Plan-Do-Check-Act Lifecycle for Agents

How a classical idea on quality control from manufacturing and synthetic data can help build reliable agents

The Deming Legacy in the Era of Agents

In the 1950s, W. Edwards Deming popularized the PDCA — Plan-Do-Check-Act — cycle as a practical model for continuous improvement. Ford's use of PDCA is rooted in its major quality improvement efforts in the late 1970s and early 1980s, when the company worked to reach world-class product quality. Ford's later Eight Disciplines Methodology, or 8D, expanded on the PDCA approach to help teams identify, correct, and eliminate recurring problems.

The core philosophy is timeless: don't just build a product — build a system that learns from its failures and improves over time.

That idea is newly relevant as enterprises move from deterministic software to autonomous AI agents. Agents are powerful and flexible, but they are also non-deterministic. While frontier models continue to improve, teams still need rigorous evaluation infrastructure to know whether agents are ready for production — not just demos.

A PDCA Lifecycle View of Agents

Deploying an agent is not a one-time launch. It is a continuous cycle of learning and refinement. The PDCA framework offers a simple way to think about that lifecycle:

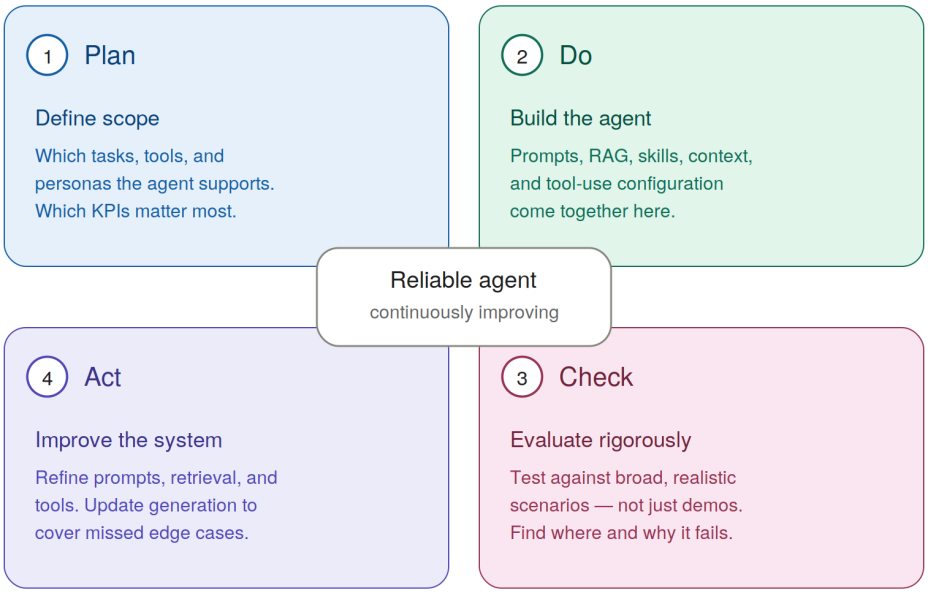


Figure 1: The PDCA Lifecycle for AI Agents

- **PLAN:** Define the agent's scope. What tasks should it handle? What tools, APIs, and personas does it need to support? What KPIs matter — accuracy, latency, cost, safety, or task completion?
- **DO:** Build and configure the agent. This includes prompt engineering, RAG, skills, context setup, and tool-use configuration.
- **CHECK:** Evaluate the agent rigorously. Instead of relying on a few demo examples, teams should test agents against broad, realistic scenarios to identify where they succeed, where they fail, and why.
- **ACT:** Use those findings to improve the system. Refine prompts, improve retrieval, adjust tools, and update data generation scripts to cover missed edge cases.

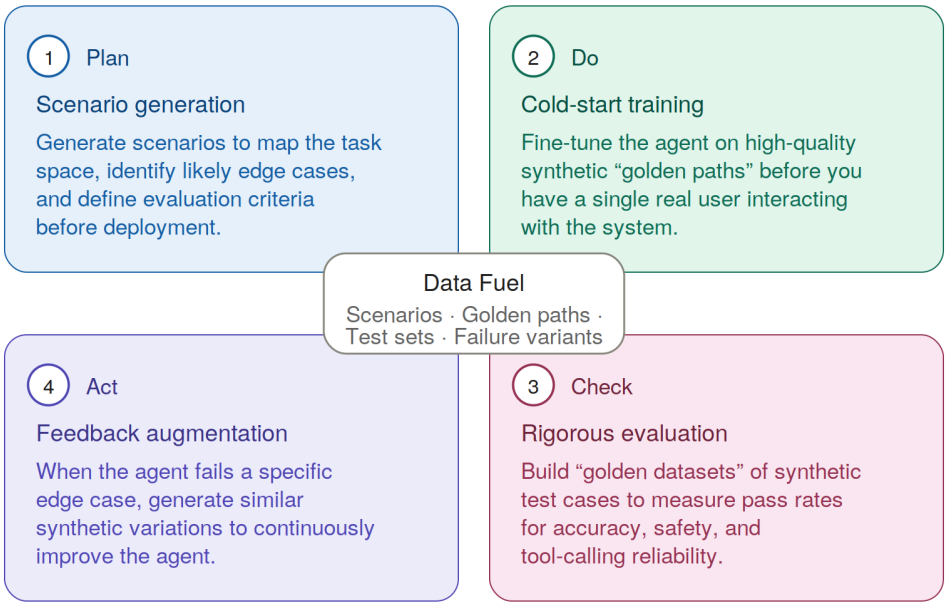
Synthetic Data—The "Data Fuel" for enabling PDCA for Agents

Real-world data is often messy, scarce, or contains sensitive PII (Personally Identifiable Information)— and it almost never covers the edge cases that matter most. Synthetic data closes that gap by giving teams a scalable, privacy-safe way to generate

representative, difficult, and high-risk scenarios before those scenarios appear in the wild.

Rather than replacing real data, synthetic data strengthens the development cycle by expanding coverage, increasing test diversity, and making evaluation more deliberate.

Ford plans to use a synthetic data approach to condition Product Development datasets (Bill of Materials, Product Definition, Engineering Targets) used in data-driven AI solutions, including Agentic AI systems. In many use cases, the historical data represents facts, but is not suitable as the complete ground truth for future solutions. These future solutions need to represent the "good" part of the historical data, discard the "bad" part of the historical data and incorporate data that represents policies that apply to future solutions (which, by definition, are not reflected in historical data). The need for adding completely new data comes from the need to reflect future policies, targets and regulatory requirements in the data. A prime example is in the domain of fuel economy, where future standards require a much higher level of performance than the past standards.



Synthetic data closes the gap real data can't reach.

Figure 2: How Synthetic Data Strengthens the PDCA Cycle for Agent Development

PDCA Stage	The Role of Synthetic Data
PLAN	**Scenario Generation:** Generate scenarios to map the task space more clearly, identify likely edge cases and define evaluation criteria before deployment.
DO	**Cold-Start Training:** Fine-tune your agent on high-quality synthetic "golden paths" before you have a single real user.
CHECK	**Rigorous Evaluation:** Build "Golden Datasets" of synthetic test cases to measure pass rates for accuracy, safety, and tool-calling.
ACT	**Feedback Augmentation:** If the agent fails a specific edge case, generate similar additional synthetic variations to continuously improve the agent.

Why Rigorous Evaluation is Non-Negotiable

The **CHECK** stage is where agent development becomes an engineering discipline rather than an exercise in intuition.

In production, “it looks fine” is not enough. Reliable agents require structured evaluation methods that measure behavior consistently and expose failure modes early.

That includes:

1. **Structured Review:** Apply consistent rubrics to assess whether the agent's outputs, reasoning, and decisions meet the required quality bar across diverse scenarios.
2. **Deterministic Tests**: Verify that the agent called the correct tool, followed the expected workflow, or returned the right output under known, reproducible conditions.
3. **Adversarial Testing**: Stress the agent with difficult or out-of-distribution inputs to uncover failures in safety, robustness, or instruction-following before they manifest in production.

Synthetic data makes this practical. It allows teams to test beyond the happy path, simulate risky scenarios, and evaluate performance at a scale that would be difficult with real-world examples alone.

Conclusion: Quality is a loop, not a launch

PDCA treats quality as an ongoing process, not a one-time milestone. That mindset fits AI agents, which are probabilistic, context-sensitive systems that require continuous evaluation and improvement.

Reliable agents are not created by prompt design alone. They emerge from a disciplined cycle of defining scope, building thoughtfully, evaluating rigorously, and improving continuously. Synthetic data accelerates that cycle by making it easier to test broader scenarios, uncover hidden weaknesses, and refine systems before failures reach end users. For teams deploying agents in production, the goal is not to ship something that works once — it is to build a system that keeps getting better.

Ford sees significant potential in using Agentic AI to raise process automation to the next level and improve cost, quality, and timing across next-generation Product Creation systems. But these systems must be reliable, predictable, and durable enough for highly regulated, safety-driven environments.

To support that goal, Ford plans to use synthetic data as the primary enabler for Product Development processes, in lieu of or in conjunction with traditional requirements-driven and test-driven development of solutions. The emergence of the Agentic AI wave created a new world of opportunities for Ford's Product Creation processes. Synthetic Data capabilities allows Ford to industrialize Agentic AI.

About the Authors

Product of Rockfish Data's engagement with Ford for the use of Synthetic Data in Ford's product creation processes.

Ford brings decades of operational experience applying continuous improvement across complex engineering environments.

Rockfish helps product and engineering teams generate the high-coverage data behind evals that ship AI models and agents faster — with fewer production failures.

To learn more, request a pilot, or discuss your agent evaluation use case: rockfish.ai | contact@rockfish.ai