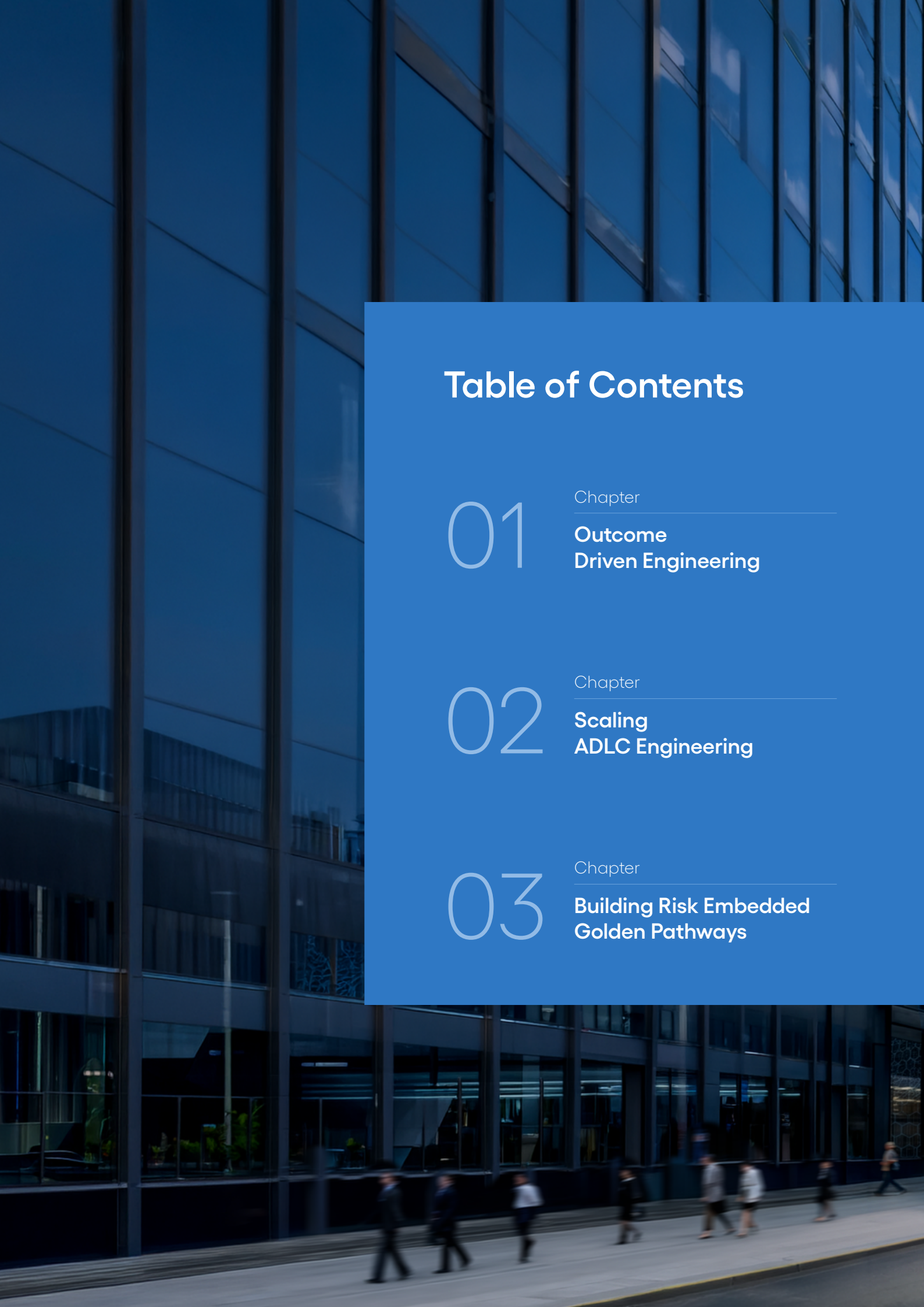
# Engineering at Scale in the Age of AI

Outcome-driven delivery with risk-embedded golden pathways in financial services industry



# Table of Contents

# Three shifts that define modern engineering at scale

## Re-engineering around value streams to unlock flow, resilience and safe AI

Finance organisations are under increasing pressure to deliver value faster, operate with greater resilience and scale AI safely.

While many are investing in more projects, more squads and more tooling, these investments alone are not delivering the step change required.

Leading banks and insurance companies are taking a different approach. They are structuring their engineering organisation around value streams, shifting from project centric delivery to outcome owned teams aligned to end to end customer and business value. At the same time, they are moving from disconnected AI tooling to governed AI capability embedded within those value streams, and from governance after delivery to risk embedded pathways.

This shift is what enables sustained flow of value, reduces friction across the value stream, scales AI with control and maintains regulatory confidence.

> “Many companies are talking about AI; fewer are turning AI potential into real outcomes. Our AI builder approach is designed to close the gap between hype and real business results.

**Ravi Kumar S**
CEO, Cognizant

## Three Shifts

| From | To | Why it matters |
|---|---|---|
| **Project centric delivery** | Value stream aligned, outcome owned teams | Continuous flow of value with clear accountability for outcomes, resilience and cost |
| **AI as disconnected tooling** | AI as governed value stream capability | AI embedded where work happens, reusable, auditable and compounding over time |
| **Governance after delivery** | Risk embedded golden pathways | Reduced friction, faster delivery and stronger regulatory confidence |

These shifts are already proving critical for banks, insurance and superannuation organisaitons in Australia, North America, and Europe. This paper examines why these shifts are inseparable, and how treating engineering as a system rather than a set of projects is now critical for value, resilience and AI at scale.

01 Chapter

# Value stream aligned, outcome owned teams

# Outcome driven engineering: Why project delivery limits value in banking and finance

## From project delivery to sustained value creation.

Finance companies face sustained pressure to reduce cost to serve while accelerating digital change to build and retain customer loyalty. While investment has improved delivery capability, many organisations struggle to translate this effort into consistent, enterprise level improvements in customer experience, resilience, and operational efficiency.

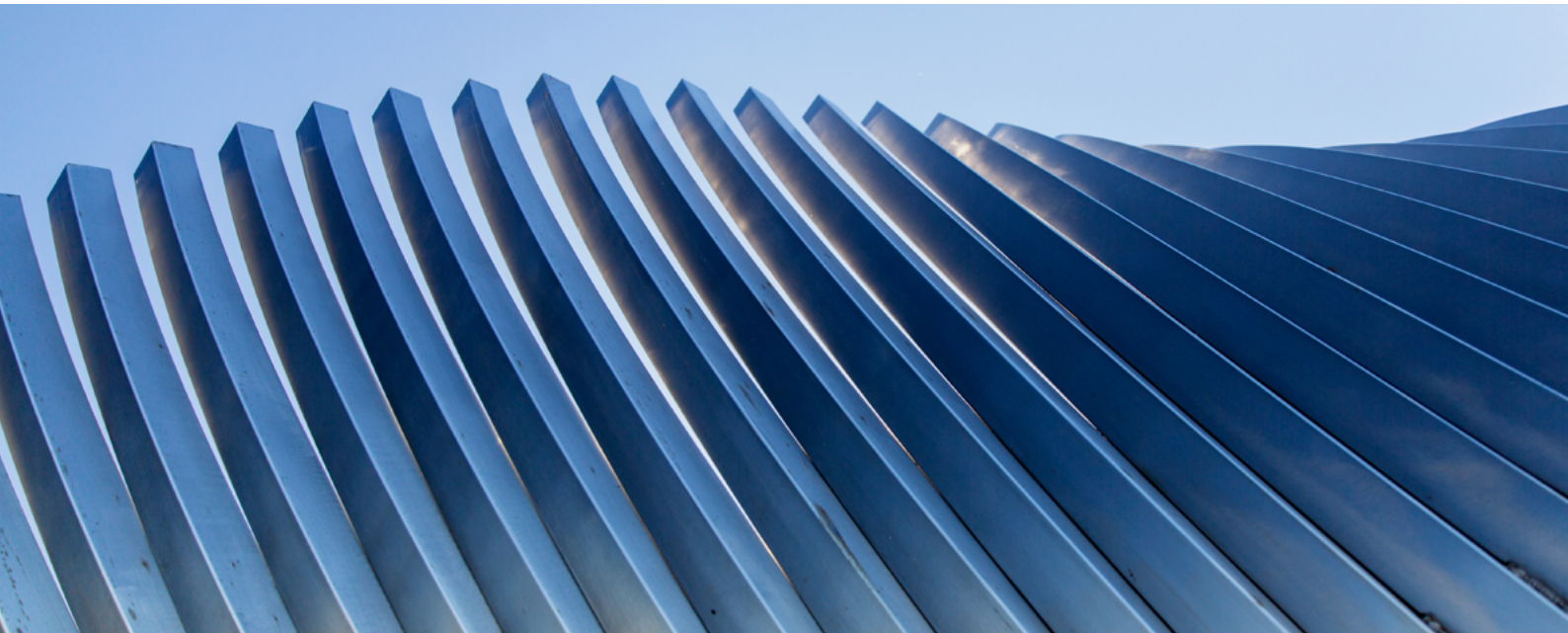
The issue is not speed of delivery, but how engineering work is funded, governed, and sustained. High performing banks and insurers are moving beyond project centric models, aligning persistent teams to products, platforms, customer journeys, and prioritising sustained outcomes over short term milestones.

### The problem with project delivery in modern finance industry

Most banks still organise technology investment around projects. Funding is approved for discrete initiatives, success is measured by scope, schedule, and cost, and teams are formed to deliver change before disbanding. This approach remains effective for large, time bound or regulatory initiatives.

However, in always on digital banking environments, a project centric model creates structural limitations. When ownership resets at the end of each initiative, accountability for optimisation, resilience, and long-term cost efficiency weakens. Over time, this leads to fragmented ownership, growing technical debt, and limited improvement in customer experience or operational performance.

Project delivery also shapes behaviour. Teams optimise within project boundaries, prioritising delivery milestones over sustained outcomes. Reliability, resilience, and cost efficiency struggle to compete for attention once immediate delivery goals are met, despite being critical to enterprise performance.



## What outcome driven engineering looks like in practice

High performing banks and insurers are responding by adopting outcome driven engineering. Technology investment is aligned to the sustained delivery of measurable customer and business outcomes, with engineering organised around products, platforms, and customer journeys rather than temporary initiatives.

This shift has three practical implications.

01

### Persistent, cross functional teams

Instead of forming teams around projects, companies establish long-lived, cross-functional teams aligned to specific products or customer journeys. These teams retain end-to-end ownership across delivery, run, optimisation, and evolution. This ownership preserves domain knowledge, supports continuous improvement, and enables teams to be accountable not just for delivery, but for the resilience, cost, and performance of the platforms they operate.

02

### Business cases evolve into outcome commitments

Traditional business cases seek to predict scope, cost, and benefits upfront, often limiting learning and adaptation as change unfolds. Outcome driven models replace this approach with explicit outcome commitments, where success is measured through tangible improvements in customer experience, cost efficiency, resilience, or risk reduction, and benefits are assessed progressively as evidence emerges. This allows investment to be increased, redirected, or stopped based on real performance rather than upfront assumptions.

03

### Governance focuses on value, not milestones

Governance evolves from project stage gates to outcome-focused review, where executives and senior leaders assess progress across outcomes, customer metrics, and engineering health, not delivery milestones in isolation. Within agreed boundaries, teams are empowered to make prioritisation decisions based on evidence and trade-offs, enabling faster decision-making and lighter governance while preserving clear executive oversight of risk and investment.

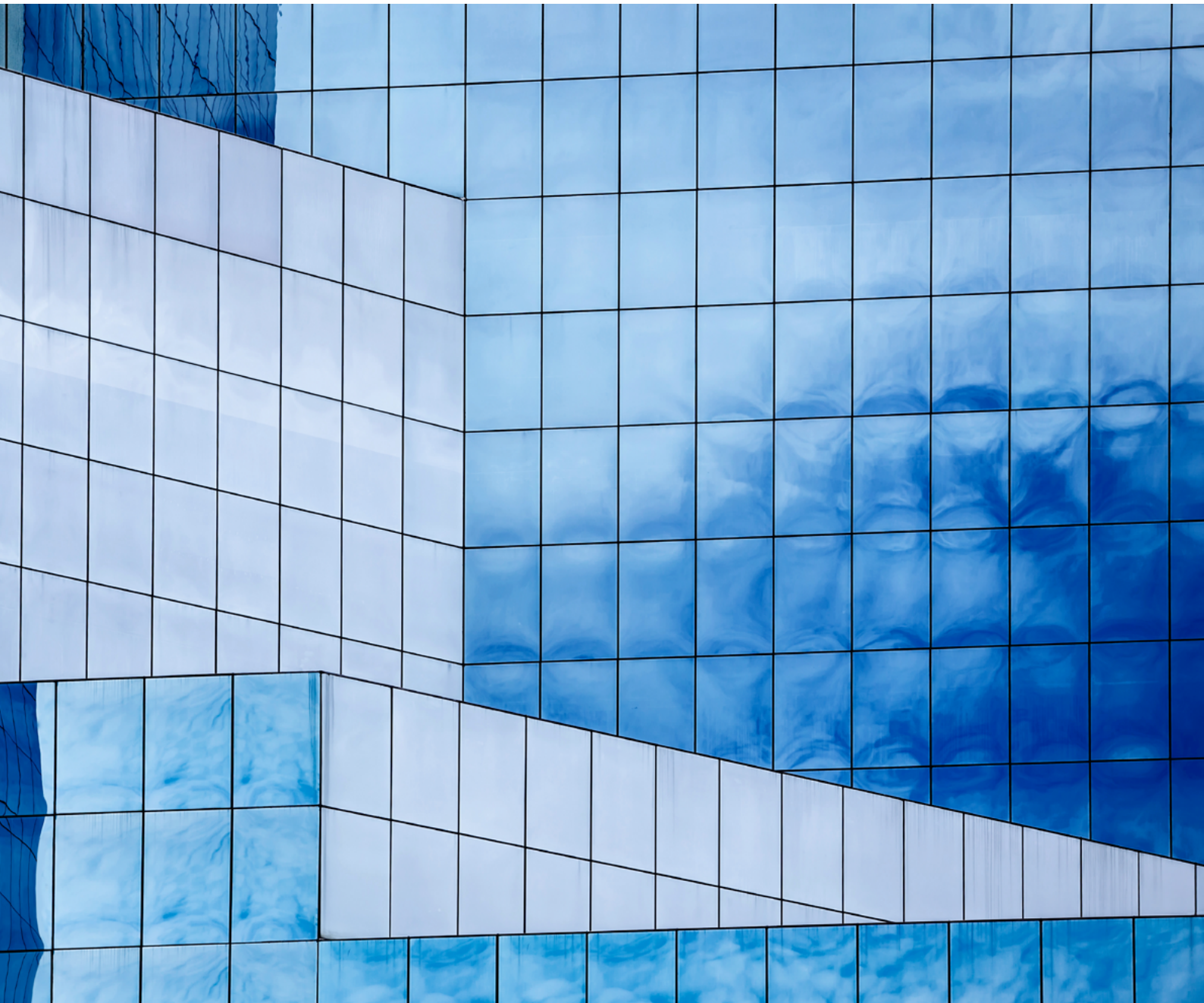
> **Sustained ownership of products and customer journeys requires funding and governance models that support teams beyond individual initiatives.**

# Outcome driven engineering in practice: The Lending squad model

## A worked example: Lending product squad, AI agents, and outcome-driven prioritisation

Let's use lending as a worked example to show how a persistent team is structured, how autonomous agents reduce cost of delivery as standing team members, and how product owners translate outcome commitments into day-to-day prioritisation decisions.

This model applies equally to other banking product lines including Deposits, Payments, Home Loans, and Wealth. The principles are consistent: persistent teams, agents as standing capacity, and product owners accountable for measurable business outcomes.



## Example: How a Lending product team is structured

A persistent Lending squad brings together a business product owner, engineering lead, platform engineers, data and risk specialists, and a UX designer. The team owns outcomes end to end across origination, decisioning, and servicing, and operates continuously rather than reforming around projects.

This structure builds durable domain expertise, reduces handover risk, and shifts accountability from feature delivery to measurable business outcomes.

Platform capability sits alongside the squad. A shared lending platform team provides core services including origination, credit policy, document management, and regulatory reporting. This separation allows the squad to stay focused on customer and business outcomes, while platform engineers evolve reusable capabilities that support multiple journeys.

## AI agents as persistent team members: Reducing cost of delivery whilst shortening delivery times

In the lending model, autonomous agents operate as standing members of the team alongside human roles, providing always on execution for high volume, rules driven work.

- Document processing agents handle verification and data extraction across loan applications
- Credit decisioning agents perform policy validation and real time scoring
- Compliance monitoring agents track regulatory obligations and surface exceptions early

These agents are not point automations. They are versioned, governed, and continuously improved as part of the product backlog.

This enables a nonlinear delivery model. As agent capacity absorbs routine work, human effort shifts to judgement intensive activities, exception handling, and continuous improvement. Delivery cost reduces while accountability and control are strengthened.

## Product owners driving business outcomes and prioritisation

The business product owner is the single point of accountability linking the team to the outcomes the bank has committed to.

In the Lending squad, this accountability is anchored in a small set of business critical metrics:

- Time to conditional approval
- Straight through processing rate
- Cost per funded loan
- First time approval rate

These are funding metrics, not reporting metrics. They determine where investment continues, expands, or is redirected.

Prioritisation operates at two levels:

**Team level:**
Backlog is continuously ranked based on outcome impact, balancing feature delivery, agent capability uplift, technical debt, and regulatory obligations

**Portfolio level:**
Cross journey dependencies and trade offs are surfaced to governance forums, where funding and sequencing decisions are made based on outcome evidence

This shifts governance from milestone tracking to evidence based decision making.

# Outcome driven engineering aligns technology investment to measurable improvements

## Measuring what matters to drive sustained engineering performance

Effective outcome driven engineering depends on practical, deliberate measurement. Banks that succeed do not attempt to measure everything. Instead, they use a small, explicit set of measures that together provide a balanced view of value, experience, and sustainability.

Crucially, these measures are reviewed together. Speed without stability, or delivery without resilience, is treated as a governance signal rather than a success. Measurement becomes a mechanism for informed decision-making, not a reporting exercise.

### Examples of outcome-based measurements

| Measurement area | KPI examples | What it measures (practical definition) | Typical use in governance | Framework Alignment ** |
|---|---|---|---|---|
| 1. **Business and customer outcomes** | • Cost to serve per journey<br>• Straight through processing rate Operational losses trend<br>• Regulatory response time<br>• Digital adoption uplift | Whether engineering effort is improving bank economics, customer value, and risk outcomes. Examples include reduced cost to serve, improved straight through processing, reduced operational losses, faster regulatory response, and increased digital adoption. | ExCo and portfolio reviews to decide whether to increase, reduce, or redirect capacity based on outcome movement rather than delivery volume. | Measures customer satisfaction and happiness with outcomes (HEART framework)<br>Tracks business impact and perceived value (SPACE framework) |
| 2. **Product and journey performance** | • Journey completion rate<br>• Adoption of delivered capabilities<br>• Rework or manual intervention rate Customer effort indicators<br>• Time to complete journey | Whether delivered changes are being used and are improving customer experience. Adoption, completion, rework, and customer effort are more informative than feature counts. | Product governance to stop low value change, prioritise friction removal, and focus roadmaps on measurable experience improvements. | Tracks engagement, adoption, and task success (HEART framework)<br>Tracks customer value and efficiency of delivered capabilities (SPACE framework) |
| 3. **Engineering health and sustainability** | • Platform reliability and availability<br>• Incident frequency<br>• Deployment quality (change failure rate)<br>• Security posture<br>• Automation coverage<br>• Technical debt trend | Whether engineering can deliver at pace without increasing operational and security risk. Measures include reliability, incident frequency, deployment quality, security posture, automation coverage, and technical debt trends reviewed alongside delivery outcomes. | Operational and risk governance to trigger investment in resilience, security, automation, and debt reduction before issues become outages or losses. | Tracks deployment speed and delivery stability (DORA metrics)<br>Tracks engineering efficiency and process health (SPACE framework) |

****DORA** - Four metrics benchmarking software delivery performance: Deployment Frequency, Lead Time for Changes, Change Failure Rate, and Time to Restore Service.

**HEART** - Google's UX framework measuring product quality across Happiness, Engagement, Adoption, Retention, and Task Success.

**SPACE** - Developer productivity framework spanning Satisfaction, Performance, Activity, Communication & Collaboration, and Efficiency & Flow.

Outcome based measurement drives better decisions and sustained performance.

A small, balanced set of metrics across business outcomes, customer experience, and engineering health ensures near term value while strengthening long term resilience.

Measurement shifts from retrospective reporting to forward looking decision support, enabling faster, more confident trade offs on funding, priorities, and execution.

As the operating model evolves to product and journey aligned teams with clear decision rights, measurement becomes an active governance tool, accelerating high impact areas and redirecting low value work.

Over time, this creates compounding improvements in customer outcomes, reliability, cost efficiency, and engineering productivity, reducing reliance on large transformation programs and establishing a more predictable, sustainable delivery model.

# Cognizant's offering: Outcome driven engineering at enterprise scale

## Turning outcome driven operating models into sustained business results

## Turning outcome driven operating models into sustained business results

Cognizant partners with banks and finance organisations to design, implement, and scale outcome driven engineering operating models that deliver measurable business value over time. We combine operating model design, deep engineering capability, and AI augmented delivery using Cognizant Flowsource™ for feature development to enable faster, safer, and more repeatable delivery of customer and platform features. This allows finance organisations to move beyond episodic change and establish sustained ownership of their most critical platforms and customer journeys.

Our approach recognises that delivery capability alone is not sufficient. Lasting impact requires aligned funding models, outcome focused governance, practical measurement, and engineering practices that can operate confidently at enterprise scale within regulated environments.

## Why Cognizant

Designing outcome driven operating models is not primarily a technology problem. It requires practitioners who have navigated the funding restructures, governance changes, and leadership conversations that make persistent team models stick inside large, regulated institutions. Cognizant has designed and embedded these models across major banks in Australia, the UK, and North America, working from operating model blueprint through to live squad implementation.

Cognizant Flowsource™ for feature development provides a structured, industrialised pathway for feature delivery, breaking down complex banking change into governed, traceable, and reusable delivery flows. This enables persistent teams to operate with speed while maintaining auditability and regulatory alignment. Anthropic and Google Cloud partnerships ensure access to frontier AI within an enterprise governance structure from the outset, not as a retrofit.

| Offering component | What Cognizant provides | What it enables for finance organisations | Examples |
|---|---|---|---|
| **Outcome driven operating model design** | Design of product and value stream aligned operating models, replacing project centric delivery with persistent, cross functional teams accountable for end-to-end outcomes | Clear ownership of customer journeys and platforms, reduced fragmentation, and sustained value creation rather than episodic change | Cognizant consulting engagements to establish product aligned squads for core banking platforms, digital channels, or lending journeys |
| **Outcome based governance and measurement** | Replacement of milestone reporting with a small, explicit set of business outcomes, product and journey performance metrics, and engineering health indicators | Executive visibility into value, risk, and sustainability, enabling better funding, prioritisation, and intervention decisions | Design of outcome dashboards covering customer experience, resilience, cost to serve, and flow efficiency to provide transparency |
| **Outcome based feature development** | Design and industrialisation of feature delivery using Cognizant Flowsource. Decomposes complex banking initiatives into structured, reusable feature flows with embedded governance, testing, and release controls, enabling consistent execution across squads | Predictable, scalable feature delivery with reduced variability. Faster time to market while maintaining strong auditability and regulatory compliance. Improved developer experience through standardised pathways and reusable engineering patterns | Deployment of lending, payments, and digital channel features using Flowsource aligned delivery flows, integrated into core banking platforms with embedded controls and traceability |

02 Chapter

# AI as a governed value stream capability

# From AI as a tool to AI as a capability: The architecture imperative

## Why scaling squads is no longer the answer

Chapter 1 established why the project-centric delivery model limits value in banking, insurers and superannuation companies and what a better operating model looks like: persistent teams, outcome-focused funding, and agents as standing delivery capacity. Chapter 2 addresses the next question. Once you have the right operating model, how do you scale execution without coordination consuming the value it creates? The traditional DevSecOps model has reached its structural limit. In large, regulated environments the constraints are now systemic, not cultural. More agile ceremonies cannot resolve a flawed coordination architecture. The same practices that unlocked value at one scale are now the ceiling at another.

### 1. The three structural constraints

**Coordination overload:**

As systems grow, dependencies outpace team capacity. Adding more squads' compounds cost without increasing output.

**Latency by design:**

Sequential reviews and manual approval workflows introduce bottlenecks at every stage of the lifecycle.

**Fragmented accountability:**

End-to-end outcomes are lost across team boundaries, creating ownership gaps that no ceremony can bridge.

Each compounds with every additional team added to the programme.

### 2. The hidden risk: Structural fragility

Expertise in banking and financial services is not held in any one system. Critical domain logic lives in the heads of a few senior individuals, and when those people leave, the reasoning behind the code disappears with them. It cannot be searched, versioned, or governed. Transferring this tacit knowledge takes months and is rarely complete, creating a permanent state of operational risk.

AI adoption accelerates this fragility rather than reducing it. Tools that amplify individual judgment also amplify the gaps when that judgment is absent or inconsistent. An unreviewed agent workflow, an ad hoc context window, a system prompt written by someone who has left. Each is invisible technical debt that surfaces at the worst possible moment.

For regulated institutions this is not just a delivery risk. It is a governance risk. When the reasoning behind a critical data pipeline or credit decisioning rule exists only in a senior engineer's head, the company cannot demonstrate the control framework that regulators require. Knowledge that cannot be audited cannot be governed. And knowledge that cannot be governed creates regulatory exposure that no amount of retrospective documentation resolves.

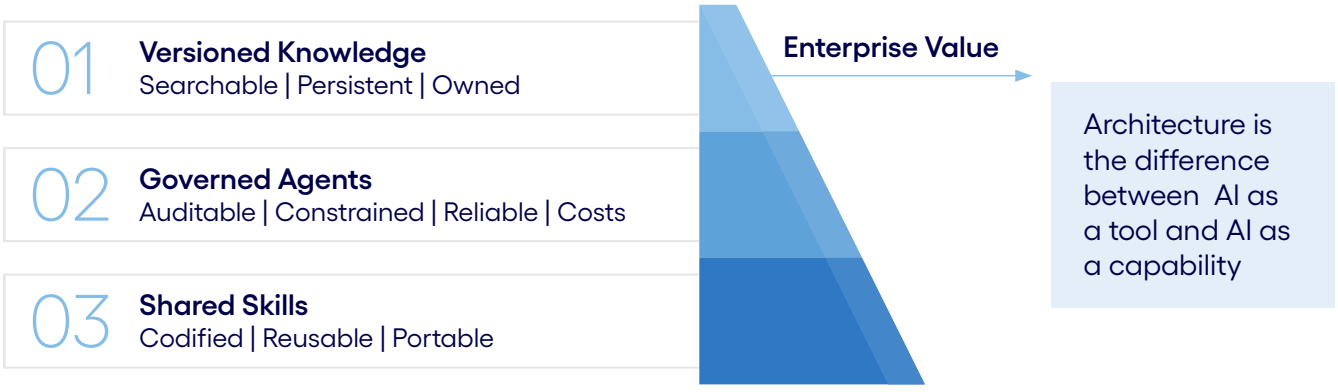
## 3. The AI paradox: Tools versus architecture

AI is being adopted as a tool, not built as a capability. Unreviewed agent workflows and ad hoc prompts create invisible technical debt that surfaces at the worst possible moments. Without a governed framework, agents re-derive solutions instead of reusing versioned, validated knowledge, creating redundant effort across teams. And individual productivity gains do not aggregate into enterprise improvement when the underlying operating model remains unchanged.

Without deliberate architecture for how knowledge is codified, how agents are governed, and how skills are shared, AI adoption scales inconsistency faster than advantage. The productivity gains visible at the individual level do not aggregate into enterprise-level improvement when the operating model remains unchanged.

> **The response is not more squads and not more agile ceremonies. It is a fundamental change in how engineering work is designed, governed, and scaled. Architecture is the difference between AI as a tool and AI as a capability, controlling not just risk and quality, but also cost, including how token consumption scales across the enterprise.**

### Deliberate Architecture




# Cognizant's perspective on the journey

### Three stages from manual, large-team delivery to governed, agent-driven engineering

The transition to agentic engineering follows a clear progression. Most organisations today are at Stage 1 or early Stage 2. The move forward is not primarily a technology decision. It requires deliberate changes to team structure, funding model, skills, and governance. Each stage represents a different answer to a single question: where does engineering value come from?

Knowing which stage your organisation occupies today is the starting point for designing a credible path forward. The decisions made now, on specifications, tooling, governance, and skills, determine whether Stage 3 is reachable within a short term horizon or remains aspirational.

## Cognizant's perspective on the journey to ADLC maturity


## Stage 01

## Traditional SDLC. Manual processes, large teams, sequential handoffs

**High headcount pyramid:** Architects and Leads at the top, Senior Devs, BAs, and Modellers in the middle, Junior Devs, Testers, and Ops at the base. All artefacts are created and validated by humans. Sequential handoffs between Dev, Test, Security, and Ops introduce latency at every stage. Knowledge is siloed by role and leaves when people do.

**Skillset profile:** manual execution. Ingestion and batch, source to target mapping, manual testing, siloed knowledge. Productivity is bounded by headcount. Scaling delivery means adding people, which compounds coordination overhead and cost.

### Stage 02

### ADLC 1.0. Human in the loop, multi-agent, coordinated delivery.

**Medium headcount:** Architects and Product Owners at the top, Senior Devs, BAs, Modellers, QA, and Ops in the middle, agents at the base handling stories, code generation, and test generation. Agents absorb junior execution while humans supervise and validate. Sequential quality gates enforced in CI/CD.

**Skillset profile:** hybrid and prompting. Prompt and context engineering, code validation, test automation, agent coordination. Repeatable delivery effort reduces by 30 to 50 percent. Human coordination is still required at every lifecycle step.

### Stage 03

### ADLC 2.0. Humans govern outcomes, agents execute the lifecycle.

**Low headcount, high value:** Technical Product Managers and AI Architects at the top, AI harness in the middle, agents executing specification, engineering, testing, and sub-agent orchestration at the base. No junior execution roles. Agents run the full delivery lifecycle. Compliance is structurally guaranteed: violations cannot complete the pipeline.

**Skillset profile:** declarative. Product management, spec authoring, agent reliability engineering, strategic problem solving. Expertise is versioned, governed, and reusable. It survives attrition and compounds with scale. Every improvement to the specification library permanently raises the engineering floor for all teams.

> **The decisions made now determine which stage is reachable by 2027. Starting at the right point on this progression is the single most imporant engineering investment a organisation can make this year.**

# Productivity, cost, and compounding returns

**Agentic engineering is not a speed upgrade. It is a shift to an AI Driven Lifecycle where design, execution, and governance are continuously integrated rather than staged.**

As execution becomes more automated and repeatable, the marginal cost of change declines permanently. Enhancements, defect fixes, and regulatory updates can be introduced with less incremental effort while maintaining stronger controls and auditability. Issues that traditionally surface late in delivery are identified and addressed earlier, resulting in fewer production incidents and significantly less rework across the programme.

## The system gets stronger with scale, not more fragile.

ADLC changes how capability accumulates. Specifications, agent workflows, and decision logic persist and improve with each iteration rather than resetting at project boundaries.

Each improvement raises the baseline for all teams, creating a system that becomes more predictable, resilient, and cost effective with scale. Gains do not plateau because they are embedded directly into how work is designed and executed.

## Productivity outcomes are measurable and structural across four dimensions.

ADLC shifts productivity from individual effort to system level efficiency.

- Repeatable tasks are automated through governed agents, reducing manual effort by 30 to 50 percent.
- Quality is stabilised through embedded validation and continuous testing, with automation coverage exceeding 80 percent in critical paths.
- The effective cost per unit of delivery declines materially as reuse increases across specifications and workflows.

Innovation becomes a structural allocation, not a residual. A defined portion of delivery capacity is continuously reinvested into improving the system itself.

## Measurement and cost discipline within ADLC

ADLC makes cost visible and controllable at the point of execution.

Token consumption, agent execution, and workflow complexity become measurable units of delivery cost. Without ADLC, these scale non linearly through duplicated prompts, ungoverned agents, and fragmented knowledge.

With ADLC:

- Token usage is optimised through reuse of specifications and workflows.
- Agent execution is governed, auditable, and aligned to outcomes.
- Cost scales predictably with value delivered, not with activity or duplication.

Cost discipline is therefore achieved through architecture, not consumption limits.

## Measurement enables continuous governance.

In ADLC, measurement is embedded directly into the lifecycle. Outcome metrics, engineering signals, and cost indicators are continuously captured and used to guide decisions.

This enables faster trade offs, earlier intervention, and more confident investment decisions. Teams demonstrating strong outcome movement are accelerated, while low impact activity is simplified or stopped.

## Compounding advantage through ADLC

Organisations that adopt ADLC early build structural advantage that compounds over time. As specifications and agent systems improve, gains in productivity, reliability, and cost efficiency accelerate.

This creates a widening gap between AI as a tool and AI as a capability, establishing a more predictable and sustainable delivery model.

# Case study of ADLC in action: transforming data engineering delivery

**How a major Australian bank modernised how it builds and governs data pipelines.**

Cognizant is actively applying the ADLC model inside a major Australian financial institution, transforming how data engineering pipelines are specified, built, tested, and governed. The engagement demonstrates in practice what the maturity progression described in this chapter looks like when applied to a real data engineering environment operating under APRA regulatory obligations. The work is live, the metrics are real, and the transformation is progressing toward supervised autonomy.

## Example: The data engineering lifecycle before the change: manual, slow, knowledge-dependent.

The starting point was a traditional data SDLC: large delivery teams of Service Owner, Scrum Master, Technical Lead, Data Modeller, Tech BA, four Data Engineers, and a Batch Engineer. Each pipeline required sequential handoffs through development, testing, and governance. Source-to-target mapping was manual. Business requirements were translated into JIRA tickets by hand. Test data was created and managed manually. Governance was a phase-gate process applied after delivery. Knowledge was siloed in senior engineers. Attrition was an existential delivery risk.

The data engineering programme covered five delivery stages: Product Discovery covering story definition, solution design, data modelling, and gold record specification; Engineering covering agent sandbox, data sourcing, source-to-target transforms, and unit test validation; Scenario Validation covering synthetic test data management, SIT, UAT, and SVP; Governance covering audit trail, governance approvals, and compliance; and Production and Ops covering deployment, monitoring, and L2/L3 remediation automation.

## What changed: From manual execution to governed agentic delivery.

Cognizant introduced a skill-driven AI harness across all five delivery stages. A Gold Record Specification agent defines the target data schema as machine-readable JSON/YAML before engineering begins. A Story Creator agent converts business requirements directly into structured user stories and source-to-target mappings. A Lineage Blueprint agent auto-captures decision lineage from specification through to pipeline output, making audit trails automatic rather than manual. A Reconciliation Harness agent validates gold records against baseline specs, replacing human sign-off at the quality gate.

A Synthetic Test Data agent generates SIT and UAT test data scoped to the gold schema with no manual fixture creation. A Self-Healing agent detects drift in gold records and self-corrects within specification boundaries. Compliance checks are automated throughout CI/CD: non-compliant outputs cannot complete the pipeline regardless of delivery pressure. The five-dimension evaluation harness measures trigger accuracy, isolation, coexistence, instruction adherence, and output quality on every pipeline run. All outputs are APRA-ready by default through the harness, not by human review after the fact.

### The outcome: Measurable improvement across delivery, team, and skills dimensions.

Delivery outcomes: engineering effort on repeatable source-to-target and testing tasks reduced by 30 to 50 percent. Regression test automation coverage reached 80 percent or more. The first fully autonomous data pipeline was delivered end-to-end by agents with human governance only at the specification and approval gates, not at each execution step. A Skill Registry and MCP gateway went live, enabling skills to be reused across squads rather than rebuilt for each product.

### Team evolution: From large manual squads to small, governed roles.

The delivery team moved from the traditional structure of eight to ten human roles per squad toward a Stage 2 configuration of Architect, Senior Dev, BA, QA Automation Engineer, Ops Reliability Engineer, plus agents. The target Stage 3 configuration is Service Owner, Domain Evaluator, Agent Orchestrator, Policy and Safety Guard, and AIOps Analyst only, with agents executing the full delivery lifecycle. The transformation embedded new roles progressively: Logic Architect and Harness Engineer roles were defined and filled first, followed by an Agent Orchestrator embedded in the pilot squad, followed by a Policy and Safety Guard aligned with Risk and Compliance. BAU delivery continued throughout.

> **The transformation demonstrates that ADLC is not theoretical. It is running in production inside a regulated Australian financial institution today, delivering measurable improvements in quality, compliance, and delivery economics on live data engineering pipelines.**

# Cognizant's offering: ADLC engineering at enterprise scale

## Turning ADLC operating models into measurable delivery improvement and structural engineering advantage.

Chapter 1 covered how Cognizant helps banks design the right operating model: persistent teams, outcome-based funding, agentic delivery, and governance. Chapter 2 examined the delivery architecture that makes that model scale. The four capabilities below build directly on the operating model foundations from Chapter 1, adding the specification frameworks, deterministic tooling, talent model, and shared accountability structures that turn operating model design into sustained delivery improvement.

## Why Cognizant

The argument in this chapter is only credible if the tools and track record exist to back it. Cognizant built BMad and SpecKit as proprietary specification frameworks, purpose-designed for deterministic delivery in regulated environments. Agent Foundry and MAKER are operating in production on live delivery programmes, not in proof of concept. The ADLC maturity progression has been applied at major Australian, UK, and North American banks, with stage-by-stage outcomes that are measured rather than asserted. For banks that need a credible path from Stage 1 to Stage 3 within a two-year horizon, Cognizant provides the delivery architecture, tooling, and embedded capability building to make that transition real.

| Offering component | What Cognizant provides | What it enables for finance organisations | Examples |
|---|---|---|---|
| **Deterministic, spec-driven ADLC delivery** | Design, implement, and operate deterministic agentic delivery systems using BMad, SpecKit, and sequential quality gates. Specifications define intent before implementation begins. Agents execute within constrained, auditable workflows. Compliance is structurally guaranteed. | Engineering output that is reliable, auditable, and regulatorily compliant at scale. Compliance exceptions cannot complete the pipeline regardless of time pressure or human oversight gaps. | Deployment of spec-driven agentic scrum squads across lending, payments, and platform modernisation programmes, with 30 to 50 percent effort reduction on repeatable delivery tasks. |
| **ADLC maturity progression and operating model design** | Design of the operating model transformation from Stage 1 traditional SDLC through Stage 2 ADLC 1.0 to Stage 3 ADLC 2.0, covering team structure, funding model, governance, and skills architecture. | A clear transformation roadmap with stage-by-stage milestones, measurable outcomes at each stage, and a practical path to Stage 3 engineering leverage without scaling headcount. | ADLC maturity assessments and transformation programmes for Australian FSI clients, embedding change within delivery cadences rather than alongside them. |
| **T-shaped capability and agentic talent model** | Build and embed the T-shaped engineers, Logic Architects, and Agent Reliability Engineers that Stage 3 requires, alongside specification frameworks, skill libraries, and governance practices. Skills are codified, versioned, and governed rather than locked in individuals. | Engineering capability that survives attrition and compounds with scale. Expertise persists in the system rather than in people, permanently raising the engineering floor for every team that uses it. | Embedded capability building programmes that transfer skills into the client engineering system, with versioned specification libraries and agent reliability practices that remain after Cognizant engagement concludes. |
| **Shared accountability for outcomes, not just delivery** | Live transformation backlog, shared OKRs spanning AI adoption rate, delivery flow efficiency, engineering quality, and toil reduction. Named leaders on both sides review progress weekly against agreed outcome measures with full transparency. | Both organisations remain continuously aligned on what is working, what requires adjustment, and where to invest next. Investment is redirected based on evidence rather than deferred to programme reviews. | Weekly outcome reviews with shared dashboards covering deployment frequency, change failure rate, agent reliability metrics, and cost per unit of delivery across the programme. |

# 03 Chapter

# Risk embedded Golden Pathways

# The Golden Pathway

## From governance after the fact to governance by construction

The response to the Velocity Risk Paradox is not more policy, more review boards, or more tools. It is a change in architecture.

The Golden Pathway is Cognizant's model for embedding governance directly into how AI agents design, build, validate and release product. It ensures that compliance is not something teams do at the end, but something the platform enforces continuously as work is executed.

This is what makes compliance self-governing and architecturally guaranteed, rather than statistically likely.

### What the Golden Pathway is (and is not)

The Golden Pathway is not a single tool and not another layer added on top of existing delivery.

It is a layered control surface that spans the full lifecycle. From intent capture and design, through agent execution, to release, monitoring and audit.

It establishes a single, governed way for work to move from idea to production, whether that work is performed by humans, agents, or both.

### Humans govern outcomes, not tasks

The model deliberately separates accountability from execution. Humans set intent, define constraints and approve control gates. Agents execute the work within those boundaries. Responsibility remains explicit and human, while latency and manual intervention are removed from the delivery flow.

### The delivery lifecycle is the governance spine

Governance is embedded directly into the delivery lifecycle. Discovery, engineering, validation, governance and production run through a single continuous pipeline that is agent-executed and policy-controlled. There are no hand-offs between delivery and governance, and no late-stage reviews where risk is identified too late to matter.

### The AI Harness governs execution continuously

Execution is enforced by the AI Harness, which applies gold specifications, validates transformations, reconciles outputs, certifies releases and monitors production in real time. Beneath this layer, specialised agents perform discrete delivery tasks across the lifecycle. New agents inherit the same controls by default, ensuring governance is never weakened as capability scales.

### Context is the trust boundary

Agents do not operate on prompts alone. They execute using governed skills, live enterprise integrations and managed regulatory knowledge. This ensures every action is grounded in institutional context rather than inferred intent, making execution predictable, auditable and trustworthy.

### Compliance is structural, not statistical

The compliance anchor sits beneath the entire pathway, continuously linking execution to approved requirements, architectures and regulatory obligations. Evidence is produced automatically as work progresses. This is what allows the platform to deliver structural compliance by design, rather than retrospective or probabilistic assurance.

# The technology foundation of the AI Golden Pathway

## Governance execution at scale

The Golden Pathway is enabled by a deliberately composed technology stack. Not a monolithic platform, but a set of tightly integrated capabilities that turn governance intent into executable control.

Each layer exists to solve a specific failure in traditional delivery models: fragmented tooling, inconsistent policy enforcement, and manual evidence creation.

### End-to-end AI Builder operating model

**Personas**
(Human Layer; Roles and Responsibilities)

**Discovery & Strategy**
- Product Manager (p/t)
- Logic Architect (p/t)

**Build & Validate**
- Agent Orchestrator
- Policy and Safety Guide

**Release & Ops**
- Release Governance Lead (p/t)
- AI Ops Analyst

**Data engineering stages**
(Workflow)

| Product discovery | Engineering | Scenario validation | Governance | Production and ops |
|---|---|---|---|---|
| Story Definition, Solution Design, Data Modelling, Gold Records | Agent Sandbox, Data Sourcing, Transforms, Unit Test Validation | Synthetic Test Data Management, SIT, UAT, and SVP | Audit Trail, Governance Approvals and Compliance | Deployment to Prod, Monitoring, L2/ L3 Remediation Automation |

**AI harness**
(Process, People and Tools)

**Spec Defined** | **Data Transformations** | **Reconciliation Harness** | **Gold Certification** | **Production Monitoring**

CLI Based Command Line to Create New >

**Sub agents**

Story Creator | Traceability Blueprint | Sandpit Environment | Synthetic Test Data | Test Planning | SIT Testing | Audit Trail Validation | Specification Updates | AI Ops& Costs | Deploy to Production | Source Profiling | Solution Design Artefacts | Source Data | Unit Test Scripts | UAT Testing | SVP Testing | Compliance Checks | Solution Readiness Checks | Self Healing | Issue Resolution

**AI asset marketplace**
(Discover and Acquire Assets)

**Search & browse**

**Agent Skills**
Product & Standard Skills

**MCP Servers**
Context Integrations

**Custom Agents**
Prebuilt Specialised Agents

**Managed Knowledge**
Organisation Knowledge and Models

## A platform, not a pile of tools

At the foundation, the Golden Pathway is built as a platform, not a collection of independent tools stitched together by process.

Core delivery systems, cloud platforms, security services and data platforms are integrated through standardised interfaces. This allows governance to be applied consistently, regardless of team, technology choice or delivery approach.

Technology decisions are abstracted away from individual teams, while control remains central and explicit.

## Policy as code, enforced by default

Security, risk and compliance controls are not documented policies. They are executable logic.

Identity, access, network controls, data handling rules and regulatory constraints are encoded as policy and enforced automatically through the AI Harness. Agents and pipelines cannot bypass these controls because they are embedded directly into execution.

This removes the gap between what policy says and what the system actually does.

## The AI Harness as the control plane

The AI Harness acts as the runtime control plane for the Golden Pathway. This is what allows people to govern outcomes and policies while removing them from the execution path.

It orchestrates agents, applies validation rules, captures lineage and reconciles outputs continuously. Every artefact is checked against approved specifications and policies before it can progress.

This is what allows the platform to operate at machine speed without sacrificing control. Governance is applied per action, not per release.

## Context as a first-class technical primitive

During engineering, agents A critical architectural decision is that agents never operate on prompts alone.

Context is provided through managed knowledge, enterprise integrations and approved artefacts. This ensures agents act on real institutional information: regulatory requirements, architectural standards, design patterns and environment-specific constraints.

By making context explicit and governed, the platform ensures predictability, auditability and trust in agent behaviour.

## From Governance intent and execution

| Governance intent | Technology mechanism | Execution outcome |
|---|---|---|
| Consistent security controls | Policy-as-code enforced by AI Harness | Controls applied uniformly, no per-team variation |
| No bypass under pressure | Runtime validation and reconciliation | Non-compliant outputs cannot progress |
| Trust in agent behaviour | Governed context via managed knowledge and integrations | Agents act on institutional intent, not prompts |
| Audit readiness | Automated lineage and evidence capture | Evidence always available, no reconstruction |
| Scalable change | Modular harness and central policy update | New rules applied without team disruption |

## Continuous evidence, zero reconstruction

Every layer of the platform is designed to generate evidence automatically.

Lineage, decisions, validations and approvals are captured as execution happens. Logs and metrics are correlated into a single trace from intent through to production behaviour. This eliminates the need for post-hoc evidence collection. Audit readiness becomes an always-on technical capability, not a periodic human activity.

## Designed for scale, change and evolution

The Golden Pathway technology stack is intentionally modular.

New agent capabilities can be introduced without weakening governance. Policies can evolve without redeploying applications. Regulations can be updated and enforced centrally without changing team workflows.

This ensures the platform can adapt as delivery velocity increases, threat landscapes evolve and regulatory expectations change.

> **The Golden Pathway works because the technology is designed to enforce what the operating model intends.**

# Case study: Adopting the AI Harness in a regulated enterprise

## Scaling when execution is governed by architecture, not by human review

As regulated financial institutions scale agent-driven delivery, the critical question is no longer whether AI can accelerate engineering, but whether control can operate at the same speed. This is where the AI Harness layer becomes essential.

In one large, highly regulated financial services environment, the AI Harness was introduced not as a productivity tool, but as a runtime control layer designed to sit inside the existing delivery lifecycle. The objective was explicit: enable agent-assisted engineering while preserving auditability, traceability and regulatory confidence at enterprise scale.

### Example: Establishing a governed execution boundary

Rather than allowing AI agents to operate directly within developer environments, the organisation introduced the AI Harness as a mandatory execution boundary. All agent-based activity, including code generation, data transformation, test creation and deployment automation, was routed through the harness.

This ensured, by design, that:

- Every agent action was context-aware and policy-constrained
- Outputs could not bypass validation or approval logic
- Governance controls executed in real time, not after release

Developers interacted with agents through familiar tools, but execution authority always sat with the harness.

### Context-driven agents, not prompt-driven behaviour

A key design decision was that agents would never operate on raw prompts alone. The AI Harness enforced the use of managed context, combining:

- Approved architectural standards and patterns
- Regulatory obligations and control requirements
- Environment-specific constraints and data access rules

By making context explicit and governed, agent behaviour became predictable, auditable and repeatable. The same task executed by different teams or agents produced consistent outcomes, regardless of delivery pressure.

This was a decisive shift from experimental AI usage to enterprise-grade capability.

## Continuous validation embedded into delivery

Validation was embedded directly into the harness rather than layered on as a pipeline stage. Every artefact generated by an agent was automatically checked against approved specifications and policies before it could progress.

This included:

- Structural and semantic validation of code and data transformations
- Alignment with security and architectural standards
- Compliance with regulatory and operational risk constraints

Non-compliant outputs were rejected immediately. There were no manual overrides and no promotion by exception, and no fast paths under pressure.

This approach removed the need for late-stage reviews that previously surfaced risk too late to be effective.

## Evidence as a by-product of execution

One of the most consequential outcomes of adopting the AI Harness was the elimination of manual evidence reconstruction.

The harness automatically captured:

- Decision lineage from intent through to production artefacts
- Validation results and control outcomes
- Deployment approvals and operational actions

This created a continuous, end-to-end audit trail without additional effort from delivery teams. Audit readiness became an operational state rather than a periodic exercise.

## Scaling safely without re-introducing friction

As agent adoption expanded across domains, the organisation was able to introduce new agent capabilities without weakening governance. Because policies, validation logic and context controls were enforced centrally by the harness, scale did not create fragmentation.

Regulatory requirements could be updated once and enforced everywhere. Delivery velocity increased without re-introducing risk or manual approval queues.

This demonstrated a critical principle: governance does not have to slow delivery if it is engineered into the execution layer itself.

> This example illustrates why the AI Harness is not optional in regulated environments. Without it, agent-driven delivery inevitably creates control gaps, invisible risk and regulatory exposure.

**Governance fails at scale not because teams ignore rules, but because controls cannot keep up with execution speed**

# The people and process transformation

## Modernising the operating model

Once governance is embedded into Golden Pathways and enforced through the AI Harness, control becomes part of execution, not a separate activity.

For finance organisations, this removes reliance on manual reviews, audit preparation and post release validation, replacing it with continuous, regulator ready control at every step.

The result is a fundamental shift in how people contribute, where accountability sits and how risk is managed at scale.

### From inspectors to platform builders

In traditional banking or insurance delivery, risk, security and operations act as downstream reviewers, validating changes after they occur and responding to exceptions under pressure.

In the Golden Pathway model, these roles move upstream:

- Controls, policies and approval logic are designed once and embedded into platforms
- Regulatory requirements such as CPS 230, data privacy and model risk are codified into execution pathways
- Governance becomes a design capability, not a review function

People stop chasing risk across projects. They engineer the system that prevents it.

### Developers inside the pathway, governance around it

For developers, the experience changes just as profoundly.

Developers no longer sit inside governance loops, negotiating approvals or reconstructing security decisions. Instead, they become consumers of a governed platform. Secure defaults, compliant infrastructure and approved architecture patterns are provided as part of the Golden Pathway.

This dramatically reduces cognitive load. Developers focus on business logic and customer outcomes, while the platform continuously enforces identity, network, security and compliance constraints in the background.

### People move into the continuous governance loop

Crucially, this does not remove people from accountability. It changes where they add value.

Security, risk, architecture and operations leaders sit around the delivery loop, not inside it. They continuously:

- Tune policies and guardrails based on telemetry and outcomes
- Adjust risk thresholds as context changes
- Improve pathway patterns as new threats or regulations emerge

Human judgment is applied to system behaviour, not individual changes. This is how governance scales without slowing delivery.

## Consistency becomes structural, not enforced

Because every application flows through the same Golden Pathway, consistency is no longer dependent on training, discipline or review. It is structural.

Security controls, architecture patterns and compliance postures are applied uniformly across teams, products and portfolios. Variation is possible only where it is explicitly designed, approved and governed.

## Audit-ready by construction

One of the most visible people impacts is the removal of manual audit preparation.

In traditional models, audits trigger weeks of effort as teams reconstruct decisions and artefacts. In the Golden Pathway, evidence is generated automatically as work progresses. Every commit, validation and release is traceable by default.

Audit readiness becomes a continuous operational state, not a periodic disruption.

## Process changes that reinforce human roles

The operating model shift is reinforced by three key process changes.

Shift-left: Vulnerability scanning, policy checks and validation move directly into the developer workflow. Issues are identified while code is being written, not after deployment.

Shift-smart: Risk decisions move from blunt “block all” rules to contextual risk scoring based on intent, environment and impact. Controls become more precise as automation increases.

Continuous feedback loops: Platform performance and delivery flow are measured using DORA metrics, including deployment frequency and lead time for change. These signals ensure that the Golden Pathway remains both safe and fast, rather than drifting back into friction.

## Resolving the Velocity Risk Paradox at a human level

Governance no longer slows delivery, and delivery no longer undermines governance. People roles become clearer, lighter and more strategic. Expertise is applied once and reused everywhere. Teams move faster because the platform carries the burden of control.

This is not just a technology upgrade. It is a modern operating model built for agent-driven delivery at scale

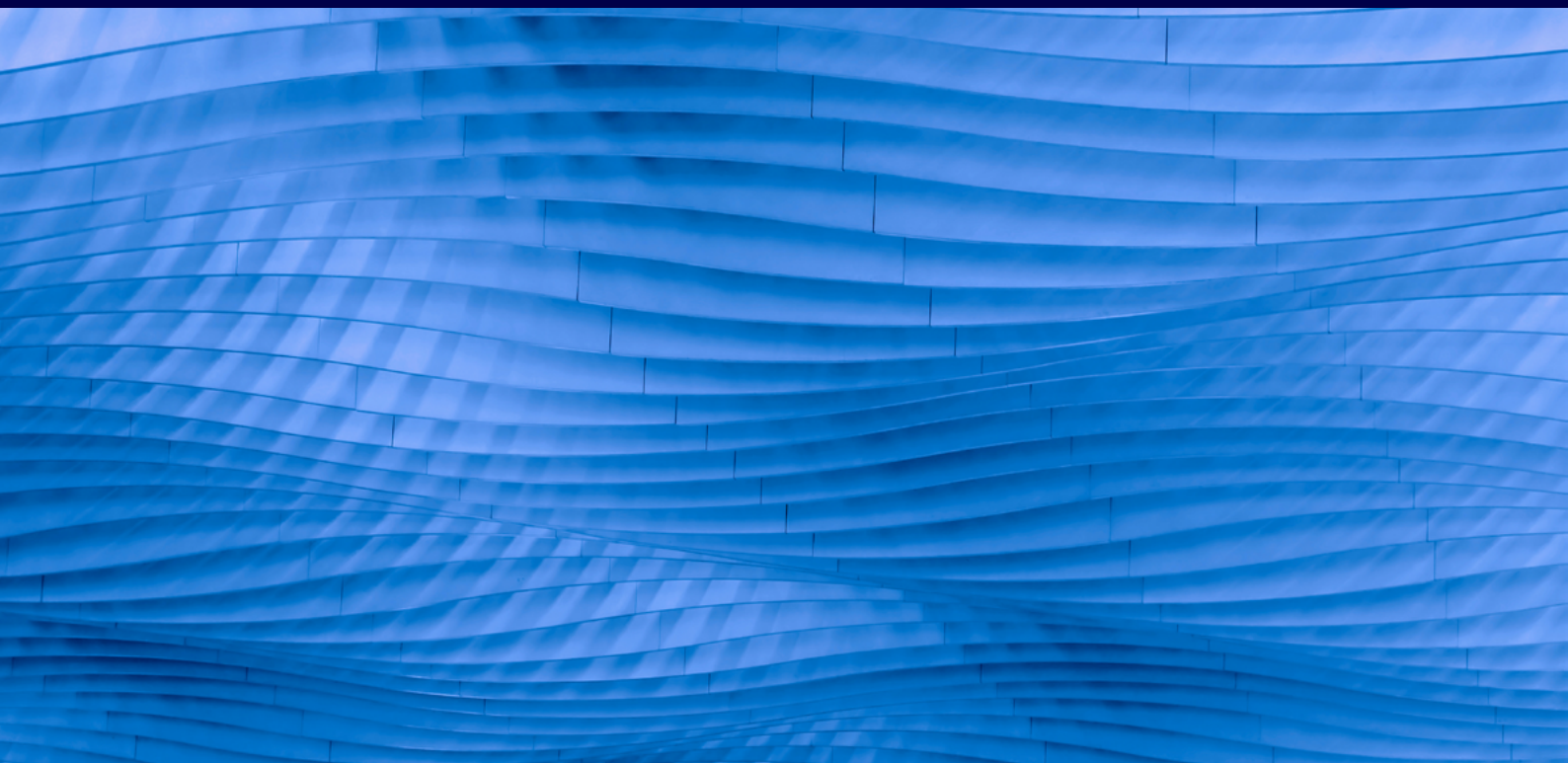
# Cognizant's offering: Risk-embedded Golden Pathways at enterprise scale

## Turning governed agentic application engineering into structural delivery advantage and regulatory confidence

Chapter 2 established the delivery architecture required to scale agentic engineering. Chapter 3 focused on what makes that architecture trustworthy in regulated environments. Cognizant services and solutions listed further introduce enterprise-grade specification design, harness-governed execution with embedded validation agents and MCP context, built-in traceability and governance, and maturity progression that together turn agentic acceleration into structural compliance confidence.

## Why Cognizant

The argument in this chapter is only credible if the tools and track record exist to back it. Cognizant built BMad and SpecKit as proprietary specification frameworks, purpose-designed for deterministic delivery in regulated environments. Agent Foundry and MAKER are operating in production on live delivery programmes, not in proof of concept. The ADLC maturity progression has been applied at major Australian, UK, and North American banks, with stage-by-stage outcomes that are measured rather than asserted. For financial services organisations that need a credible path from Stage 1 to Stage 3 within a near term horizon, Cognizant provides the delivery architecture, tooling, and embedded capability building to make that transition real.



| Offering component | What Cognizant provides | What it enables for financ organisations | Examples |
|---|---|---|---|
| **AI Harness and Validation Layer** | A governed execution environment with embedded validation, reconciliation and certification agents that enforce policy at runtime and reject non-compliant outputs automatically. | Guarantees governance at execution speed. Prevents bypass under delivery pressure. Shifts assurance from human review to architectural enforcement. | Continuous validation of data transformations, API behaviour and security posture as code is generated and deployed. |
| **Context Management and MCP Integration** | Managed context layer combining enterprise knowledge, regulatory artefacts, architectural standards and live system integrations via MCP servers. | Ensures agents act on institutional context, not generic prompts. Increases predictability, traceability and trust in agent behaviour. | Agents executing with bank-specific CPS obligations, architecture standards and environment rules embedded into context. |
| **Enterprise Platform Engineering** | Design and build of Golden Pathways as reusable, opinionated delivery routes integrated with core SDLC, cloud, security and DevSecOps platforms. | Consistent delivery foundation across products and teams. Reduces fragmentation and lowers cognitive load for developers. Accelerates onboarding and scaling. | Secure-by-default microservice, data product or AI agent pathways adopted across portfolios. |
| **Risk and Compliance by Design** | Encoding of security, risk and regulatory controls as policy-as-code, enforced centrally through the platform rather than through manual process. | Structural compliance rather than retrospective assurance. Continuous audit readiness with no manual evidence reconstruction. | Automated lineage, approval and evidence capture aligned to APRA CPS 230 expectations. |
| **Operating Model and Role Transformation** | Transition of security, risk and ops teams from inspectors to platform builders. Enablement of developers as consumers of governed pathways. | Reduces operational friction while strengthening control. Clarifies accountability and scales governance without increasing headcount. | Central platform teams defining guardrails once. Product teams shipping faster without negotiating approvals. |
| **Assessment, Maturity and Adoption Path** | A pragmatic delivery maturity assessment, followed by a staged adoption roadmap tailored to bank context and regulatory posture. | Low-risk adoption path from current state to agent-driven delivery. Clear visibility of readiness, dependencies and control gaps. | 4–6 week assessment leading to pilot Golden Pathway, then scaled adoption across domains. |

## Conclusion

# Engineering at scale in the age of AI

### Re-engineering around value streams to unlock flow, resilience and safe AI

As financial services organisations move toward agent driven engineering, the question is no longer whether delivery can be accelerated, but whether control, cost and accountability can scale at the same pace.

This paper has shown that modern engineering performance is not determined by any single factor. It is shaped by the interaction of operating model, delivery architecture, and governance model. Organisations that treat these as separate concerns will continue to see fragmented outcomes. Those that design them as a system unlock compounding advantage.

Across the three shifts explored in this paper, a consistent pattern emerges. Sustained improvement comes not from adding more teams, more tooling or more governance layers, but from redesigning how engineering work flows end to end.

### Outcome driven engineering

Reorienting delivery around value streams and persistent teams creates clear accountability for customer outcomes, resilience and cost. It replaces episodic project delivery with sustained ownership and continuous optimisation.

### AI as a governed engineering capability (ADLC)

Embedding AI into the lifecycle through specification driven, governed execution transforms productivity from individual effort into system level efficiency. Capability compounds over time as knowledge, workflows and agents are reused and improved.

### Risk embedded Golden Pathways

Integrating governance directly into execution removes the gap between delivery and control. Compliance becomes structural, audit readiness becomes continuous, and risk is managed at the speed of change.

Cognizant's experience across regulated financial institutions shows that these shifts are not theoretical. When applied together, they create measurable improvements in delivery velocity, cost efficiency, resilience and regulatory alignment, while establishing a platform that strengthens with scale rather than becoming more complex.

Organisations that move early are not simply accelerating delivery. They are redesigning their engineering system to create a durable, compounding advantage in how technology drives business outcomes.

**Cognizant partners with banks, insurance and financial services organisations to move from assessment to implementation, bringing together operating model design, ADLC delivery capability, and Golden Pathway engineering to establish governed, scalable engineering systems.**

**We welcome the opportunity to discuss how these principles apply within your organisation and to explore a practical path toward building an engineering system that delivers sustained outcomes at scale.**

Email us: contact_au@cognizant.com

Cognizant (Nasdaq: CTSH) is an AI Builder and technology services provider, bridging the gap between AI investment and enterprise value by building full-stack AI solutions for our clients. Our deep industry, process and engineering expertise enables us to build an organization's unique context into technology systems that amplify human potential, drive tangible outcomes and keep global enterprises ahead in a fast-changing world. See how at cognizant.ai or @cognizant.


### World Headquarters

300 Frank W. Burr Blvd.
Suite 36, 6th Floor
Teaneck, NJ 07666 USA
Tel: +1 201 801 0233

### European Headquarters

280 Bishopsgate
London
EC2M 4AG
England
Tel: +44 (01) 020 7297 7600

### India Corporate Office

Siruseri-Software Technology Park of India (STPI)
SDB Block—Ground Floor North Wing
Plot No H4, SIPCOT IT Park
Chengalpattu District
Chennai 603103, Tamil Nadu
Tel: 1800 208 6999

### APAC Headquarters

1 Fusionopolis Link, Level 5
NEXUS@One-North, North Tower
Singapore 138542
Tel: +65 6812 4000





WF4833151