Outcome-Driven Frame-works Deep Dive

A Typology with C2O as Canonical Model

Scan for full guide with examples, checklists, and updates:
contribution2outcome.com/kb/foundations/outcome-driven-frame-works

contribution2outcome.com

December 17, 2025
Version 1.0
Owner: Stark Burns

Table of Contents

Formalizing Outcome-Driven Frameworks

A Typology with C2O as Canonical Model

1. Executive summary

Organizations are surrounded by frameworks: OKRs for goals, RACI and MOCHA for responsibilities, DACI and RAPID for decisions, Outcome Mapping for evaluation, and various lifecycle models for project governance.

Each of these tools solves part of the collaboration puzzle. And yet, teams still face familiar problems:

- Slow, re-litigated decisions
- Experts becoming bottlenecks in AI and transformation projects
- Internal service teams treated as gatekeepers rather than partners
- Launches that succeed on paper but fail to be adopted

One reason: most of these frameworks are **task-first** or **decision-first**, not **outcome-first**.

This paper introduces and formalizes the category of **Outcome-Driven Frameworks** — models that organize collaboration around the lifecycle of a specific outcome. We propose a simple typology of frameworks and position **Contribution to Outcome (C2O)** as a canonical outcome-driven framework for the AI-augmented enterprise.

By the end of this paper you will:

1. Understand a practical typology of collaboration and governance frameworks.
2. Recognize what makes outcome-driven frameworks distinct.
3. See how C2O implements the outcome-driven pattern in a concrete, usable way.
4. Know how to stack C2O alongside the tools you already rely on (OKRs, RACI, DACI, Outcome Mapping, and others).

2. A typology of collaboration and governance frameworks

2.1 Why a typology

When the word "framework" covers everything from goal-setting to decision protocols, teams talk past each other and stack tools on top of one another without a clear logic.

Having a typology helps you:

- Choose the right tool for the problem in front of you.
- Avoid "framework soup" where overlapping tools compete.
- See clearly where something genuinely new — like C2O — adds value instead of simply rebranding existing ideas.

2.2 Two simple axes

For practical purposes, we use two axes:

1. **Primary focus**
    - Tasks and responsibilities

- Decisions
- Outcomes and goals
- Evaluation and learning

2. **Time horizon**
    - Event or point-in-time
    - Lifecycle or ongoing

2.3 Placing common frameworks

On these axes, familiar tools cluster as follows:

Task-driven frameworks

- Examples: RACI, RASCI, MOCHA, responsibility assignment matrices.
- Focus: who is responsible, accountable, consulted, or informed for each task or deliverable.
- Horizon: mostly event-level, for example this task or that deliverable.

Decision-driven frameworks

- Examples: DACI, RAPID, decision rights matrices.
- Focus: who recommends, who decides, who performs, who must agree.
- Horizon: individual decisions or clusters of decisions.

Outcome or goal-driven frameworks

- Examples: OKRs (objectives and key results), strategy maps.
- Focus: what success looks like and how to measure progress.
- Horizon: quarters, years, sometimes multi-year strategy cycles.

Evaluation-driven frameworks

- Examples: Outcome Mapping, Outcome Harvesting, contribution analysis.
- Focus: how a programme contributes to observed changes in the world.
- Horizon: programme lifecycles and beyond.

Lifecycle-driven frameworks

- Examples: project lifecycle models such as initiate, plan, execute, close.
- Focus: phases, gates, and assurance activities.
- Horizon: projects and programmes.

We can summarize this in a simple table:

Table 1: Category / Examples / Primary focus

Category	Examples	Primary focus	Time horizon	Main question
Task-driven	RACI, MOCHA, RAM	Tasks and responsibilities	Events or tasks	Who does what
Decision-driven	DACI, RAPID	Decisions	Events	Who recommends, decides, and executes
Outcome or goal-driven	OKRs, strategy maps	Outcomes and goals	Quarters or years	What does success look like
Evaluation-driven	Outcome Mapping, OH	Contribution to change	Programmes	How did we influence these changes
Lifecycle-driven	PM lifecycles, PM²	Phases and gates	Projects and programmes	Where are we in the process

Outcome-driven	C2O	Outcomes and contributions	Outcome life-cycle	How do we contribute to this outcome over time

Most mature organizations have reasonable coverage in the first five categories. **Outcome-driven frameworks** are where the gap has been.

3. Defining outcome-driven frameworks

3.1 Working definition

Three ideas are doing work in that definition:

- **Outcome** — a clear description of the change you are trying to create.
- **Lifecycle** — the journey from first signal or idea through to sustained adoption.
- **Contribution** — how different people and systems show up across that journey.

3.2 Necessary characteristics

To qualify as outcome-driven in this sense, a framework should demonstrate at least six characteristics:

1. **Outcome as the organizing object**
   Work is planned, coordinated, and measured at the level of outcomes such as "increase activation rate from forty to fifty five percent," rather than simply at the level of projects or deliverables.
2. **Lifecycle structure**
   The framework defines a small number of phases every outcome passes through. For example, discover, design, build, operate, adopt.

These phases are more than labels; they carry distinct decision patterns, evidence requirements, and risk profiles.

3. **Contribution types, not only titles**
   Roles are defined by how they contribute, not only by job title or org chart position. Contribution types are reusable across outcomes and teams.
4. **Explicit treatment of adoption and change**
   Adoption, enablement, and behaviour change are not afterthoughts, they appear as named phases with clear responsibilities and evidence patterns.
5. **Shared ownership orientation**
   The framework is explicitly designed to foster shared ownership, often called collective psychological ownership, not just clear blame assignment.
6. **Integrative posture**
   It is intended to sit alongside task, decision, and goal frameworks — integrating them into a coherent operating system rather than competing with them.

3.3 Why outcome-driven frameworks are emerging now

Three shifts are making outcome-driven frameworks more important:

1. **AI-augmented work and expert bottlenecks**
   AI tools increase the volume and speed of work but also increase the burden on senior experts who must supervise, validate, and govern that work. Without a clear collaboration pattern, experts become bottlenecks.
2. **Cross-functional outcomes as the norm**
   Most important outcomes span product, data, security, legal, finance, operations, and external partners. Silos make these outcomes fragile.
3. **Adoption as the real constraint**
   In many transformations, the technology ships on time. The value is lost in adoption. Outcome-driven frameworks bring adoption into the same model as delivery.

4. C2O as a canonical outcome-driven framework

4.1 C2O in one paragraph

C2O, Contribution to Outcome, is an outcome-driven collaboration framework that maps how roles **Drive, Contribute, Enable, Advise, and Inform** the lifecycle of an outcome across **Discover, Decide, Build, Run, and Adopt**. It is designed to foster collective ownership, elevate internal services as partners, and integrate AI systems as first-class contributors while coexisting with traditional governance tools such as RACI and DACI.

4.2 Mapping C2O to the characteristics

1. **Outcome as anchor**
   C2O starts by defining outcomes with acceptance criteria and signals. Every contribution is explicitly tied to those outcomes.
2. **Lifecycle structure**
   The vertical axis of the C2O matrix is a fixed lifecycle: discover, decide, build, run, adopt.
   Each phase has clear entry and exit criteria, decisions, and evidence patterns.
3. **Contribution types, expressed as verbs**
   The horizontal axis uses verbs: drive, contribute, enable, advise, inform.
   These verbs describe how actors, human or AI, participate at each phase.
4. **Adoption as first-class**
   Adopt is a fully articulated phase with its own drivers, contributors, enablers, and advisers.
   Adoption responsibilities are not pushed into a vague "business as usual" bucket.
5. **Shared ownership orientation**
   C2O is paired with an ownership overlay — practices from modern leadership and team-of-teams literature mapped to each verb.
   The explicit aim is collective psychological ownership across functions.
6. **Integrative posture**
   C2O is positioned to sit alongside OKRs, RACI and MOCHA, DACI and RAPID, and

evaluation frameworks.

It provides the **collaboration fabric** into which these tools plug.

4.3 Structural distinctions

C2O differs structurally from other frameworks in three important ways:

1. **Lifecycle by verbs matrix**
   Most responsibility matrices are tasks by roles. C2O is outcomes by lifecycle by verbs. This keeps the focus on results and shared contribution, not on tickets.
2. **Reframed enable role**
   In many matrices, support or helper roles are invisible and subordinate.
   In C2O, **Enable** is a first-class contribution type, often owned by internal service teams such as security, platform, finance, and data who become genuine partners.
3. **Formal treatment of AI contributors**
   C2O explicitly treats AI systems as potential contribute or enable roles with human supervision and clear decision rights.
   This allows responsibility for AI-mediated work to be distributed and governed intentionally.

5. Using the typology in practice

5.1 Choosing the right tool for the job

Use the typology as a decision aid:

- If your problem is that you do not know who owns a task, start with RACI or MOCHA.
- If your problem is that you cannot get a decision made, use DACI or RAPID.
- If your problem is that you do not know what you are aiming for, clarify OKRs.

- If your problem is that you do not know whether you made a difference, use Outcome Mapping or similar.
- If your problem is that you keep stalling on cross-functional outcomes, especially with AI in the mix, adopt an outcome-driven framework such as C2O.

5.2 Stacking frameworks coherently

A simple stack might look like:

- Strategy and goals through **OKRs**.
- Outcome-level collaboration across lifecycle and functions through **C2O**.
- Formal accountability and specific tasks through **RACI, MOCHA, or DRI**.
- High-stakes decisions within each phase through **DACI or RAPID**.
- Long-term learning through **Outcome Mapping** or other evaluation frameworks.

In this picture, C2O is not an extra layer of bureaucracy. It is the **connecting tissue** that makes the other frameworks work together instead of competing.

5.3 Implementation pattern

A practical adoption path:

1. **Map your current stack.**
   Where do OKRs live. Where are RACI charts used. How are major decisions currently made and recorded.
2. **Choose one high-value outcome.**
   Cross-functional, visible, and currently painful is ideal.
3. **Build a C2O map for that outcome.**
   Define the outcome, then assign drive, contribute, enable, advise, and inform roles across discover, decide, build, run, and adopt.
4. **Overlay existing tools.**
   Map the RACI accountable to sponsors.

Identify where DACI or RAPID would help in specific decisions.
Ensure OKRs are aligned with the outcome.

5. **Run the experiment for one lifecycle.**
   Use the C2O map in real meetings, gates, and reviews.
   Adjust roles and contributions as you learn.
6. **Measure impact.**
   Track decision latency, escalation volume, expert load, and adoption metrics before and after.
   Collect qualitative feedback on provider versus partner dynamics.
7. **Scale with patterns, not heroics.**
   Turn successful C2O maps into templates.
   Build lightweight guides and examples tailored to your context.

6. Research directions and next steps

Formalizing outcome-driven frameworks is a starting point, not an endpoint.

Promising areas for further work include:

- **Empirical evaluation** of outcome-driven frameworks versus task- or decision-driven frameworks on metrics such as speed, adoption, and resilience.
- **Integration with AI governance** standards, mapping risk and accountability to outcome-driven roles and phases.
- **Sector-specific variants** for example in healthcare, financial services, or ESG reporting that share the same pattern but specialize lifecycle phases and verb definitions.

For now, the Contribution to Outcome framework provides a practical, field-tested example of an outcome-driven model that you can deploy today.

Use this typology to clarify language inside your organization, position C2O alongside tools you already trust, and design experiments that move you from task-driven coordination to genuine, outcome-driven collaboration.

About this document

This deep-dive is a companion to the practitioner-level KB article at contribution2outcome.com/kb/foundations/outcome-driven-frameworks.

For more on C2O:

- What is C2O?
- Lifecycle Overview
- Decision Rights

Related Documents

- Contribution Mapping Canvas — C2O Template
- contribution2outcome.com/kb/foundations/outcome-driven-frameworks