Lifecycle by Design

INSIDE THE OPERATING MODELS OF IT SERVICE PROVIDERS

From Legacy Foundations to Modern Scale—
and Why the Structure Still Fails

Jeremiah Loscalzo
Lifecycle Authority Group

April 2024

Lifecycle by Design - Research Series

Inside the Operating Models of IT Service Providers

From Legacy Foundations to Modern Scale—and Why the Structure Still Fails

---

Introduction — The System Behind the Industry

The IT services industry has never been more capable.

Across outsourcing providers, global system integrators, solution providers, and emerging technology firms, organizations deliver complex, multi-domain solutions that span cloud, cybersecurity, data platforms, and AI-enabled operations. They operate at global scale, manage highly distributed environments, and support mission-critical systems for the world's largest enterprises. By most visible measures, the industry appears mature.

Internally, a different pattern emerges.

Despite decades of evolution, the same challenges persist. Margins fluctuate, delivery outcomes vary, and rework remains common. Senior expertise is repeatedly required to stabilize engagements that should have been predictable. Projects begin with clear intent yet require adjustment during execution, and delivery teams compensate for decisions made earlier in the lifecycle without having shaped those decisions themselves.

These conditions are widely recognized, but they are frequently misunderstood.

They are treated as failures of execution. Organizations respond by improving utilization, refining delivery processes, and introducing new tools and governance structures. These efforts produce incremental gains, but they do not resolve the underlying issue. Across providers—regardless of size, segment, or geography—the same patterns continue to appear.

The problem is not execution; it is the system in which execution occurs.

The operating model that governs most IT services organizations was not designed for the complexity it now supports. It originated in an earlier era of structured outsourcing, where environments were more stable, services could be decomposed into discrete components, and lifecycle transitions were relatively contained. Within those conditions, the model performed well by introducing discipline, enabling scale, and delivering predictable outcomes.

As the industry evolved, technologies changed, client expectations expanded, and services became continuous, interconnected, and increasingly dependent on coordination across domains. The operating model did not evolve with these changes. Instead, it was extended across new technologies, geographies, and service categories without fundamental redesign. Functional

separation remained, lifecycle handoffs persisted, and decision-making authority continued to be distributed across organizational boundaries.

What had once been manageable friction became structural strain.

This paper examines that strain across the industry as a whole.

Drawing on observations from multiple generations of providers—from legacy outsourcing firms such as EDS and Perot Systems, to global integrators including Infosys and NTT DATA, to solution providers such as CDW, Sirius, and Logicalis, and more recent entrants like Pellera Technologies (formerly known as Mainline Information Systems)—the analysis identifies patterns that persist regardless of business model or scale.

The objective is not to compare organizations or evaluate performance; it is to reveal the system they share.

Viewed in this way, the conclusion is direct. The industry has modernized its services, expanded its capabilities, and scaled its operations, but the operating model at its core remains largely unchanged. Until that model is reconsidered, performance will continue to depend on effort rather than design.

Table of Contents

Inside the Operating Models of IT Service Providers

1. Executive Summary — The Structural Reality of IT Services

The global IT services industry has evolved into a highly sophisticated system, delivering cloud transformation, cybersecurity, AI enablement, and continuous digital operations at scale. Leading providers—including global system integrators, outsourcing firms, solution providers, and emerging technology services organizations—operate with advanced capabilities, deep technical expertise, and mature delivery frameworks.

Yet despite this maturity, a consistent pattern persists.

Margins remain volatile. Delivery outcomes are inconsistent. Execution frequently depends on intervention rather than design.

These conditions are often attributed to delivery performance—utilization, staffing, or project management discipline. However, as observed across providers and generations, the root cause is structural.

The operating model that governs most IT services organizations was not designed for the complexity it now supports.

Across firms, critical decisions—scope definition, architecture, pricing, and delivery feasibility—are distributed across multiple functions with misaligned authority. Sales defines opportunity intent, solution teams assemble proposals under time pressure, and delivery organizations inherit commitments they did not fully shape. Each transition introduces interpretation. Each boundary introduces risk.

This pattern is not isolated to a specific segment of the market.

It appears consistently across:

- Legacy outsourcing providers such as EDS, Perot Systems, and Atos
- Global system integrators including Infosys, NTT DATA, and Unisys
- Solution providers such as Sirius, CDW, ePlus, and Logicalis
- Engineering-led and emerging firms including World Wide Technology and Pellera Technologies

The implication is clear:

The industry does not operate on multiple models.
It operates on one model—applied at different scales.

As complexity has increased, this model has not evolved structurally. It has expanded operationally.

The result is a system in which performance depends on coordination, escalation, and individual expertise rather than on integrated design.

This paper examines that system.

It does not evaluate companies or compare capabilities. Instead, it identifies the structural conditions that persist across the industry and explains why those conditions continue to produce inconsistent outcomes.

The conclusion is direct:

The industry has modernized its services.
It has not modernized its operating model.

2. The Model We Inherited

How Legacy Outsourcing Structures Became the Foundation of Modern IT Services

The operating model that governs today's IT services industry originated in an earlier era—one defined by structured outsourcing, predictable workloads, and relatively contained technical environments.

Organizations such as EDS, Perot Systems, and later Atos pioneered a model designed to deliver efficiency, control, and scalability in enterprise IT operations. Work was organized into functional domains—compute, storage, network, and applications—each managed by specialized teams. Contracts defined scope explicitly, and delivery followed a largely linear progression from design to execution.

This model introduced discipline into a previously fragmented environment. It enabled organizations to standardize delivery, reduce operational costs, and scale services across large client environments. For its time, it was highly effective.

Its effectiveness, however, was based on a set of underlying assumptions.

First, complexity was relatively stable. Enterprise environments changed slowly, and solutions could be designed with a high degree of certainty before execution began.

Second, services could be decomposed into discrete components. Work could be segmented into towers, with each function responsible for a defined portion of the solution.

Third, lifecycle transitions were manageable. The movement from sales to design to delivery introduced some friction, but that friction was limited by the relatively linear nature of the work.

Within these conditions, the model performed well.

Ownership could be distributed across functions without significantly impacting outcomes. Architectural assumptions remained valid throughout delivery. Contracts aligned closely with execution. And learning, while not formally structured, was sufficient to sustain operational performance.

Over time, this model became institutionalized.

As the industry expanded, new providers adopted similar structures. Global delivery models extended the same approach across geographies. Solution providers and channel-led firms applied it to integration services. Even as new technologies emerged, the underlying operating model remained consistent.

What began as a fit-for-purpose design became the default structure for the industry.

The problem is not that this model was flawed.

The problem is that it was never redesigned.

3. The Model Expands Without Structural Evolution

How a single operating model scaled across the entire industry

As demand for technology services expanded, the industry did not replace its operating model. It extended it.

Global system integrators such as Infosys, NTT DATA, and Unisys scaled the outsourcing model into industrialized delivery systems. Work was distributed across global delivery centers, process maturity increased, and governance frameworks became more sophisticated. These organizations achieved scale by replicating the same structural design across geographies and service lines.

At the same time, solution providers—including Sirius Computer Solutions, CDW, ePlus, and Logicalis—extended the model into client-facing integration services. Their advantage was speed, proximity, and solution assembly. They operated closer to the client, often with tighter coordination between sales and engineering, but the underlying structure remained consistent. Sales originated demand, solution teams assembled proposals, and delivery executed engagements shaped earlier in the lifecycle.

Engineering-led firms such as World Wide Technology added deeper technical capability and integration expertise. Emerging providers such as Pellera Technologies introduced modern tooling, cloud-native approaches, and more agile execution models. These firms appeared structurally different, but beneath the surface they operated within the same boundaries: functional separation, lifecycle transitions, and distributed decision authority.

Across all segments, the services themselves changed.

Infrastructure gave way to cloud platforms. Security became embedded across every domain. Data and AI introduced continuous processing and feedback loops. Clients shifted from discrete projects to ongoing outcomes.

The operating model did not change with them.

Work continued to be organized into domains. Sales, solutioning, and delivery remained distinct stages. Proposals were assembled through coordination across functions. Delivery teams inherited commitments shaped earlier in the lifecycle. Each extension of the model increased its reach without resolving its structural limitations.

The result was a system that scaled horizontally but not structurally.

As long as complexity remained manageable, coordination compensated for misalignment. As complexity increased, coordination became insufficient. The model began to rely more heavily on individual expertise, escalation paths, and informal alignment mechanisms to maintain performance.

This dependence is not a characteristic of a specific company or segment. It is a property of the model itself.

4. A Cross-Generational, Cross-Segment View

What the industry reveals when viewed as a single system

The IT services industry is often described through its differences.

Outsourcing providers emphasize long-term operational control. Global system integrators operate at scale across complex transformation programs. Solution providers prioritize speed, proximity, and commercial agility. Engineering-led firms differentiate through deep technical capability. Emerging providers position themselves as modern alternatives, built for cloud-native environments and continuous delivery.

These distinctions are real at the surface.

Structurally, they converge.

Across generations and segments, the same operating model persists. The differences lie in emphasis and execution, not in design. When examined through a structural lens, each segment reflects the same underlying system: functional separation, lifecycle handoffs, and distributed decision authority.

The continuity becomes clear when the industry is viewed as a whole rather than as a collection of individual firms.

Table — Structural Patterns Across the Industry

Segment	Representative Providers	Structural Design	Execution Emphasis
Legacy Outsourcing	EDS, Perot Systems, Atos	Tower-based delivery, contract-defined scope, linear lifecycle	Control, predictability, operational efficiency
Global System Integrators	Infosys, NTT DATA, Unisys	Global delivery, multi-tower integration, layered governance	Scale, process maturity, transformation delivery
Solution Providers / VARs	Sirius, CDW, ePlus, Logicalis	Sales-led motion, rapid solution assembly, localized delivery	Speed, client proximity, commercial agility
Engineering-Led / Emerging Providers	World Wide Technology, Pellera Technologies	Deep technical integration, modern tooling, flexible execution	Capability depth, solution complexity, adaptability

---

Despite variation in execution, each segment operates within the same structural boundaries.

Work is decomposed into domains. Sales originates demand. Solutioning assembles the response. Delivery executes the engagement. Operations sustain the environment. Each stage is owned by a different function, and each transition introduces a boundary across which intent must be interpreted.

These boundaries define the system.

They also define its limitations.

Across all segments, similar behaviors emerge—not because organizations lack capability, but because the structure produces them. In outsourcing environments, contractual rigor provides control but does not eliminate interpretation between stages. In global integrators, governance structures manage complexity but introduce additional layers between decision and execution. In solution providers, speed increases commercial effectiveness but compresses the time available for architectural validation. In engineering-led firms, technical depth compensates for structural gaps but does not remove them.

Each model compensates differently. None resolves the underlying condition.

This continuity across decades is not coincidental. From early outsourcing models to modern cloud and AI-driven services, the same structural characteristics remain in place. The

technologies have changed, the expectations have expanded, and the operating model has remained largely intact.

The implication is direct. The IT services industry does not operate on multiple fundamentally different models. It operates on a single model applied across different contexts, scales, and technologies.

Understanding this continuity reframes the problem. What appear to be isolated challenges—margin variability, delivery inconsistency, and reliance on heroics—are predictable outcomes of a shared system.

5. Where the Model Breaks — Structural Failure Points Across the Lifecycle

The limitations of the operating model do not appear in a single moment. They emerge at predictable points across the lifecycle of an engagement.

From opportunity identification through delivery and operations, the same structural conditions introduce risk. These conditions are not failures of execution; they are the natural consequences of a system in which responsibility is distributed, decisions are made across boundaries, and no single authority governs continuity from intent to outcome.

Four failure points persist across all segments and generations.

5.1 Fragmented Authority

In most IT services organizations, no single function owns the full outcome of an engagement.

Sales defines opportunity intent and commercial positioning. Architects and solution teams design the approach. Pricing and commercial terms are negotiated through a combination of sales and finance. Delivery teams are responsible for execution. Operations sustain the environment after implementation.

Each function performs its role effectively.

The problem is that ownership does not extend across the lifecycle.

When authority is fragmented, decisions are optimized locally rather than systemically. Sales prioritizes revenue and client acquisition. Solution teams balance technical feasibility with speed. Delivery focuses on execution within the constraints provided. Operations manages stability and ongoing performance.

No function is structurally accountable for aligning these decisions end to end.

Misalignment is not identified early; it is absorbed during execution.

5.2 The Translation Problem

At each transition in the lifecycle, intent must be transferred from one function to another.

Opportunity intent becomes solution design. Solution design becomes contractual scope. Contractual scope becomes delivery execution. Delivery execution becomes operational state.

Each transition introduces interpretation.

Even when documentation is thorough and communication is strong, differences in perspective, incentives, and context lead to variation in how intent is understood. Architectural assumptions may be incomplete. Scope definitions may be interpreted differently. Client expectations may evolve during engagement.

The system relies on individuals to bridge these gaps.

Over time, small variations accumulate. By the midpoint of delivery, the executed solution may differ materially from the original intent—not because of failure, but because of translation across boundaries.

5.3 Pattern Immaturity at Scale

Most services organizations develop experience through repeated engagements. Similar solutions are delivered multiple times across different clients. Lessons are learned. Improvements are identified.

This experience is rarely converted into enforceable structure.

Service patterns—defined architectures, scope models, pricing frameworks, and delivery methods—often exist informally. They are used when convenient, adapted when necessary, and bypassed when timelines require speed.

Without formal ownership and enforcement, patterns remain optional.

Each engagement is partially reinvented, even when similar work has been performed many times before. Estimation accuracy varies. Architectural consistency declines. Delivery efficiency depends on the experience of the individuals involved.

Scale increases activity, but it does not guarantee repeatability.

5.4 Learning That Does Not Compound

Delivery organizations generate continuous insight.

Projects reveal gaps in architecture, inaccuracies in estimation, and opportunities for improvement. Engineers and delivery leaders gain a deeper understanding of what works and what does not.

In most operating models, this learning remains local.

It may improve performance within a team or a specific account, but it does not consistently influence how future opportunities are shaped, how proposals are assembled, or how pricing models are constructed. The feedback loop between delivery, solutioning, and sales is weak or informal.

The same issues recur.

Organizations learn, but the system does not.

6. The Consequence — Scale Without Consistency

When structural failure points persist, performance becomes conditional.

Organizations can scale revenue, expand service offerings, and increase technical capability, but they cannot ensure consistent outcomes across engagements. Each project becomes dependent on coordination between functions and the experience of the individuals involved.

This produces a characteristic pattern.

High-performing teams deliver strong results. Experienced leaders stabilize complex engagements. Critical projects receive attention and resources. At the same time, variability persists across the broader portfolio. Margins fluctuate, timelines shift, and delivery effort increases beyond initial expectations.

The system functions, but it does not operate predictably.

This condition is often masked by effort. Senior engineers are introduced to resolve issues. Delivery leaders intervene to realign scope and expectations. Governance structures escalate decisions that could not be resolved earlier in the lifecycle.

Recovery becomes normalized

Table — From Structure to Outcome

Structural Condition	Operational Response	Observed Outcome
Fragmented authority	Leadership alignment and escalation	Decisions corrected late in lifecycle
Translation gaps	Increased coordination and communication	Rework and scope adjustment during delivery
Pattern immaturity	Reliance on experienced individuals	Variable delivery efficiency and quality
Non-compounding learning	Local process improvements	Repetition of issues across engagements

---

These responses enable delivery, but they do not eliminate the underlying causes.

The most visible symptom is reliance on what is often described as “heroic delivery.” Complex engagements succeed because experienced individuals compensate for structural misalignment. They identify gaps, resolve conflicts, and adapt execution in real time.

This capability is valuable.

It is not scalable.

As organizations grow, the number of engagements increases faster than the availability of individuals capable of performing this role. The system becomes increasingly dependent on a limited set of experts, and variability expands across the broader delivery base.

Scale is achieved, but consistency is not.

---

7. Why the Model Persisted

Given these limitations, the persistence of the model requires explanation.

The operating model works well enough to support growth.

It enables organizations to scale revenue, expand into new markets, and deliver a wide range of services. Functional specialization allows for depth of expertise. Distributed ownership enables

parallel execution across large portfolios of work. Governance structures provide mechanisms for managing complexity as it arises.

For many years, these characteristics were sufficient.

The cost of structural misalignment was absorbed through margin compression, delivery effort, and organizational overhead. As long as demand for services remained strong, firms could grow despite inefficiencies in how work was designed and executed.

Several reinforcing factors sustained the model.

Financial structures prioritize revenue growth over margin stability, encouraging rapid deal assembly and acceptance of execution risk. Organizational design reinforces functional separation, embedding lifecycle handoffs into the structure itself. Governance overlays—PMO, customer success, and coordination roles—address symptoms without changing underlying conditions. Incremental optimization strengthens individual functions without redesigning the system.

These mechanisms stabilize the model while increasing its complexity.

Organizations respond to challenges by adding layers rather than redesigning the structure. Each addition improves control locally but increases coordination requirements across the system. Over time, the model evolves through accumulation rather than transformation.

Changing this structure is difficult. It requires redefining decision authority, realigning incentives, and integrating functions that have historically operated independently. It challenges established organizational boundaries and shifts the focus from functional optimization to system design.

As a result, most organizations continue to operate within the inherited model, adapting around its limitations rather than replacing it.

For many years, this approach was viable.

The question is whether it remains viable under the conditions the industry now faces.

---

8. Why the Current Era Exposes the Gap

Why modern demands are incompatible with the inherited structure

For much of its history, the IT services operating model was able to absorb its own limitations. Complexity increased gradually, and organizations compensated through coordination, governance, and experience. The gap between structure and execution remained manageable.

That condition no longer holds.

The current era introduces a different type of demand—faster, more interconnected, and less tolerant of fragmentation across the lifecycle. Three forces are exposing the limits of the inherited model.

The first is the compression of time between intent and execution.

Cloud platforms, automation, and AI-driven services have reduced the distance between design and deployment. Clients expect rapid translation from requirement to outcome, often within iterative delivery cycles. Decisions that were previously sequenced can no longer be separated without introducing delay or misalignment. A model built on distributed decision-making struggles to operate at this speed without increasing coordination overhead.

The second is the rise of platform complexity.

Modern solutions are no longer confined to a single domain. Cloud, security, networking, data, and application layers are tightly integrated, and changes in one domain affect others. This interdependence requires coordinated design and execution across functions that were historically separated. The model was designed for decomposition; the environment now requires integration.

The third is the shift from project-based delivery to lifecycle accountability.

Clients increasingly expect providers to deliver outcomes, not just implementations. Advisory, implementation, and managed services now operate as a continuous system. This requires continuity of intent from initial engagement through ongoing operation. A model built on handoffs introduces discontinuity where continuity is required.

These forces do not introduce new problems; they amplify existing ones.

Fragmented authority becomes more visible when decisions must be made quickly and in coordination. Translation gaps become more costly as misalignments propagate across interconnected systems. Pattern immaturity limits repeatability. Learning that does not compound slows adaptation.

The model does not fail abruptly. It becomes increasingly difficult to operate.

Organizations respond by adding coordination, governance, and specialized roles. These additions provide short-term control but increase system complexity, slowing decision-making and reinforcing dependence on individual expertise.

The gap between what the system was designed to handle and what it is now required to deliver continues to widen.

9. The Structural Shift Required

What must change for performance to become consistent

Addressing the limitations of the current model does not require incremental improvement. It requires structural change.

The core issue is not capability, process maturity, or technology adoption. It is the absence of an operating model designed to align decisions, execution, and learning across the full lifecycle of a service.

A different structure must satisfy four conditions.

First, authority must be aligned across the lifecycle.

Decisions that define an engagement—opportunity qualification, architectural design, pricing, and delivery feasibility—cannot remain distributed across functions without shared accountability. Each stage of the lifecycle requires clear decision ownership, but those decisions must also be connected. Authority must be defined not only within functions but across them, ensuring that what is sold, designed, and delivered reflects a single, coherent intent.

Second, patterns must be owned and enforced.

Experience must be converted into structured, repeatable design. Service patterns—covering architecture, scope, pricing, and delivery methods—must become authoritative rather than optional. This does not eliminate flexibility; it establishes a baseline from which variation is intentional rather than reactive. Without enforceable patterns, scale produces variability rather than consistency.

Third, commercial and delivery design must be integrated.

The separation between how work is sold and how it is executed introduces risk at every stage of the lifecycle. Proposal development must incorporate architectural validation and delivery acceptance before commitments are made. Pricing must reflect real delivery conditions, not assumptions formed under time pressure. Integration of these activities reduces the need for correction during execution.

Fourth, learning must be structured to compound.

Delivery generates continuous insight, but that insight must be captured, formalized, and reintegrated into future opportunities. Feedback loops between delivery, solutioning, and sales must be explicit and repeatable. Patterns must evolve based on observed outcomes, and those updates must influence how future work is designed and priced. Learning becomes a system property rather than an individual capability.

Together, these conditions define a shift from a functionally optimized model to a lifecycle-designed system.

In such a system, performance is not dependent on coordination across boundaries because those boundaries are structurally aligned. Decisions are made with visibility into their downstream impact. Execution reflects the intent established earlier in the lifecycle. Learning improves future outcomes by design rather than by exception.

The objective is not to eliminate complexity.

It is to ensure that complexity is managed through structure rather than absorbed through effort.

---

10. Conclusion — From Scaled Models to Designed Systems

The IT services industry has achieved extraordinary scale.

It has expanded across technologies, geographies, and service categories. It has developed deep technical expertise, mature delivery processes, and the ability to support some of the most complex environments in operation today. By every conventional measure, it is an advanced and capable industry.

Yet the same structural limitations persist.

Across providers, outcomes remain inconsistent. Margins fluctuate. Delivery depends on intervention. Organizations compensate for misalignment rather than preventing it. These conditions are not the result of insufficient capability; they are the result of a model that has been extended beyond the conditions for which it was designed.

The industry did not evolve its operating model.

It scaled it.

For decades, this approach was sufficient. Growth masked inefficiency, and coordination compensated for structural gaps. The cost of misalignment was absorbed through delivery effort and organizational overhead.

That balance is changing.

Modern demands—faster execution cycles, integrated platforms, and lifecycle accountability—require a level of alignment the inherited model does not provide. The gap between structure and expectation is no longer hidden. It is visible in performance, financial outcomes, and the increasing effort required to sustain delivery.

The implication is not that the industry must abandon its foundations.

It must redesign them.

Performance in the next generation of IT services will not be determined solely by technical capability or delivery excellence. It will be determined by operating model design—by the ability to align authority, integrate execution, and compound learning across the lifecycle of every engagement.

Organizations that continue to rely on fragmented structures will sustain performance through effort, coordination, and expertise. Organizations that redesign their operating models will achieve performance through structure.

The difference between these approaches is not incremental. It is fundamental.

It is the difference between systems that scale and systems that endure.

The difference is not in capability. It is in design.

11. Research Approach and Sources — Basis for Structural Conclusions

This paper is based on direct observation of operating models across multiple segments of the IT services industry, combined with publicly available disclosures from leading service providers.

The analysis reflects patterns observed across:

- Legacy outsourcing providers
- Global system integrators
- Solution providers and channel-led organizations
- Engineering-led and emerging service firms

These observations were not collected as isolated case studies. They were derived from repeated exposure to similar structural conditions across different organizations, operating environments, and market contexts.

In each case, the focus was not on individual company performance, but on how work moved through the lifecycle of a service—from initial client engagement through solution design, delivery, and ongoing operations.

Across these environments, consistent patterns emerged.

Decisions that define scope, architecture, pricing, and delivery feasibility are distributed across functions. Responsibility for execution is separated from responsibility for shaping commitments. Learning generated during delivery is not consistently reintegrated into future opportunities.

These conditions appear regardless of company size, geographic model, technology domain, or level of process maturity.

The consistency of these patterns across organizations forms the basis for the structural conclusions presented in this paper.

Sources of Observation

The findings reflect experience across:

- Large-scale outsourcing and managed services environments
- Global system integration and transformation programs
- Solution provider and channel-led service delivery models
- Practice and portfolio development across multiple organizations

This includes direct involvement in:

- Service portfolio design and commercialization
- Bid and proposal development for complex engagements
- Delivery leadership across multi-domain programs
- Organizational restructuring of service practices and operating models

These roles provide visibility into how decisions are made at each stage of the lifecycle and how those decisions affect execution outcomes.

Use of Public Disclosures

Public disclosures were used to validate scale, operating structure, and financial characteristics of IT service providers.

These sources include:

- Annual reports and investor presentations
- SEC filings, including Forms 10-K and 20-F
- Segment reporting and service line disclosures

Across these materials, a consistent structural pattern is visible.

Organizations describe their businesses in terms of service lines, functional domains, and global delivery constructs. Revenue is reported across project-based and managed services categories. Delivery is consistently positioned as a distributed function operating across regions and specialized teams.

These disclosures do not define the conclusions of this paper. They reinforce them.

Representative Public Sources

The following materials were referenced to validate industry structure and operating characteristics:

Global System Integrators and Outsourcing Providers

- Infosys Limited — Form 20-F
- NTT DATA Group Corporation — Annual Report
- Atos SE — Universal Registration Document / Annual Report
- Unisys Corporation — Form 10-K

Solution Providers and Channel-Led Firms

- CDW Corporation — Form 10-K
- ePlus inc. — Form 10-K
- Logicalis Group — Annual disclosures via Datatec Limited

Engineering-Led and Private Providers

- World Wide Technology — publicly available materials and industry disclosures

Legacy Context

- Historical materials related to EDS and Perot Systems

Method of Interpretation

These sources were not used to compare companies or evaluate performance.

They were used to identify structural consistency across the industry, including:

- Separation of sales, solutioning, and delivery responsibilities
- Organization around service lines or technology domains
- Scaling through distributed and global delivery models
- Revenue structures tied to project-based and managed services

By examining these characteristics across providers, a consistent operating model becomes visible.

The objective was not to identify best practices.

It was to identify what persists.

Limitations and Scope

This analysis does not attempt to measure performance quantitatively across firms, nor does it rank organizations or assess competitive positioning.

The conclusions are structural rather than statistical.

They are intended to explain recurring patterns observed across the industry, not to evaluate individual companies or engagements.

Closing Note on Interpretation

The intent of this research is not to critique specific organizations.

The patterns described are not the result of poor execution or insufficient capability. They are the result of a model that has been widely adopted and extended across the industry.

Understanding that model—and its limitations—is a prerequisite to improving performance.

This analysis incorporates publicly available disclosures from leading IT service providers, including annual reports, SEC filings, and investor materials, to validate structural observations.