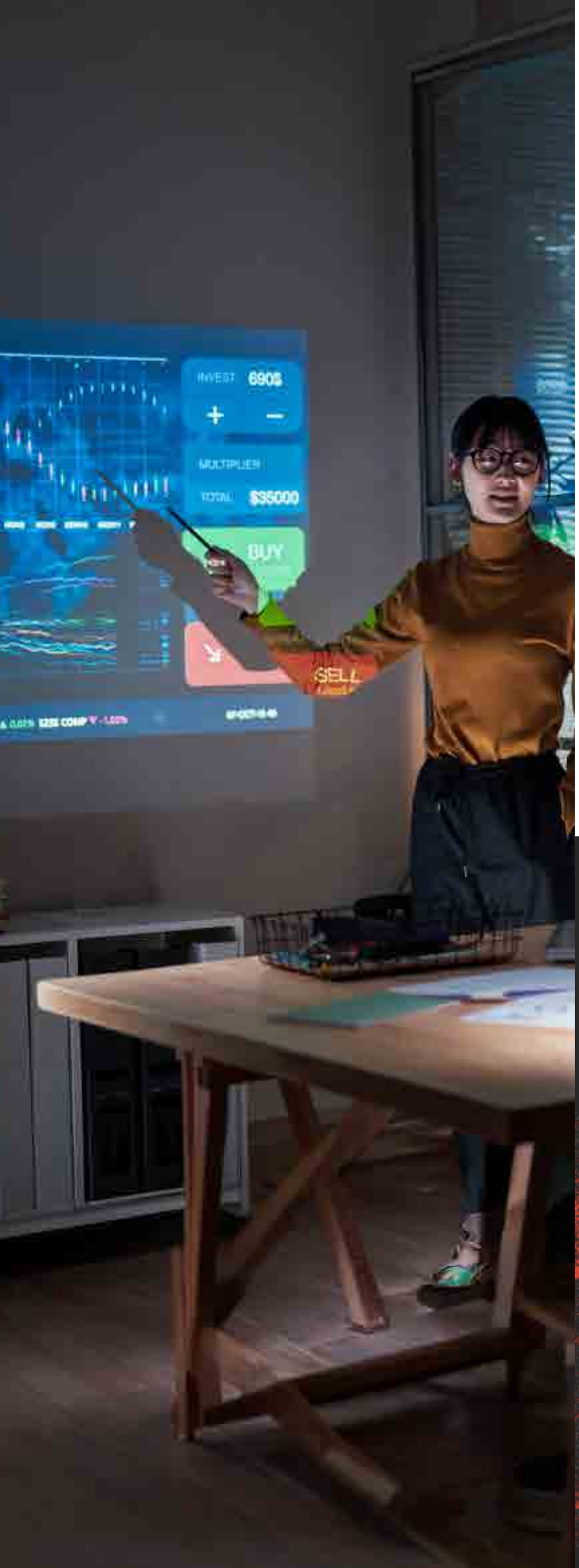


LTM

WHITEPAPER

Transforming from Salesforce CPQ to Agentforce Revenue Management: Owning Outcomes Through an Agent-First Approach Beyond Lift-and-Shift

Apurva Anil Durge
Principal – Architecture Salesforce, LTM

Table of Contents

Executive Summary

This white paper reimagines the transformation journey from the legacy Salesforce CPQ, formed by the acquisition of Steelbrick CPQ, to Agentforce Revenue Management (ARM) - the next-generation, agent-first revenue operations architecture within Salesforce's evolving Revenue Management ecosystem.

Rather than treating this transition as a technical upgrade or pure lift and shift, this paper highlights why ARM represents a **paradigm shift**: revenue processes become agentic, autonomous, and intelligent. With LLM driven **agents** capable of reasoning, acting, orchestrating workflows, and continuously optimising outcomes, organisations can elevate their revenue operations far beyond what CPQ ever allowed.

ARM introduces a unified, AI native foundation designed to automate decisions, accelerate sales cycles, reduce operational overhead, and create high fidelity customer and seller experiences. Business leaders and architects will find strategic insights to future proof their revenue lifecycle with an agent centric operating model.
The whitepaper aims to target enterprise business leaders who sense that traditional CPQ is reaching its limits and seek to shorten the sales cycle, improve forecast accuracy, or scale revenue without increasing sales headcount. The paper is also highly valuable for Salesforce architects, platform owners and Salesforce CoEs who want to gain clarity on how to modernise their CPQ-led revenue implementations, thinking beyond lift-&-shift.

With Salesforce's strategic shift towards Agentforce and AI-led solutions, Salesforce CPQ is no longer a future for enterprises that adopted it years ago and now struggle to understand how to upgrade their revenue systems. This paper helps them understand what Salesforce is really building towards.
Revenue complexity is growing day by day. Buyers expect personalised experiences and quick interactions, whereas sellers are overwhelmed by manual steps, approvals, and broken, fragmented systems. This paper demonstrates how agentic revenue operations can directly address these challenges rather than masking them through layers of automation.

Introduction

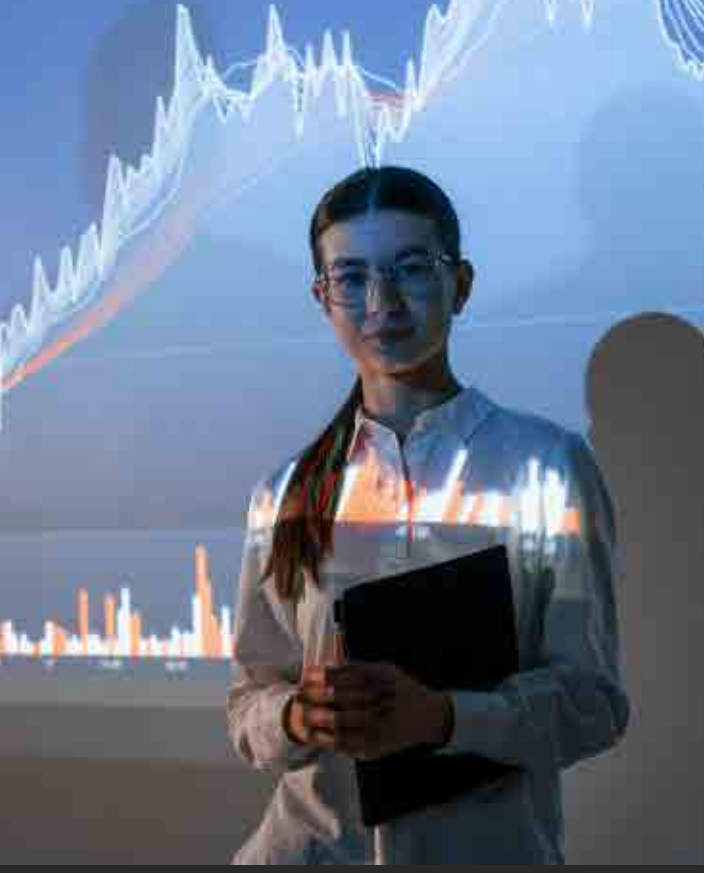
For over a decade, **Salesforce CPQ** served as the backbone for quoting processes. It gave revenue operations a genuine foothold inside CRM. CPQ centralised logic in a way that made it audible, repeatable and trainable. While it is a powerful tool for rule based configuration and pricing, Salesforce CPQ is constrained by:

- Rigid rules architectures
- Technical debt due to high customisation
- Limited automation beyond guided selling
- Siloed integration with Billing engines and external CLM engines
- A lack of AI intelligence and reasoning

In most CPQ implementations reviewed, the average quote cycle time recorded in the system significantly underestimates the actual elapsed time because 'pending approval' intervals, when the process sits in someone's inbox, are excluded from cycle time metrics. The real number is often 2-3 times higher than what leadership sees on dashboards.

Agentforce Revenue Management is Salesforce's re-establishment of the revenue lifecycle as an agent-orchestrated process. The word 'agent' here carries a specific meaning. Salesforce defines agents as AI-powered actors that can perceive events in the data environment, reason about them using configured objectives and constraints, take autonomous action within defined parameters, and escalate to humans when necessary.
This is meaningfully different from workflow automation, which is rule-based and deterministic. An agent that handles quote modifications is not simply executing a lookup table. It evaluates the modification against pricing strategy, contract history, customer risk profile, and current margin targets, and makes a contextual decision on whether to approve, adjust, or escalate.

It is important to note that, with this upgrade, the underlying commercial logic remains unchanged. Pricing strategy, approval authorities, and contract terms remain the purview of humans. What really changes is that the system now executes against those decisions intelligently. People still need to set the rules, but they need not be present at every transaction where rules apply.



Business Drivers for Transformation

After spending years designing and fixing revenue stacks built on Salesforce CPQ, one hard truth keeps repeating itself: Salesforce CPQ was built to enforce rules, not to think and optimise outcomes autonomously. Most large enterprises didn't hit CPQ limits at the start, but they did after scale, complexity, and market changes exposed the limitations of static rule engines. ARM exists because of what we repeatedly encountered in those moments.

Here are some of the challenges surfaced, which became drivers of the transformation:

Agentic Automation Across the Revenue Lifecycle

Salesforce CPQ was not designed to execute revenue work end-to-end without constant human orchestration. For one of the global SaaS clients, a simple enterprise deal required.

- Sales team configuring products
- Deal desk validating pricing, discounts, and ensuring profitability
- Legal team drafting terms of contract
- Finance team reconciling co-terms and billing schedules

Most of the above steps were done within separate systems and by actors connected through email, Salesforce tasks, or manual hand-offs.

ARM does exactly what legacy Salesforce CPQ lacked. They don't just calculate, they act:

- Configure the deal based on historical win patterns
- Generate quotes and contracts
- Adjust pricing dynamically
- Progress the transaction without waiting for humans to translate context

Reduced Customisation and Technical Debt

For those who live in a mature Salesforce CPQ org, it will sound familiar:

- Numerous product and pricing rules overlap each other
- Twin field-dependent product rules
- Constraint rules no one remembers authoring
- Plugins are built without realising their need and true potential

At one enterprise manufacturer, 70% of CPQ change requests were classified as 'high risk' because no one could predict the downstream impact. More than 50% of the rules were not sequenced and overlapped, resulting in incorrect results.

ARM replaces brittle mechanics with policy intent. You can define margin thresholds, Compliance constraints, and revenue policies. AI Agents interpret those policies in real time, resulting in:

- Fewer rules with reduced deployment timelines
- Dramatically reduced regression risks

Cross System Intelligence

Salesforce CPQ assumes the world begins and ends with a quote. However, in complex enterprises, the reality is far beyond quotes. The true picture can be depicted as:

- Fewer rules with reduced deployment timelines
- Dramatically reduced regression risks

We've repeatedly seen CPQ customisations explode when orders need re-rating post-signature, billing needs re-categorising mid-cycle, or the ERP rejects something CPQ has already approved.
ARM agents operate across the full chain: Quote -> CLM ->Order -> Billing -> ERP

ARM replaces brittle mechanics with policy intent. You can define margin thresholds, Compliance constraints, and revenue policies. AI Agents interpret those policies in real time, resulting in:

- Detect misaligned start dates between Order and CLM
- Re calculate billing schedules
- Amend contracts
- Sync ERP before invoicing errors occur

That is a cross system intelligence, not an integration theatre.

CPQ v/s ARM, and Can They Co-Exist in the Same Org?

ARM doesn't replace CPQ rules. It supersedes them with reasoning capable agents that understand product constraints, pricing strategy, margin tolerance, approval policies, and customer history.ically happens at the order or contract handoff, ensuring data consistency without forcing a full rip and replace.

Capability	Salesforce CPQ	Agentforce Revenue Management (ARM)
Architecture	Apex heavy, Packaged rules	AI native, Agent orchestrated
Product Model	Local product rules	Globalised catalogue with reasoning
Automation	Guided selling	Autonomous agents performing tasks
Intelligence	Rule-based	LLM powered decisioning & predictions
Scalability	Limited performance at scale	Event driven, High volume capable
Extensibility	Code first	Policy driven + agent driven
Integrations	Siloed	Natively orchestrated across the revenue stack

Many organisations have challenged the coexistence of Salesforce CPQ and ARM in the same org, and it has been the first step in many transformations. Organisations worry about the immediate impact on user experience and resistance to change, and so expect the best of both worlds until users realise the full potential of ARM.

Salesforce CPQ and ARM can comfortably co exist in the same Salesforce org when each is used for what it does best. For one of the enterprise organisations, we leveraged CPQ to serve as the frontline tool for sales teams, handling complex product configuration, pricing, discounting, and quote generation. ARM, on the other hand, took over once the deal was won, focusing on contract execution, invoicing, renewals, and lifecycle revenue operations.

This separation could clarify rather than duplication. Sales teams stay productive in CPQ without being burdened by downstream finance or contract management processes. In contrast, finance and operations teams benefit from a more robust and automated revenue backbone in Agentforce. Integration between the two typically happens at the order or contract handoff, ensuring data consistency without forcing a full rip and replace.

Transformation Strategies: Approaches for Successful Migration and Modernisation

A successful transition from CPQ to ARM requires a holistic transformation strategy rather than simply replicating existing processes. When we first started talking about migrating from CPQ to RCA, the early instinct was simple: "Let's recreate what we have today—just on the new platform."

That approach nearly trapped us in the same complexity we were trying to leave behind. Here are the strategies that made the biggest difference, along with the missteps that taught us why they matter.

1 Redesign Rules into Agentic Logic and not port them

In CPQ, rules are everything: eligibility rules, pricing rules, constraint rules, discount exceptions, and approval routing. Instead of porting thousands of CPQ rules, shift to agent policies and reasoning prompts.

In one programme, over 70% of legacy product and pricing rules were removed by allowing agents to infer behaviour from policy context. The result was a leaner model that was easier to extend and far less fragile.

2 Define an Agent First Operating Model

While we build agents to drive work autonomously, people feel worried that their jobs will disappear. But ARM changes how teams work, not just tools. Don't sell "replacement." Sell "relief." People adopt faster when they see agents removing the painful parts of their day.

3 Simplify Product & Pricing Structures

Years of CPQ usage usually leave behind bloated catalogues and tangled bundles—successful transformations started by consolidating products into a clean, central knowledge layer that agents could reason over. In several cases, duplicate SKUs surfaced only once agents began producing inconsistent answers, forcing long-overdue cleanup.

4 Establish ARM Governance & Policy Framework

Governance works best when expressed as clear, natural-language policies for discounts, approvals, margins, and compliance. Centralising these policies allowed teams to change business rules quickly, sometimes during a quarter, without waiting for code deployments or release cycles.

5 Shift to Event-Driven Orchestration

CPQ environments often end up with a maze of flows, triggers, and middleware doing invisible gymnastics. In one instance, we tried to preserve old automation by rebuilding flows "as-is." It worked right away, but debugging became a nightmare because the agent and flows were following each other's steps.

Rather than chaining flows and triggers, use business events (Opportunity → Quote → Order → Billing) to drive agent actions. This approach reduced automation conflicts and made revenue processes easier to trace end-to-end, especially during troubleshooting.

6 Replace Custom Code with Agent Automation

Much CPQ custom code exists to compensate for rigid rule structures. By retiring Apex logic and heavy Flow automation, agents were able to take over validations, configuration logic, and document generation. This significantly reduced technical debt and long-term maintenance costs.

7 Prioritise High Value Use Cases for Phase 1

Early success matters. Phase 1 deployments focused on areas with immediate impact:

- Deal configuration
- Dynamic pricing
- Renewals and amendments
- Contract and order document generation
- Deal-desk approvals

These wins earned us trust and accelerated adoption across sales and finance teams.

8 Use Simulation Sandboxes

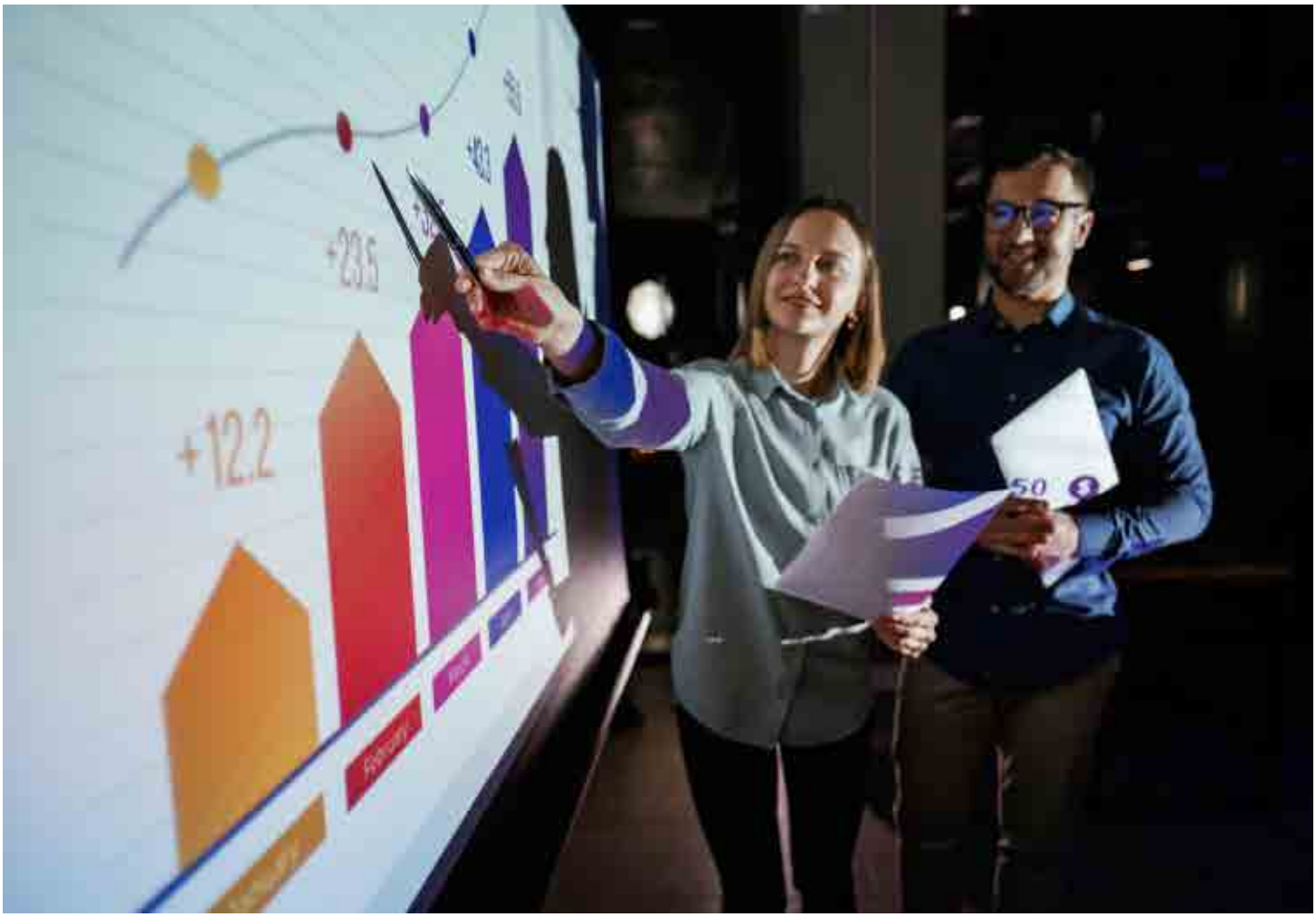
Before production rollout, agent behaviour was trained and tested in controlled environments. Complex deal scenarios, renewals with upgrades, legacy entitlements, and regional pricing variations were used to validate the consistency of reasoning and the quality of the output.

9 Modernise Integrations into a Unified Revenue Backbone

Instead of maintaining custom API layers, ERP, billing, and CLM data were consolidated into a unified intelligence layer and connected through events. This reduced integration brittleness and surfaced data ownership issues early in the transition.

10 Establish Continuous Agent Tuning Cycles

Unlike static CPQ rules, agents improve over time. Regular reviews using feedback from Sales, Finance, and Operations helped refine prompts and policies, ensuring the system stayed aligned with evolving business needs.



Technical Considerations: Data, Integrations, Customisations, and Scalability

Transitioning from CPQ to ARM is not a simple system swap. It demands thoughtful planning, rigorous execution, and strong governance.

Our experience from past implementations indicates that the success of this move often hinges on how well the core technical foundations are handled.

Data Migration:
Data is always the most underestimated aspect of a CPQ to ARM transition. Historical quotes, contracts, pricing references, and order records are often stored inconsistently. Early data profiling and cleansing are essential steps to avoid last minute surprises that impact timelines and confidence in reporting.

A key lesson learned is to define “clean data” standards upfront and to validate migrated data with business users, not just technical teams. Maintaining data integrity is non negotiable, particularly where regulatory compliance, revenue audits, and forecasting accuracy are concerned.

System Integrations:
Revenue systems do not operate in isolation. Most organisations depend on tight integrations with CRM, ERP, billing engines, and downstream analytics platforms. A technological upgrade is an ideal opportunity to reassess and simplify the integration landscape rather than replicate legacy point to point connections.

ARM’s native integration capabilities often allow for standardisation and more resilient interfaces. A common lesson learned is that integration failures typically occur at exception points, not happy paths, so the integration testing must include edge cases and error handling scenarios.

Customisations:
CPQ environments tend to accumulate heavy customisations over time, many of which are workarounds for past limitations. A disciplined review of these customisations is critical. Successful programmes challenge the “we’ve always done it this way” mindset and actively leverage ARM’s out of the box capabilities.

One recurring lesson is that carrying forward unnecessary custom logic increases technical debt, complicates upgrades, and inflates support costs. Wherever possible, configuration should be prioritised over custom code to keep the solution flexible and supportable.

Security and Compliance:
Security and compliance requirements must be embedded throughout the transformation, not treated as a final checkpoint. Data privacy, role based access, audit trails, and regulatory reporting need to be designed into the solution architecture from day one.

A key lesson learned is that misaligned access models can quickly erode trust with legal, finance, and audit stakeholders. Engaging these groups early ensures ARM meets both operational and compliance

Leveraging Advanced ARM Features: AI, Analytics, and Revenue Optimisation

With ARM, organisations can move beyond static pricing and reactive reporting to build smarter, more resilient revenue operations. In my experience, these capabilities translate directly into measurable business impact when applied thoughtfully.

Artificial Intelligence:
AI-driven recommendations have proven to be effective during deal negotiations. In one instance, while supporting a global sales team in closing a deal, AI-based cross-sell recommendations highlighted complementary products that could provide greater value to the customer.

These were not generic suggestions but were crafted based on customers' usage patterns and historical buying behaviour. When incorporated during actual deal negotiations, they helped the team close a larger deal without extending the sales cycle. They built customer confidence in our understanding of their business processes and systems.

Predictive Analytics:
Predictive analytics became a turning point during a renewal cycle for a SaaS portfolio with declining retention. By analysing renewal probability scores and early behavioural signals, we were able to flag at-risk accounts months in advance.

I recall a situation in which a large customer, previously considered stable, was identified as high risk due to reduced license usage. That insight prompted a targeted engagement plan, resulting in a successful renewal that would likely have been missed using traditional reporting alone.

Revenue Optimisation:
Scenario modelling and what-if analysis have been invaluable during pricing and packaging discussions at the leadership level. During a planned price increase, we used these tools to model different discount thresholds and contract durations across regions.

This exercise revealed that a smaller price uplift combined with longer-term commitments actually drove higher lifetime value with less customer resistance. The ability to simulate outcomes before rollout helped align finance, sales, and product teams around a data-backed decision.

Recommendations for a Smooth Transformation

1. **Start with a Clear Business Case**

   Choose the right starting point that allows you to learn safely. Rather than touching the company's largest or most visible product lines, identify a smaller business unit that operated with relative independence. If something broke (and it happens in many, as pilots tend to do), the fallout should be enough to be contained.

2. **Prioritise Simplicity as "Stupid Simple" Wins**

   Follow a rule – If it takes to build a deck for training, it's already complex. In one of the programs, the biggest breakthroughs came when we stopped replicating legacy CPQ logic and began thinking about how to simplify the setup process and its logic.

   Instead of rebuilding numerous CPQ rules, we reimagined them as agent-first logic, focused on guiding decisions rather than enforcing rigid processes, and the result was lower in customisation, minimal complexity, and intuitive design every single time.

3. **Data Quality Should Be the Topmost Priority**

   Data can be a silent killer for any transformation work. Our learnings indicate that no amount of automation or AI can compensate for messy, inconsistent, or incomplete data.

   In our example, the initial migration stalled because pricing and product hierarchies were poorly governed. Rather than pushing forward, we paused the programme and fixed data issues at the source. It felt uncomfortable in the moment, but it saved months of reactive firefighting later.

4. **Use Transformation to Change How You Operate**

   The most forward-thinking organisations don't treat transformation as a one-time project. Transformation works best when it becomes a habit of continuous improvement.

   When the teams involved in transformation start asking "Is this still the best way to sell?" rather than "How do we recreate the old process?", the cultural shift ultimately delivers more value than a mere technological shift.

5. **Partner with People Who've Been There Before**

   Finally, no organisation needs to do this alone. The right partners don't just implement; they help you make smarter choices along the way. Experienced implementation partners bring pattern recognition, knowledge of what typically breaks, what scales well, and what sounds good in theory but fails in practice.

   By combining internal domain expertise with seasoned Salesforce specialists, our client avoided common pitfalls and accelerated decision-making. The partnership became a safety net, and, at times, a compass.

Conclusion

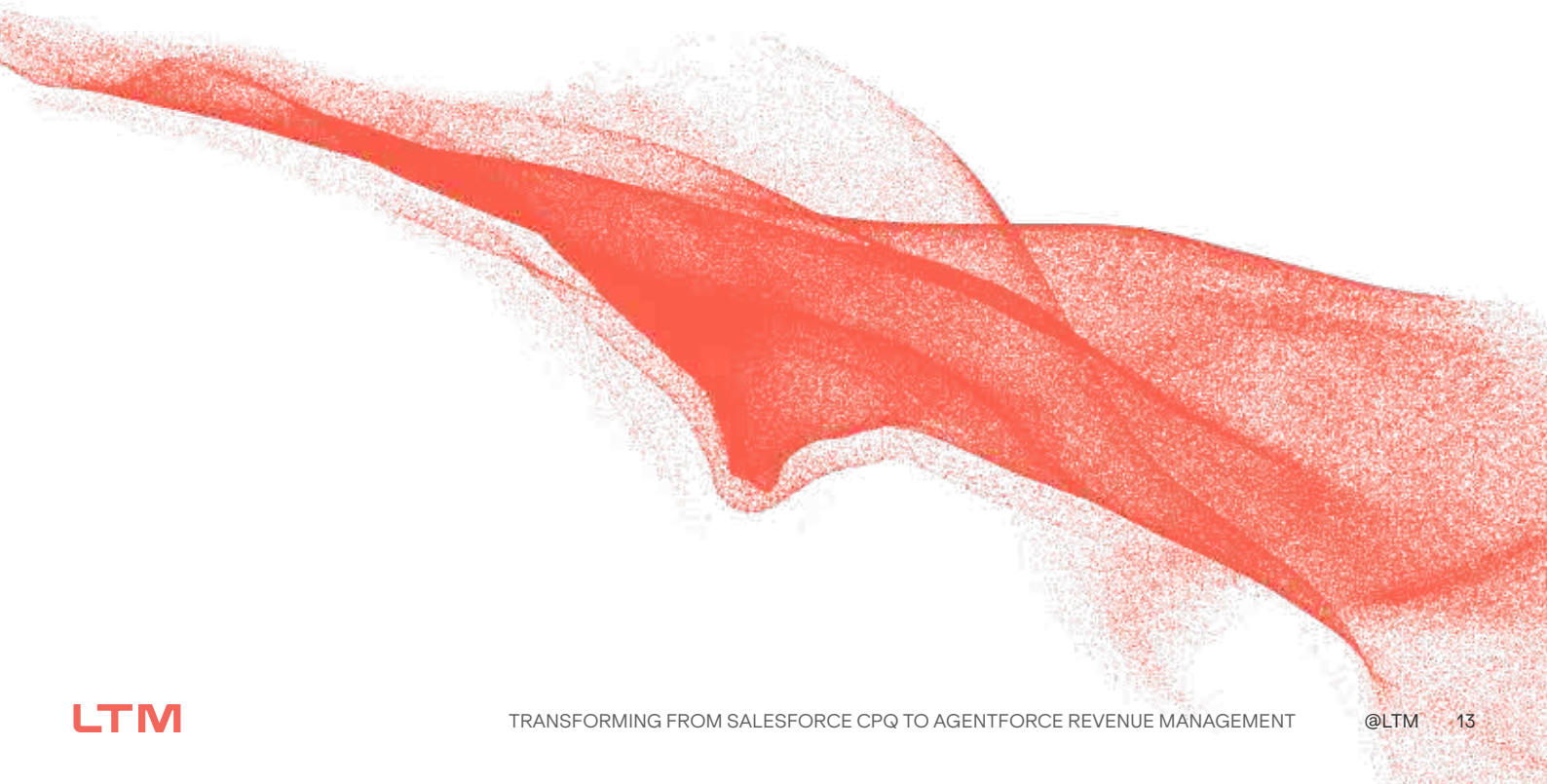
Every technology platform eventually becomes a legacy system. The question is whether the organisation builds something on top of the new platform that would be hard to replicate quickly or recreates what it had before in a newer wrapper. The agent-first approach is a deliberate choice to build something genuinely difficult to copy.

The migration from Salesforce CPQ to Agentforce Revenue Management is a reasonable technology decision for most B2B organisations managing complex, high-volume revenue operations. But the technology decision is the easy part.

The harder decisions, what processes to redesign rather than recreate, how to govern agents with the same rigour applied to human decision-makers, how to develop the team skills that the new operating model requires, and how to sequence the transformation so that value is demonstrated rather than deferred, are what determine whether the investment pays off.

Organisations that approach this transformation with intellectual honesty about their starting point, genuine ambition about their destination, and a phased plan that earns executive confidence at each stage have a realistic path to a revenue operations function that is both more efficient and more strategically capable than what they have today. That combination, doing more with better judgment, not just doing the same with fewer people, is the case for the agent-first approach.

The window for competitive advantage from this shift will not stay open indefinitely. The early movers are already Outcreating operational expertise and accumulating the training data and governance experience that will make their revenue systems progressively smarter. The time to begin the diagnostic work, at a



Meet the author

Apurva Anil Durge

Principal – Architecture Salesforce,LTM

Apurva is a seasoned Salesforce leader and the Revenue Cloud Practice Lead at LTM, with 18+ years of IT experience. His experience spans roles as a Solution Architect, Revenue Cloud Consultant, and Revenue Management Coach. He specialises in modernising Quote-to-Cash for enterprises, enabling scalable, future-ready revenue platforms through deep expertise and strategic insight in an AI-driven world.

References

1. Migrate Your Data to Agentforce Revenue Management, Salesforce Help, https://help.salesforce.com/s/articleView?id=ind.rev_data_migration.htm&language=en_US&type=5.

2. Maximising ROI with Agentic AI: Why Agentforce Is the Fast Path to Enterprise Value, Dion Hinchcliffe and Nick Patience, The Futurum Group, February 2025, https://www.salesforce.com/en-gb/wp-content/uploads/sites/16/documents/agentforce/salesforce-maximizing-roi-with-agentforce-2025-uk.pdf.

3. 3 Important Tips: Migrating from Salesforce CPQ to Revenue Management (RCA), Claudia McPhail, Salesforce Ben, December 3, 2025, https://www.salesforceben.com/3-important-tips-migrating-from-salesforce-cpq-to-revenue-management-rca/.

4. Is an Agentforce Revenue Management Migration the Right Path?, Matt Rooke, Salesforce Ben, March 16, 2026, https://www.salesforceben.com/is-an-agentforce-revenue-management-migration-the-right-path/.

LTM is a global technology services and consulting company and the Business Creativity partner to the world's largest and most disruptive companies. We bring human insights and intelligent systems together to help enterprises across industries rewire their business models, accelerate innovation, and drive AI-centric growth. With our integrated operations, transformation, and business AI services, we design and deliver solutions that create new productivity paradigms and new roads to value. Together with 87,000 employees across 40 countries and our global network of hyperscaler partners, LTM — A Larsen & Toubro company — owns business outcomes for over 700 clients, helping them to not simply outperform the market, but to Outcreate it.