

# Reflection-in-Action and Reflection-on-Action: Why Agile Iterations Matter More Than “Lessons Learnt”

by **Akolade Oladipupo** [April 2026]



Data collection exercise during a monitoring visit on the iLEAD Programme. [Source: LEAP Africa, 2024]

**Some** of the most brilliant ideas in development do not arrive when they are most needed.

They surface at the end.

In lessons learnt workshops. In evaluation reports. In reflective memos written long after programmes have closed. You hear them and think: that could have changed everything. But by then, the budget is spent, the team has moved on, and the communities have adapted to whatever outcomes the programme produced.

So why do we wait?

Why are we still treating learning as a post-programme activity in a world where the problems we’re trying to solve will not wait?

Across the development sector, monitoring, evaluation, and learning (MEL) has become standard practice. Yet learning itself is still too often treated as an end-stage activity rather than a continuous process embedded in implementation. This matters, especially now.

## The Stakes: Why Getting Implementation Right Has Never Mattered More

With fewer than five years remaining to the 2030 target for the Sustainable Development Goals (SDGs), progress is deeply uneven. The Sustainable Development Report 2025 finds that only 17 percent of SDG targets are currently on track globally, with sub-Saharan Africa lagging behind most other regions despite Africa being home to 12 of the 20 fastest-growing economies in the world.<sup>1,2</sup> The 2025 Africa Sustainable Development Report is explicit: the current pace of progress is insufficient to meet the 2030 deadline.<sup>2</sup> Goals related to zero hunger, decent work, quality education, and gender equality remain significantly off-track across much of the continent.

Resources are constrained. Contexts are complex and rapidly shifting. The margin for inefficiency is shrinking.

If development programmes are to deliver meaningful results, they must not only be well designed. They must be able to learn, adapt, and improve in real time. This is where reflection-in-action becomes not just useful, but essential.

## Beyond “Lessons Learnt”: Understanding the Two Modes of Reflection

Reflection has long been recognised as central to professional practice. John Dewey argued that without deliberate reflection, professionals risk becoming captive to routine and impulse, operating on autopilot in situations that call for genuine thought and adaptation.<sup>3</sup> Building on this foundation, Donald Schön, a professor at MIT, gave the development and education worlds one of the most practically useful frameworks for understanding how professionals actually learn.

Schön distinguished between two fundamentally different modes of professional reflection. The first,

reflection-on-action, involves stepping back after an experience to examine what happened, understand why, and draw lessons for the future. It is the mode that governs most of what many call MEL in development programming: endline evaluations, after-action reviews, annual learning reviews, and lessons learnt documentation. It is deliberate, structured, and retrospective. It has undeniable value.

The second mode, reflection-in-action, is something else entirely. Schön described it as thinking within the action-present, the window of time during which a practitioner can still influence the outcome of a situation.<sup>4</sup> It is not thinking after the fact. It is thinking within the fact. A jazz musician who adjusts their improvisation in response to what the other players are doing. A surgeon who reconceives their approach mid-operation when unexpected tissue is encountered. A field officer who notices that beneficiary attendance is dropping halfway through a programme cycle and pauses to ask why, and to act on the answer, before the full cohort has already been lost.

As Nguyen neatly puts it: in reflection-in-action, thinking and action are linked; in reflection-on-action, they are separated.<sup>5</sup> Both matter. But in a sector where implementation windows are narrow and course corrections become exponentially more costly the later they are made, the first mode carries particular urgency.

A third mode, reflection-for-action, rounds out the picture. Proposed by Olteanu as an extension of Schön's original framework, it describes the deliberate, anticipatory thinking that happens before an action: mapping assumptions, stress-testing designs, running pre-mortem analyses.<sup>6</sup> Together, the three modes form a complete learning architecture for any serious programme.

<sup>1</sup> Sachs, J. D., Lafortune, C., Fuller, C., & Iablonski, G. (2025). Financing sustainable development to 2030 and mid-century. Sustainable Development Report 2025. SDSN; Dublin University Press.

<sup>2</sup> UNDP. (2025). [2025 Africa sustainable development report](#). United Nations Development Programme.

<sup>3</sup> Dewey, J. (1933). *How we think*. University of Wisconsin Press.

<sup>4</sup> Schön, D. A. (1994). *The reflective practitioner: How professionals think in action*. Taylor & Francis.

<sup>5</sup> Nguyen, Q. D. (2015). *What is reflection? A conceptual analysis of major definitions and a proposal of a five-component definition and model* [Dissertation, University of Montreal].

<sup>6</sup> Olteanu, C. (2017). [Reflection-for-action and the choice or design of examples in the teaching of mathematics](#). *Mathematics Education Research Journal*, 29, 349-367.

## Box 1: Three Modes of Professional Reflection

- ▶ **Reflection-for-action (BEFORE):** Planning with intent. Surfacing assumptions. Pre-mortem analysis. Theory of change review.
- ▶ **Reflection-in-action (DURING):** Real-time thinking. Micro-adjustments during implementation. Iterative learning loops. Adaptive management.
- ▶ **Reflection-on-action (AFTER):** Post-implementation review. Lessons Learnt. Evaluation findings. Institutional memory.

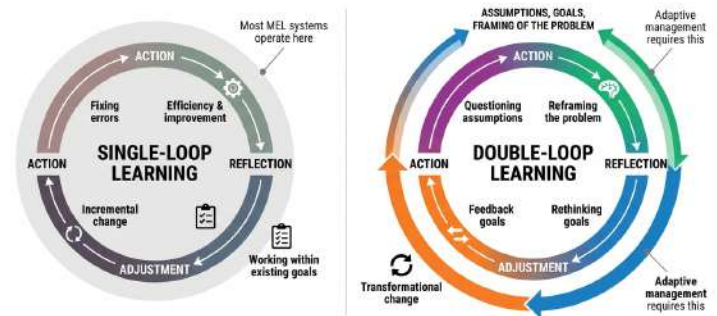
## Is Evaluation Overshadowing Monitoring? The MEL Imbalance

There is a quiet but consequential distortion in how many development organisations approach MEL. Evaluation, typically resource-intensive, often externally commissioned, and timed to programme milestones, midline or closeout, has come to occupy the centre of most learning conversations. Monitoring, which should provide the continuous, real-time intelligence that drives adaptive management, has too often been reduced to a compliance function: data collected, reports filed, boxes ticked.

The consequences of this imbalance are predictable. Static MEL systems fail to keep pace with dynamic realities. Monitoring data is collected but not used to adjust implementation. Learning events are scheduled but deprioritised under competing pressures. And by the time a formal evaluation produces its findings and recommendations—well reasoned and carefully documented—the opportunity to course-correct has often passed.

Argyris and Schön made a distinction here that is worth holding onto.<sup>7</sup> They separated single-loop learning, making small adjustments within existing assumptions, from double-loop learning, which

involves questioning and revising the underlying assumptions themselves.<sup>8</sup> Single-loop asks: how do we do this better? Double-loop asks: should we be doing this at all, and why did we assume it would work? Most MEL systems are structurally designed for single-loop learning at best. True adaptive management requires double-loop capacity. And that requires a culture and practice of honest, continuous reflection during implementation, not just after it.



Single-loop vs double-loop learning [Gen. with Gemini, April 8 2026]

## Frameworks for Real-Time Learning: What Practitioners Can Borrow

### ▶ USAID's Collaborating, Learning, and Adapting (CLA) Framework

Perhaps the most systematic institutional response to this challenge in the development sector is USAID's CLA framework. CLA is not a methodology so much as an orientation: it calls for intentional collaboration, continuous learning, and adaptive management to be embedded in programme design from the outset, not bolted on at the end.

The CLA framework operates across two dimensions. The first is CLA in the programme cycle, covering how learning is integrated into planning, implementation, and review processes. The second is enabling conditions, encompassing the organisational culture, daily processes, and resource allocation that allow learning to actually happen.<sup>9</sup> Both dimensions are necessary. Organisations that invest in learning tools

<sup>7</sup> Argyris, C. (1982). Reasoning, learning, and action: Individual and organisational. Jossey-Bass.

<sup>8</sup> Bassot, B. (2016). The reflective practice guide: An interdisciplinary approach to critical reflection. Routledge.

<sup>9</sup> USAID. (n.d.). Collaborating, learning, and adapting framework and key concepts (Implementing Partner Version 1.0). USAID.

without building a learning culture will find that the tools gather dust.



USAID CLA Framework<sup>9</sup>

USAID’s analysis of CLA case studies found five consistent patterns: that collaboration leverages resources for collective benefit; that local engagement produces stronger ownership and better outcomes; that intentional knowledge management generates transferable good practices; that feedback loops increase the likelihood that evidence actually informs decisions; and that CLA, once practised, tends to spread.<sup>10</sup> These are not abstract propositions. They are patterns drawn from real programmes across sub-Saharan Africa, South Asia, and Latin America.

► *Problem-Driven Iterative Adaptation (PDIA)*

A related but distinct framework, developed by Andrews, Pritchett, and Woolcock at Harvard’s Kennedy School, takes iterative adaptation even more seriously as a design principle.<sup>11</sup> Problem-Driven Iterative Adaptation (PDIA) argues that many

development reforms fail because they transplant prepackaged “best practice” solutions from other contexts rather than building locally appropriate responses through experimentation and iteration.

The four principles of PDIA are worth stating directly. First, reforms should focus on solving locally nominated problems, not imported solutions. Second, decision-making should be structured to encourage experimentation and positive deviance. Third, learning should be embedded in tight feedback loops that enable rapid adjustment, explicitly contrasted with the long lag times of ex post evaluation. Fourth, broad coalitions of local agents should be engaged throughout, not just external experts at design stage.<sup>12</sup>

An evaluation of PDIA’s application to public financial management reform in six African countries found that it delivered measurable short-term results, particularly where a strong authorising agent and a dedicated implementation team were in place.<sup>12</sup> These outcomes did not come from a lessons learnt document. They came from a process designed to learn continuously, within the action-present, not after it.

► *Agile Iterations: Learning from Software Development*

The software development world has grappled with the same underlying problem for decades: how do you build complex things in uncertain environments without rigidly committing to a design that the real world will quickly invalidate? The Agile Manifesto, published in 2001, represented a decisive rejection of waterfall development, with its long planning cycles and rigid implementation, in favour of iterative, incremental delivery with continuous feedback.<sup>13</sup>

Several Agile principles translate directly to development programme management. Short, structured reflection sessions embedded within

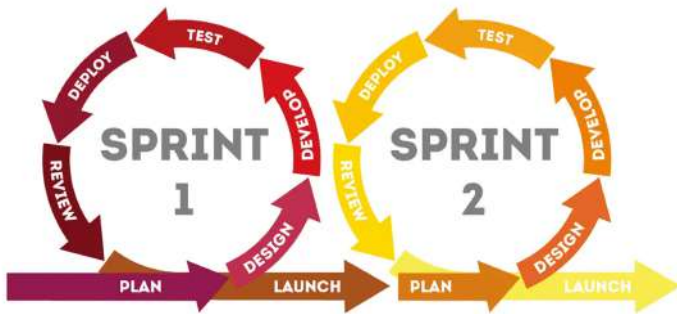
<sup>10</sup> USAID. (2016). Collaborating, learning and adapting: An analysis of what CLA looks like in development programming. USAID Bureau for Policy, Planning and Learning.

<sup>11</sup> Andrews, M., Pritchett, L., & Woolcock, M. (2013). [Escaping capability traps through Problem-Driven Iterative Adaptation \(PDIA\)](#). World Development, 51, 234-244.

<sup>12</sup> Lawson, A., & Harris, J. (2023). [Is the problem driven iterative adaptation approach \(PDIA\) a panacea for public financial management reform? Evidence from six African countries](#). World Development Perspectives, 31, 100526.

<sup>13</sup> Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., et al. (2001). Manifesto for agile software development. Agile Alliance.

ongoing implementation—akin to sprint reviews—serve as operational pause-and-reflect moments. These are not end-of-programme reflections, but rapid learning loops that occur while activities are still unfolding and can still be shaped.



Agile iterations through sprint sessions<sup>14</sup>

Retrospectives ask not just what was delivered but how the team is working and what should change. The discipline of maintaining a backlog of learning questions, and revisiting it regularly as new information emerges, maps cleanly onto a living learning agenda for a development programme. Organisations that adopt even a modified sprint-based rhythm for their programme teams create far more opportunity for in-process adaptation than those waiting for the quarterly report.

► *Plan-Do-Check-Act and the Question of When 'Check' Happens*

The PDCA cycle, developed by Walter Shewhart and popularised in quality management by W. Edwards Deming, offers development practitioners a deceptively simple framework.<sup>15</sup> The four steps are familiar. The critical question for our purposes is where, in the sequence, the Check phase falls?

In many programme management cultures, checking is treated as a terminal activity, something done at the end of a phase or cycle. But in Deming's original intent, PDCA is continuous: Plan an intervention in limited scope, Do it, Check the results while you still can influence the next step, then Act to

adjust the plan before the following cycle. Applied to development, this means building regular, structured check moments into implementation calendars as genuine learning inflection points, not bureaucratic reporting exercises.



PDCA cycle [Gen. with Gemini, April 8 2026]

**Case Studies in Development Programmes**

**1. The (Re)solve Project: Adaptive Learning Meetings in Burkina Faso**

A compelling recent example of real-time learning in practice comes from the (re)solve project, a five-year multicountry initiative testing new approaches to family planning in West Africa. In Burkina Faso, the project team designed what they called an Adaptive Learning Meeting (ALM) cycle: a frequent, rapid, cross-team meeting process built explicitly as a responsive feedback mechanism during implementation.

The ALM cycle followed a five-step process for group-based learning and problem-solving, using monitoring and observational data to allow a geographically dispersed team to learn together and adapt to real-world implementation challenges as they emerged. The result was a series of context-specific adaptations that were responsive to the needs of girls, community-based facilitators, and healthcare workers, changes that were made while the programme was still running, not after it had ended.<sup>16</sup> The ALM cycle also had an important

<sup>14</sup> RadView. (2021, April 1). [Agile and performance testing: A winning combination](#).

<sup>15</sup> Deming, W. E. (1986). Out of the crisis. MIT Center for Advanced Educational Services.

<sup>16</sup> Trasi, R., Angelone, C., & Hounkanrin, G. (2023). [Designing and implementing the adaptive learning meeting cycle: The \(re\)solve project experience in Burkina Faso](#). Global Health: Science and Practice, 11(Suppl. 2), e2200217.

side-effect: it built a team culture that valued dialogue, honest problem-sharing, and continuous improvement. Culture and tools reinforced each other.

## **2. *The iLEAD Hausa Manual: When Teams Act on What They Learn***

The iLEAD programme, implemented in 2024 by LEAP Africa across public secondary schools in Sokoto State, Nigeria, offers a compelling illustration not just of reflection-in-action, but of how learning can travel across the full programme cycle. The programme aimed to equip Senior Secondary School students with leadership, entrepreneurship, and vocational skills, delivered through trained teachers using the iLEAD manual.

From the outset, insights from the needs assessment and baseline survey pointed to a critical constraint: low English proficiency among students posed a notable barrier to comprehension and engagement with the curriculum. This was not an unexpected discovery, but it was a decisive one. In response, the programme simplified the curriculum and translated the manual into Hausa, the dominant language in the region, making it more accessible to the students.

What makes this case particularly instructive is what followed. The adaptation was not treated as a one-off design decision, but as a hypothesis to be tested in practice. During implementation, feedback from teachers, classroom interactions, and midline data provided real-time validation of the change. By midline, 89 percent of teachers reported that the Hausa version of the manual was highly effective in facilitating curriculum delivery.<sup>17</sup> The adaptation proved instrumental in improving student understanding, engagement, and overall learning experience.

The iLEAD evaluation report captures this under the understated but significant insight that “learning and adaptation are key to effective implementation”.<sup>17</sup> What this case demonstrates is not just the value of

identifying the right problem early, but the importance of continuing to learn during implementation. Reflection-for-action informed the initial adaptation. Reflection-in-action ensured that the adaptation worked. Together, they shaped outcomes in real time.

## **3. *USAID Community Connector and the Savings Model Redesign***

The USAID Community Connector (CC) project in Uganda, a flagship integrated nutrition and agriculture initiative implemented by FHI 360, built CLA explicitly into its design through a modular structure alternating learning phases with implementation phases. The approach produced multiple documented instances of real-time adaptation throughout the project’s five-year lifespan.

One of the most instructive concerns was the Village Savings and Loan Association (VSLA) model. Field staff observed, through monthly regional meetings, that community group members were using their year-end savings disbursements to fund December holiday celebrations rather than investing in productive assets or agricultural inputs. This was the stated goal of the VSLA model. It was not being met.

Rather than logging this as a finding for the next programme, CC acted within the current cycle. It conducted a rapid community mapping exercise, held focused group discussions with VSLA members, consulted local government, and redesigned the savings model into what it called Saving With a Purpose (SWAP): disbursements timed to coincide with planting seasons and school calendar cycles, with members required to set and commit to savings goals. Over 75 percent of CC community groups subsequently adopted the SWAP approach, with many groups making their own further contextual adaptations.<sup>18</sup> The adaptation was completed and rolled out while the project was still running. The lesson did not wait for closeout.

<sup>17</sup> LEAP Africa (2025). [Endline Evaluation Report of the 2024 iLEAD Programme](#)

<sup>18</sup> FHI 360/USAID. (2015). Collaborating, learning and adapting: Experiences from first four years of the USAID Community Connector Project (January 2012-June 2015). USAID Community Connector Technical Notes Series No. 9. FHI 360.

## Learning Tools across Programme Phases

One of the most actionable shifts development organisations can make is not simply to differentiate, but to intentionally align learning tools with the moments in the programme cycle where they are most useful. While many tools can be applied across multiple phases, their value depends heavily on *when* and *how* they are used.

Both in-process and post-implementation learning matter. But they must be deliberately chosen, appropriately timed, and adequately resourced. The table below offers a practical way of thinking about this alignment, highlighting where different tools tend to deliver the greatest value, without suggesting rigid boundaries.

**Table 1: Learning Tools Across the Programme Cycle\***

BEFORE Implementation	DURING Implementation	AFTER Implementation
Theory of Change review	Pause & Reflect sessions	Lessons Learnt workshop
Assumption & risk mapping	Intra-Action Reviews (IAR)	End-of-project evaluation
Pre-mortem analysis	Debrief sessions	After-Action Reviews (AAR)
Baseline & context assessment	Adaptive / Change log	Evidence briefs
Peer Assist (incoming knowledge)	Real-time dashboards	Learning briefs
Stakeholder mapping workshop	Open Space facilitation	World Cafe dialogues
Collaborative design workshop	Learning briefs (interim)	Peer Assist (outgoing)

\*This categorisation is not absolute. Several tools, such as After-Action Reviews, Peer Assists, and Learning Briefs, can be adapted across phases. The critical question is not where a tool “belongs,” but whether it is being used at a moment when it can still influence decisions.

A few tools in this table deserve particular attention in the African development context. Intra-Action Reviews (IARs), developed by the World Health Organization during the COVID-19 response, enable structured collective reflection while an emergency or programme phase is still ongoing, capturing learning that would otherwise evaporate under the urgency of delivery.<sup>19</sup> Pause and Reflect sessions, central to the CLA framework, create deliberate moments of collective sensemaking within implementation teams. Open Space Technology and World Cafe dialogues are participatory methods well-suited to surfacing the distributed knowledge that exists within and around a programme, including the insights of the communities being served, often the last voices to be consulted in traditional MEL processes.

## Why Organisations Resist In-Process Learning: Naming the Barriers

If iterative learning during implementation is so clearly valuable, why is it still the exception rather than the norm? The barriers are real, and naming them honestly is the first step toward addressing them.

**Structural barrier:** The structural barrier is the most pervasive. Most development contracts are designed around deliverables and milestones, not learning cycles. When a donor’s disbursement is tied to the production of a report, the incentive is to produce the report, not to pause and question whether the activities generating it are actually working. PDIA’s application to African public financial management reform found that flexibility and local ownership were essential to enabling adaptive action, and that both

<sup>19</sup> World Health Organization. (2020). Intra-action review: A guide for conducting reviews during the COVID-19 pandemic. WHO.

were constrained where contracting arrangements were too rigid.<sup>12</sup>

**Cultural barrier:** The cultural barrier is equally significant. Acknowledging mid-implementation that something is not working is, in many organisational cultures, experienced as failure. Programme teams operating under donor scrutiny are understandably reluctant to surface difficulties. But as both the Community Connector and (re)solve experiences demonstrate, learning from failure openly, quickly, and collectively is precisely where the greatest opportunities for improvement lie. Psychological safety for honest learning conversations is a leadership challenge as much as a technical one.

**Capacity barrier:** The capacity barrier should not be overlooked. Reflection-in-action, as Schön emphasised, is a skill that develops with practice.<sup>4</sup> It requires facilitation, it requires protected time, and it requires that programme staff have both the cognitive space and the practical tools to make sense of emerging evidence. MEL systems that are designed only to collect and report data do not automatically generate the conditions for adaptive learning.

### ⚠ **Box 2: Five Conditions That Enable Reflection-in-Action at Programme Level**

1. Dedicated time and budget for learning activities within the programme cycle
2. Real-time data systems that make monitoring information accessible to decision-makers at all levels
3. Organisational culture of openness, psychological safety, and tolerance for honest reporting of difficulties
4. Flexible contracting and management arrangements that allow adaptive action without heavy administrative burden
5. Leadership that models reflective practice and visibly rewards evidence-informed course corrections

## **This Is Not a Case Against Post-Implementation Reflection**

It would be a misreading of this article to conclude that lessons-learned exercises, end-of-project evaluations, and retrospective reflection have no value. They have enormous value, and the development sector would be poorer without them.

The most rigorous evidence for what works often comes from patient, longitudinal evaluation. Impact evaluations, systematic reviews, and meta-analyses produce credible, generalisable knowledge that no single programme can generate in real time. They surface patterns invisible to any individual implementer. They build the technical evidence base upon which future programme designs are grounded. Post-implementation reflection also brings to the surface the systemic inequities, structural barriers, and power dynamics that in-process learning, focused on immediate problem-solving, can miss. Stepping outside the frame is sometimes the only way to see the full picture.

Argyris and Schön's concept of double-loop learning is at its most powerful in retrospect, when there is sufficient distance to question not just tactics but the underlying assumptions that shaped the entire programme design. Gibbs' reflective cycle, with its deliberate sequence of description, feelings, evaluation, analysis, conclusions, and action planning, provides value precisely because it requires depth, not just speed.<sup>20</sup>

The argument here is not that retrospective reflection should be abandoned. It is that retrospective reflection should not be the primary, or often the only, mode of reflection available to a programme. The current imbalance, where learning is resourced and scheduled almost exclusively at programme close, is simply not adequate for a world that demands faster adaptation. We need both modes. The question is whether we have designed our programmes, our contracts, and our organisational cultures to make both genuinely possible.

<sup>20</sup> University of Hull. (n.d.). [Reflective writing: Reflective frameworks](#).

## Putting On the Einstein Cap Earlier

Development work is too important, and too resource-constrained, to rely primarily on hindsight. Brilliant insights that arrive at programme close are valuable, but they are also, in a fundamental sense, missed opportunities. The challenge for the sector is not just to generate knowledge, but to use it when it still matters.

We can build different architectures. The tools exist. The frameworks exist. The evidence exists. What remains is the will to deploy them *during* implementation, not merely to document them afterward.

The 2030 SDG target is not a bureaucratic milestone. It represents a collective commitment to human dignity. Meeting even a fraction of what remains undone will require programmes that do not just run well, but learn fast. In a complex world, the ability to adjust course mid-flight may matter more than the ability to plan perfectly at the start. Brilliant hindsight is valuable. Real-time intelligence is a superpower.

Treat reflection not as a phase, but as a practice. Put on the Einstein cap during implementation. That is where it matters most.

