, dead feature
A data line passed CI with all 25 services running; a real click in a browser did nothing, because the front end had never been merged.

596 runs, zero reads
An LLM code reviewer ran 596 times on merge requests; its retrieval failed every time, so it never saw the code it was reviewing.

Status flags that lie
All 31 model services were flagged as failing; live probes showed 12 were healthy. One failure had frozen the flag permanently.

A fallback that hides failure
An AI support feature “passed” in a real environment, yet its 15 s timeout was below the gateway’s 19 s latency; the fallback disguised total failure as normal.

RESEARCH QUESTIONS

RQ1 How can we measure what an LLM agent actually sees and does?

Agent harnesses assemble context from tool schemas, history, memory and task state. In one deployment I instrumented 982 calls: tool definitions consumed 40.5% of prompt tokens and history only 8.8%, and volatile sections placed early held cache reuse to 51%, against 97% in a reference harness. I am interested in attribution and replay methods that make agent behaviour inspectable, and in evaluation suites that are themselves tested, for example by mutation, so they fail when the agent regresses.

RQ2 Where should humans sit in data and agent pipelines?

Routing low-confidence model outputs to reviewers, and gating write actions behind approvals, trade human effort for reliability. How should such thresholds and interfaces be designed, and how do we measure the quality gained per minute of expert time?

RQ3 What signals reflect real usability rather than activity?

Run counts, green pipelines and status columns measure that something happened, not that it worked. I want to develop and validate operational signals, such as live behavioural probes and end-to-end acceptance checks, that track real outcomes in deployed AI systems.

APPROACH AND WHAT I BRING

- Build instrumented, open research prototypes rather than toy demos, then evaluate them in controlled studies and in real deployments.
- Treat evaluation as software: versioned, tested and reproducible.
- Combine quantitative traces with qualitative study of how practitioners decide that something is “done”.
- Shipping software since 2017, from securities trading systems to an AI data platform; Java, Python, Go, TypeScript.
- Hands-on with LLM agents (MCP tools, workflows, context management), data pipelines, model serving, Kubernetes and GitOps.
- A habit of measuring before changing, and of writing down results that count against me.

Measurements above come from my own engineering work; most were taken in internal, non-production environments and are reported as such. I do not yet have peer-reviewed publications.

FIT

These questions sit across software engineering for AI, human-centred and responsible AI, and reliable data infrastructure. I am especially interested in industry-linked or applied doctoral programmes in Europe, the UK, Ireland, Australia and New Zealand, where research prototypes meet real users.

zchaw127@gmail.com hire.jiawis.com Agent work (demo) github.com/chaw1