

Why Well-Managed Transformations Can Still Fall Short of Their Intended Goals

Introducing the Capability-Driven Transformation Framework

Allen C. Stines, Ph.D.

White Paper

What this paper covers

Some transformations are governed carefully, staffed by capable people, and delivered through familiar program, technology, and change disciplines, yet they still fail to produce the organizational outcomes they were intended to create. This paper explains why that can happen and what changes when the organization is understood before the response is designed. It introduces the Capability-Driven Transformation Framework (CDTF), outlines its seven-model architecture, and works through three situations in which a more precise diagnosis leads to a materially different design and execution.

This paper introduces the framework. [Book One](#) explains the research behind it, the models and constructs it is built on, and the reasoning that connects them.

A transformation can be well managed and still be poorly designed

Consider a transformation eighteen months in. The steering committee meets every two weeks, senior sponsors attend regularly, training completion exceeds ninety percent, and every workstream reports green. The new operating model goes live and appears to be working. Within several weeks, however, workarounds begin to appear. A year later, the organization is operating much as it did before, except that new systems, terminology, and reporting structures now sit on top of the old patterns.

None of that points to neglect or incompetence. The sponsors were present, the program plan was credible, the training was delivered, and the reporting reflected what the program was designed to measure. These cases are difficult because the usual explanations don't fit. There may be no obvious resistance or communication breakdown, and no single decision looks wrong on its own.

Active sponsorship, completed training, regular governance, disciplined workstreams, and favorable dashboards provide evidence that the program is being managed. They don't establish whether the design was built on an accurate understanding of the organization. A program can manage activity effectively while failing to account for important organizational conditions.

Managing a transformation well and designing it well are related, but they aren't the same thing. Competence in the first doesn't guarantee quality in the second.

Program management organizes scope, sequence, resources, milestones, dependencies, and identified risks. Transformation design addresses a different question: what must become true about the organization for the new way of working to survive an ordinary operating day after the program winds down? Program management is visible, and leaders review progress against the plan regularly. The second depends on conditions that are harder to observe, including incentives, interpretation, informal authority, capability ownership, evidence quality, and whether the organization learns as it executes.

Why important conditions can remain outside the program's view

A fixed methodology prepares a program to answer the questions built into that methodology. This is a legitimate and useful function because established methods create discipline, consistency, and a shared operating language, but their limitation is structural. Conditions that fall outside the method's questions can remain unexamined until they disrupt execution or produce an outcome that the program didn't anticipate.

Transformation outcomes often appear unpredictable because the conditions shaping them were not visible to the program. Those conditions may already be present in the incentive system, in the different meanings functions attach to the same language, in the location of critical capability, or in governance rules that prevent an early design decision from being reconsidered. Such conditions rarely announce themselves as identified risks, and they may never appear on a conventional status report.

What established disciplines tend to see

Every discipline involved in a transformation tends to be organized around a particular unit of analysis. Each discipline can perform competently within its own domain while nobody has

accounted for how the parts interact as a whole. The resulting gap is structural rather than a reflection of professional failure.

- Program management centers on delivery through scope, sequence, dependencies, resources, milestones, and risks that have already been identified.
- Change management, as most widely practiced today, centers on individual and stakeholder movement through awareness, willingness, personal capability, engagement, and adoption.
- Learning and development centers on knowledge and skill, often through participation, completion, and assessment, while sustained proficiency under operating conditions is less directly observed.
- Organization design centers on structures, roles, decision rights, reporting relationships, processes, and formal accountability.
- Technology delivery centers on requirements, configuration, integration, testing, deployment, and the work required to place the system into service.
- Risk management centers on known or anticipated risks that the organization has been able to identify, categorize, and assign.

Each discipline can meet its own standards while the combined system produces an outcome that no one intended. Two workstreams can deliver excellent outputs based on incompatible assumptions about how their outputs fit together. A training program can be thoughtfully designed around capability requirements that were defined incorrectly. A formal structure can assign accountability while the incentive system continues to reward the previous way of working. Every component can pass its own inspection even as the organization moves toward an incoherent result.

From established research to an organizing architecture

CDTF is grounded in decades of established theory and research across organizational science, systems thinking, sensemaking, learning, behavior, organization design, measurement, risk, complexity, capability development, and organizational learning. Its original contribution is the architecture that connects those fields through seven models and nearly eighty differentiated constructs, then translates them into diagnostic questions, design consequences, and implications for execution.

Organizational theory has explained many of the forces that shape transformation for decades. What organizations often lack is an organized way to account for those forces together and connect the evidence to decisions about what the transformation should change. CDTF provides that connection. Book One, the first in the series, shows how the architecture was built and distinguishes constructs original to CDTF from those that build on established concepts.

Constructs provide diagnostic precision

A construct is the framework's unit of diagnostic precision. It names a specific organizational condition, defines its boundaries, identifies the evidence that would support or weaken the diagnosis, and clarifies why the distinction matters for design. This level of precision is

necessary because familiar symptom labels such as resistance, poor communication, low readiness, or weak adoption can describe several different underlying conditions. Treating those conditions as interchangeable leads programs toward generic responses that may be irrelevant or counterproductive.

The framework is configured rather than applied as a standard package. A condition that deserves significant attention in one organization may have little relevance in another, and evidence gathered during execution may change which constructs remain active. The architecture therefore provides disciplined variety without turning the framework into a fixed sequence or checklist.

Lenses and levers, not phases

The seven models operate as lenses for examining the organization and as levers for shaping the transformation. As a lens, each model brings a distinct class of conditions into focus. As a lever, it converts what the diagnosis reveals into choices about governance, evidence, capability, coordination, ownership, reinforcement, and learning. The models are not phases in a prescribed sequence. Their relative importance changes with the organization, the transformation, and the evidence that emerges over time.

Diagnosis, design, and execution remain connected

CDTF connects diagnosis, design, and execution. Diagnosis identifies the condition that requires attention. Design determines what must change in response. Execution makes that response operational and produces evidence about whether the diagnosis and design remain valid. The three often come apart. A diagnosis that doesn't influence design remains an observation, and a design that never changes execution remains a document. CDTF ties the three together, and evidence from execution can reopen the diagnosis or revise the design.

Seven models, one connected system

Each model addresses a distinct class of organizational conditions and requires its own diagnostic and design logic. CDTF does not assume every model carries equal weight. The framework helps leaders determine which conditions deserve the greatest attention in the organization and transformation in front of them.

Model	What it examines	Primary question
PRIME People engagement, Receptivity, Influence, Momentum, and Executive governance	Examines whether the transformation has the engagement it needs to create genuine investment, credible sponsorship, and honest participation.	Are the right people engaged in ways that allow them to support, influence, and enable the transformation?
FABRIC Foundational Alignment Bridging Roles for Interpretive Coherence	Examines whether different parts of the organization are translating the transformation into compatible decisions, behaviors, and operating assumptions.	Are people interpreting the transformation in a way that allows the organization to move coherently?
SENSE Systemic Evaluation of Network Signals and Exposure	Examines how risks, early warning signs, informal adaptations, and emerging conditions are detected before they become visible in formal indicators.	What risks, instabilities, and emerging conditions need to be surfaced and addressed?
FORGE Functional Organizational Readiness, Growth, and Enablement	Examines whether the organization is developing the durable capability required to operate differently under real conditions, once outside support ends.	Can the organization actually operate differently?
VECTOR Velocity, Execution, Coordination, Trajectory, Orchestration, and Rhythm	Examines whether work across functions and workstreams is producing coordinated enterprise movement rather than disconnected activity.	Can the enterprise move coherently through the transformation?
ANCHOR Alignment, Normalization, Continuity, and Human-Centered Organizational Reinforcement	Examines whether the new way of operating survives without the structures, attention, and reinforcement the program provided.	Will the transformation remain in place once the program is gone?
EVOLVE Emergent Vision, Organizational Learning, and Value Endurance	Examines whether experience is being converted into organizational intelligence that changes current and future decisions.	What is the organization learning, and how does that change the decisions it makes?

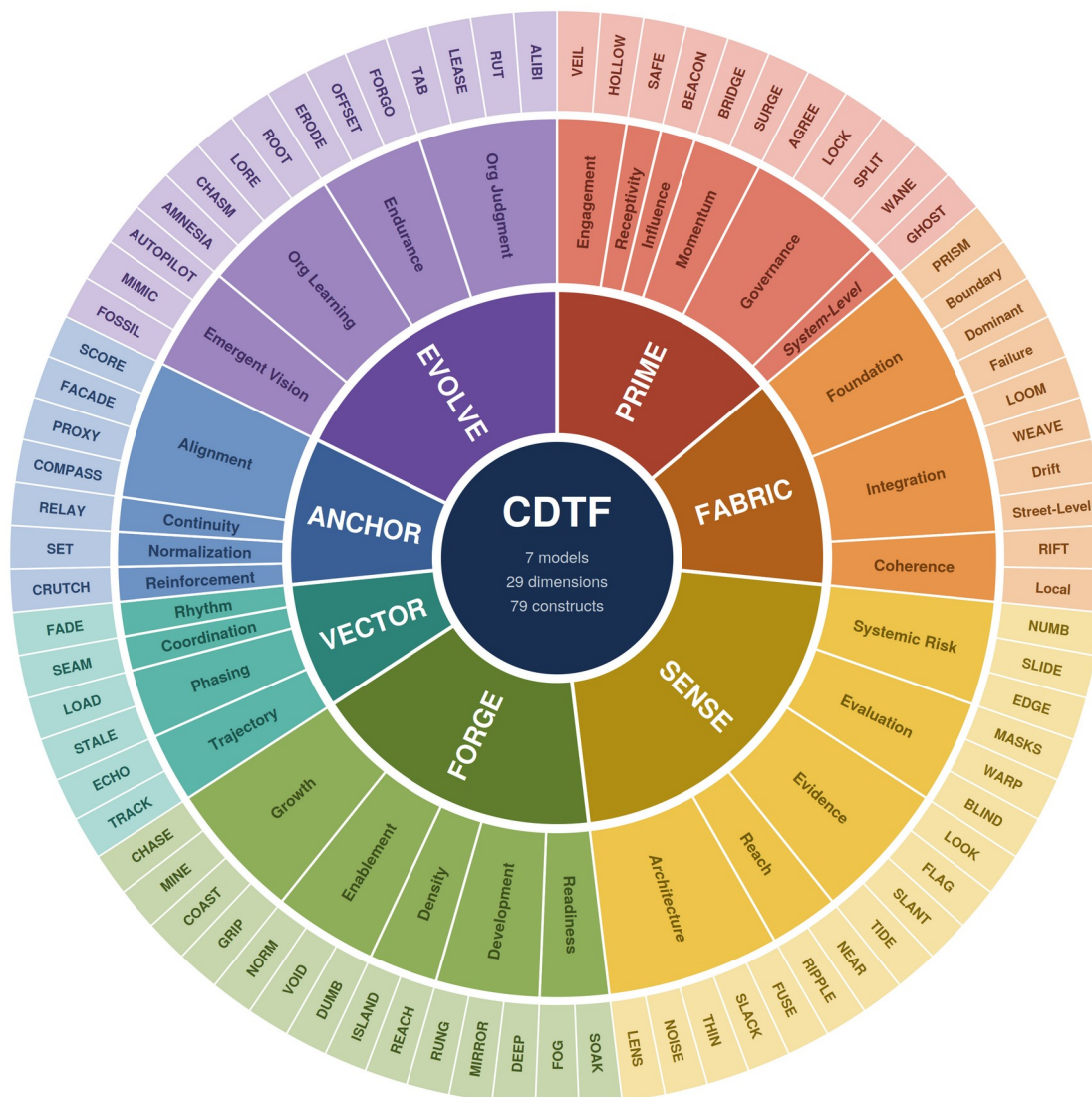
Each model contains dimensions, and each dimension organizes constructs that make the picture more specific. The models remain connected because conditions do not respect functional boundaries. Engagement affects what people are willing to say, interpretation

affects what workstreams build, capability affects what the organization can sustain, and execution generates evidence that may change the original diagnosis.

The figure below shows all three layers together. An interactive version of the wheel is available at TheCDTF.com.

The Capability-Driven Transformation Framework

Seven models each examine a different class of organizational condition. Within each model, dimensions group related conditions. Within each dimension, constructs name the specific conditions that shape the outcome.



7 models, 29 dimensions, and nearly 80 constructs. Interactive inventory at TheCDTF.com.

How a diagnosis changes the design

A framework proves its practical value when a different diagnosis produces a different course of action. Each situation below will be familiar, because on the surface it looks like the program is being run well. In each case, the explanation that is usually accepted directs attention toward the wrong intervention.

Situation one: the dashboard is green

WHAT LEADERS SEE	Adoption is reported at ninety-four percent, readiness scores are strong across every function, and the program remains on track.
THE INTERPRETATION THAT USUALLY FOLLOWS	The transformation is working. The remaining task is to finish the deployment and maintain momentum.
THE QUESTION CDTF ASKS	Is the evidence independent of the work being assessed, or is it produced by the same people whose performance, readiness, and adoption are being judged?
WHAT CHANGES IN DESIGN	The design shifts from collecting more of the same data to strengthening the evidence itself. Independent observation, operational evidence, anonymous channels, and comparison across sources are added so the picture can correct itself.
WHAT CHANGES IN EXECUTION	Progress reviews no longer treat a single reported measure as conclusive. A named owner is responsible for bringing forward evidence that could contradict what the reports say, and that evidence reaches the same decision forum as the dashboard.

The concern is not that people are dishonest. Most reporting is offered in good faith, which is why the problem is structural rather than personal. When every source of evidence shares an interest in the answer, the resulting picture can appear complete and internally consistent while remaining unable to correct itself.

Situation two: both workstreams are on track

WHAT LEADERS SEE	Two interdependent workstreams report green, and each is producing the agreed outputs on schedule.
THE INTERPRETATION THAT USUALLY FOLLOWS	Coordination is working. Any remaining differences will be resolved when the outputs are integrated.
THE QUESTION CDTF ASKS	Is anyone checking whether the workstreams still share compatible assumptions about what they owe each other, while there is still time to change course?
WHAT CHANGES IN DESIGN	The design shifts from evaluating each workstream independently to examining what passes between them. Assumptions, dependencies, and expectations are made explicit, and responsibility is assigned for the point where the two connect.
WHAT CHANGES IN EXECUTION	The workstreams' assumptions about each other get checked on a set cadence by a named owner, and differences go to the same forum that reviews status. Differences are treated as evidence about the design rather than as personality clashes or routine coordination problems.

Increasing the performance of either workstream does not resolve the condition because the failure doesn't sit within either team. It sits between them, which conventional reporting examines only after the separate outputs have been treated as healthy.

Situation three: the capability appears to be present

WHAT LEADERS SEE	The new way of operating is working, difficult cases are resolved, reports are produced, and teams receive the support they need.
THE INTERPRETATION THAT USUALLY FOLLOWS	The capability has been demonstrated, and the organization is ready to assume ownership.
THE QUESTION CDTF ASKS	Does the organization own the capability it appears to have, or does the capability remain concentrated in consultants, contractors, or a small group of specialists who will eventually leave?
WHAT CHANGES IN DESIGN	The design shifts from proving that the work can be completed to establishing where the ability to complete it resides. Capability transfer, decision rights, practice under real conditions, and evidence of independent performance become explicit design requirements.
WHAT CHANGES IN EXECUTION	The organization operates for a defined period without routine outside intervention before the support is withdrawn permanently. The resulting performance becomes the test of whether capability has actually been embedded.

Borrowed capability is difficult to detect while outside support remains active, because the tasks are genuinely getting done. The condition becomes visible when the support ends and the organization discovers the capability was never fully transferred.

Across all three situations, the value of diagnosis lies in the consequence. A more precise diagnosis changes the design, and the revised design changes what the program must do during execution.

What CDTF leaves in place and what it takes on

CDTF does not ask an organization to give up what it already does well. CDTF does not replace program management or technology delivery. Program management continues to organize delivery, and technology delivery continues to put the system into service. What CDTF changes is the organizational content those disciplines work with, meaning the conditions the program designs around and the evidence it accepts as sufficient.

These days, most conventional change management tends to center on the individual, asking whether people know the change is coming, want it to succeed, can do what the change requires, and have adopted it. CDTF centers on the organization, because an impact rarely stops where it starts. It travels vertically, changing what each level above and below must decide, supervise, sponsor, and measure. It travels horizontally, changing what neighboring functions receive and must absorb. It travels forward in time, because choices made during design determine what remains once the program ends. Communications, training, and engagement remain necessary. They become far more valuable when a diagnosis of those conditions shows where to aim them.

CDTF strengthens how a transformation is led and governed by improving the questions decision-makers ask, the evidence they require, and the design choices they make before an answer is treated as settled.

Where standard instruments reach their limits

Familiar tools remain useful when they address the condition that is actually present. Problems arise when the remedy is selected before the condition has been diagnosed.

- Training cannot resolve a condition that was never caused by a lack of knowledge or skill.
- A communications plan cannot reconcile functions that attach incompatible operational meanings to the same language.
- A readiness survey completed by the population being assessed cannot, by itself, test the accuracy of the picture the program is presenting.
- A responsibility matrix can assign tasks, but it cannot create stewardship or ensure that ownership survives a change in leadership.
- A sponsor briefing cannot compensate for scorecards, incentives, or operating pressures that reward behavior inconsistent with the transformation.

Some interventions can deepen the very condition they are intended to correct. Another listening session will not surface information that people have learned is unsafe to express. Extended go-live support can preserve performance while the organization never learns to work without it. A super-user network can concentrate expertise so effectively that the rest of the organization never has to build that expertise. These are organizational design problems, and each requires a response matched to the mechanism producing it.

A contemporary example: enterprise AI adoption

Enterprise AI adoption makes the difference between activity and organizational capability difficult to ignore, because the technology is advancing faster than many organizations can redesign decision rights, controls, operating routines, capability requirements, and learning mechanisms. The same distinction matters in any transformation that introduces new capability faster than the organization can absorb it. Approaches that appear successful in one setting may fail in another because the surrounding organizational conditions are different.

Organizations adopting AI often have to redefine roles, judgment standards, escalation rules, controls, and ways of working with little stable precedent to draw on. This places substantial demands on FORGE, which addresses the capability required to operate differently, and EVOLVE, which addresses how experience is converted into organizational intelligence as the technology and its uses continue to change.

Usage is not proof of organizational capability

When an organization ties performance ratings to AI usage, the intended goal is usually to accelerate adoption. The measure, however, changes behavior. People learn to produce the required activity, so logins, prompt counts, and weekly active use increase. The measure may improve at the same time that it becomes less informative, because it can no longer distinguish genuine value creation from compliance with the target.

Usage data still has value. It can show who has access, where experimentation is occurring, and which groups may need additional support. The problem begins when usage is treated as proof of adoption, capability, or performance. Those questions fall at different points on a progression that runs from access to durable capability.

Stage	What it establishes
Access	People can reach the approved tools and understand the basic conditions for using them.
Activity	People are using the tools with enough frequency to generate meaningful experience.
Useful application	The tools are being applied to work where they create genuine value rather than activity for its own sake.
Judgment	Users can recognize weak, incorrect, biased, or unsafe outputs and know when human judgment must override the tool's output.
Work redesign	Roles, processes, handoffs, and performance expectations have been reconsidered in light of how the tools change the work.
Governance	Decision rights, controls, evidence standards, and review routines have caught up with the new way of working.
Durable capability	The organization can continue to perform effectively without extraordinary support from a program team, vendor, or consulting partner.

A usage measure provides evidence about activity, which is the second point in the progression. It is often interpreted as evidence of durable capability, which is the seventh. No increase in reporting can close that gap, because more reporting of activity can't establish a capability the measure was never designed to assess.

Consultants and implementation partners can also make organizational capability appear more advanced than it is. They may operate the tools, tune models, resolve exceptions, and carry judgment that the internal organization has not yet developed. CDTF names this condition LEASE, Leased Expertise Assumed Same as Embedded Capability. The work is being completed, but the capability to complete it remains outside the organization. The relevant test is what the organization can still do when external support steps back. Outsourcing a capability can be a deliberate and sound choice. What matters is whether capability sits where the operating model requires it to sit, and whether the organization understands the dependencies that choice creates.

Designed for different organizations and transformations

CDTF is designed for use across organization types and industries because it examines organizational conditions rather than sector-specific procedures. The framework remains consistent, while its emphasis, evidence, language, constraints, and application change to fit the organization and the transformation. A healthcare system, government agency, manufacturer, financial institution, nonprofit organization, and technology company may use the same architecture while assigning very different weight to its models and drawing on different forms of operating evidence.

The same principle applies across transformation types. AI adoption, ERP implementation, operating-model redesign, merger integration, regulatory change, workforce transformation, and process modernization create different demands, but each requires leaders to understand the conditions that will shape whether the intended change can become part of how the organization actually operates.

The research behind each model

Each model has its own research base, drawn from several disciplines. The areas named below are a sample rather than a full account.

PRIME is built on diffusion of innovations, opinion leadership, and social influence. FABRIC is built on sensemaking and on research into why different groups read the same situation differently. SENSE is built on requisite variety, attention to weak signals, and measurement distortion. FORGE is built on absorptive capacity, the competency trap, and the balance between using existing expertise and developing new capability. VECTOR is built on escalation of commitment, sunk cost, and path dependence, as these patterns operate inside a live program. ANCHOR is built on goal displacement and institutionalization. EVOLVE is built on organizational learning, the difference between single-loop and double-loop learning, and how escalation of commitment can persist across successive transformations.

Several areas apply to more than one model. Book One shows how each research area shaped the framework.

Where the argument leads

A transformation can be managed competently and designed around an incomplete understanding of the organization because managing activity and diagnosing organizational conditions are different tasks. The conditions that shape the outcome often sit between the disciplines responsible for individual parts of the transformation. Those conditions can be examined, and a more precise diagnosis can lead to a different design, a different execution response, and a different organizational outcome.

This paper has introduced the framework and illustrated three situations in which CDTF changes the answer. It stops short of defining each construct or explaining how to apply them, and that's deliberate. Diagnostic precision depends on distinguishing conditions that appear similar, evaluating the evidence for competing explanations, and understanding how constructs interact. Book One develops that reasoning in depth.

Continue with Book One

Book One

Book One explains how the architecture was built, why its distinctions matter, and how its parts work together to do what none of them can do alone. Across its chapters, it moves from the multidisciplinary research foundation through the seven models, constructs, and diagnostic reasoning, then brings the framework together in an applied case. CDTF connects diagnosis, design, and execution. Book One establishes the diagnostic and design architecture while introducing the execution and operationalization responsibilities that later volumes will develop more fully. It is written for executives, transformation leaders, advisors, researchers, and other professionals seeking a clearer way to connect evidence, diagnosis, design, capability, and execution. Book One is expected in late 2026.

CDTF has been developed and refined by Allen C. Stines, Ph.D., over more than a decade, drawing on multidisciplinary research and more than twenty years of experience leading and advising enterprise transformation. His background spans organizational science, systems thinking, mathematics, statistics, cybernetics, business operations, and transformation.

Learn more at TheCDTF.com