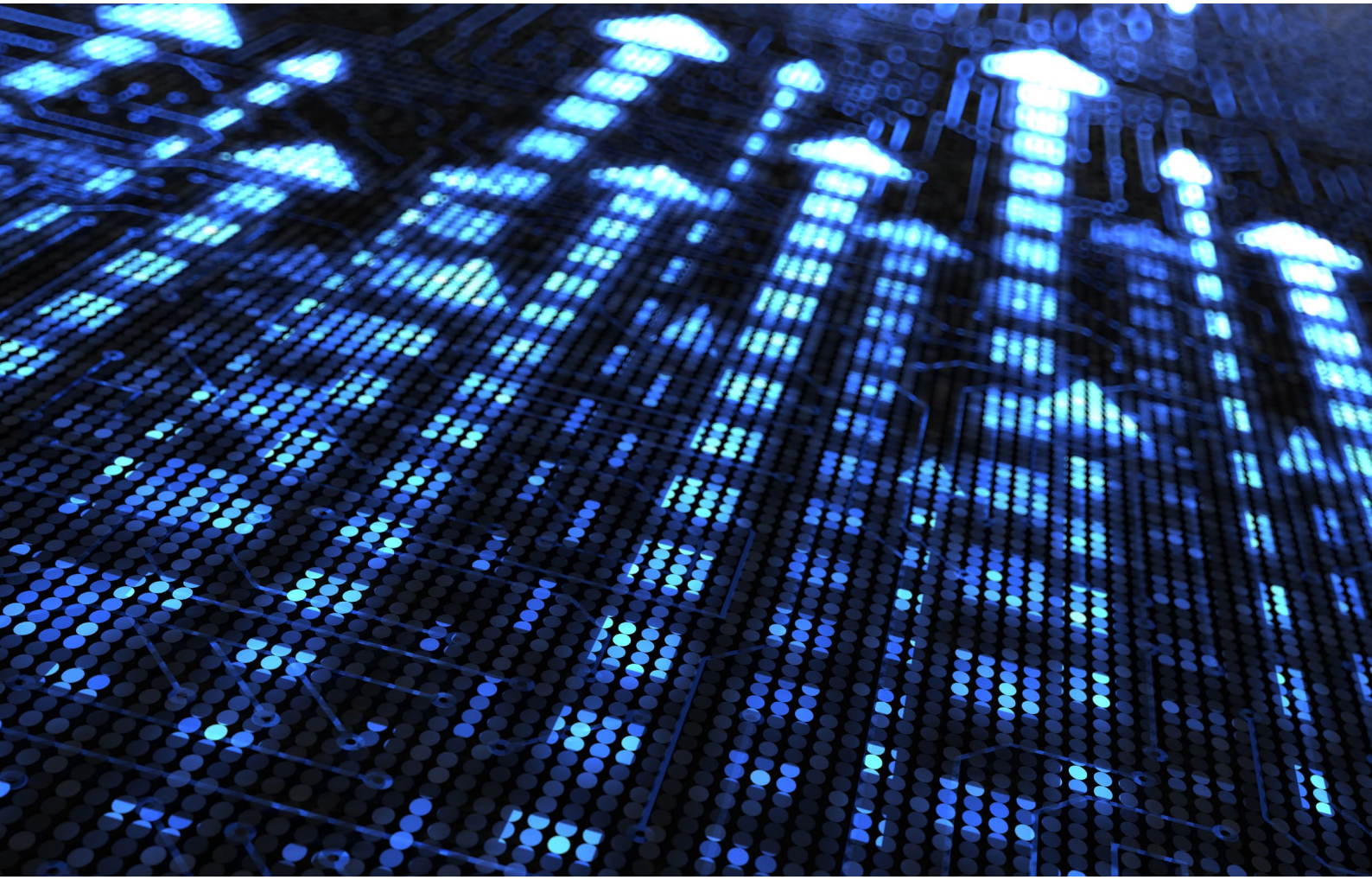
Luxoft
A DXC Technology Company



Choosing the right managed delivery model

by Balaji Venkatramani
Head of APAC Delivery Solutions,
Luxoft India

An organization's engagement with their IT vendors in a managed delivery model increases delivery ownership and risk levels. The client benefits from things like the freeing-up of in-house expertise for strategic initiatives, as well as increased efficiency, predictability and cost optimization.

There are three main types (models) of managed delivery engagement:

Autonomous squads

- Limited features and functionality within a single or small group of applications
- Delineated roles, possibly including specific onshore, nearshore or offshore locations
- Development or support (potentially) in scope

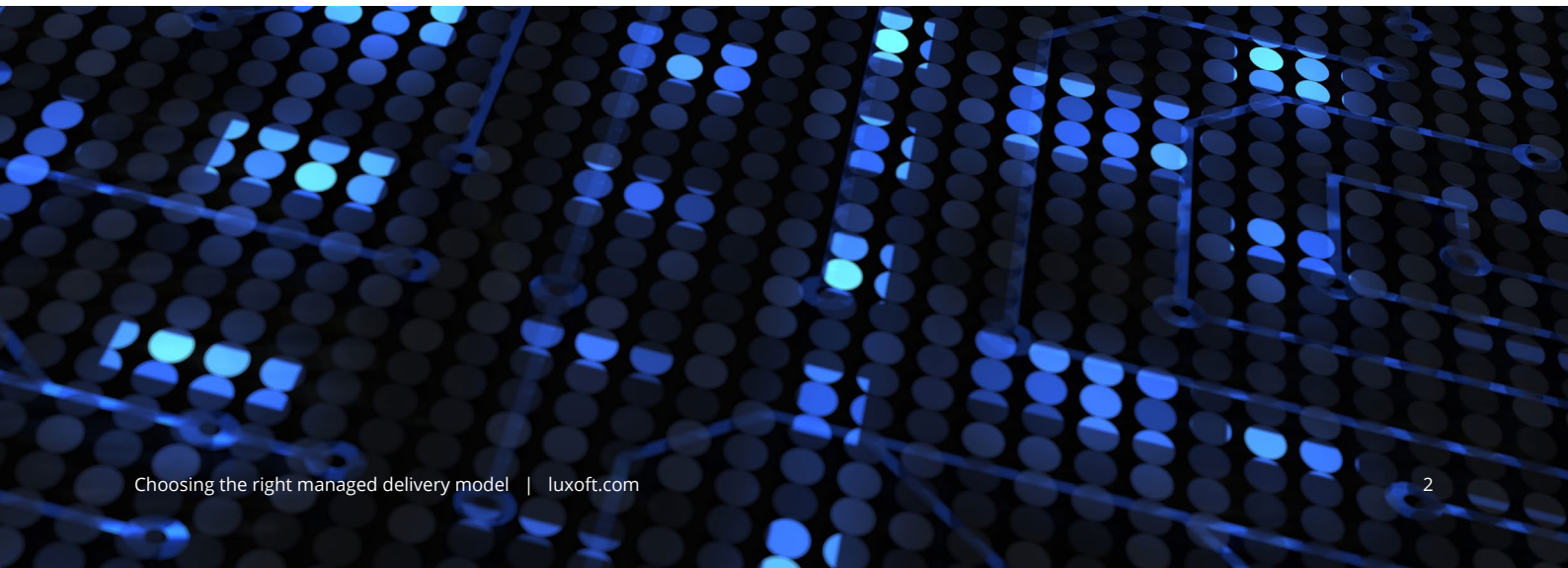
Portfolio management

- Complete portfolio of applications for end-to-end business functionality, products or services
- Includes all the relevant functional, technical, process, project, program and portfolio management skills
- Development and/or support in scope

Application ownership

- Logical group of applications with corresponding business functionality and all-encompassing roles
- Includes all the functional, technical, process and project management skills needed to manage the application group
- Development or support (potentially) in scope

Together, we're going to tackle these key models in detail, examining the different levels of benefit, characteristics and selection criteria, as well as weighing their pros and cons.



Main building blocks of the different models of a managed delivery engagement

Team composition

- Role distribution
- Squad structure
- Responsibilities segregation

Ways of working

- Development and support methodologies
- Project and program management approach
- Governance structure

Delivery performance model

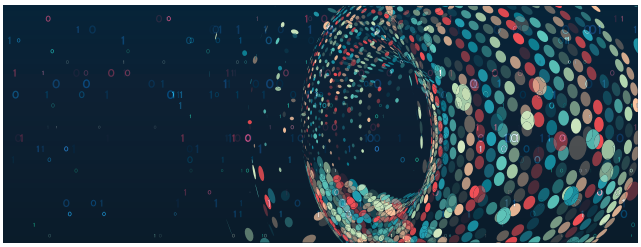
- Performance SLAs and KPIs
- Performance risk management and reporting

Pricing models

- Charging mechanics
- Benefits model

Knowledge management framework

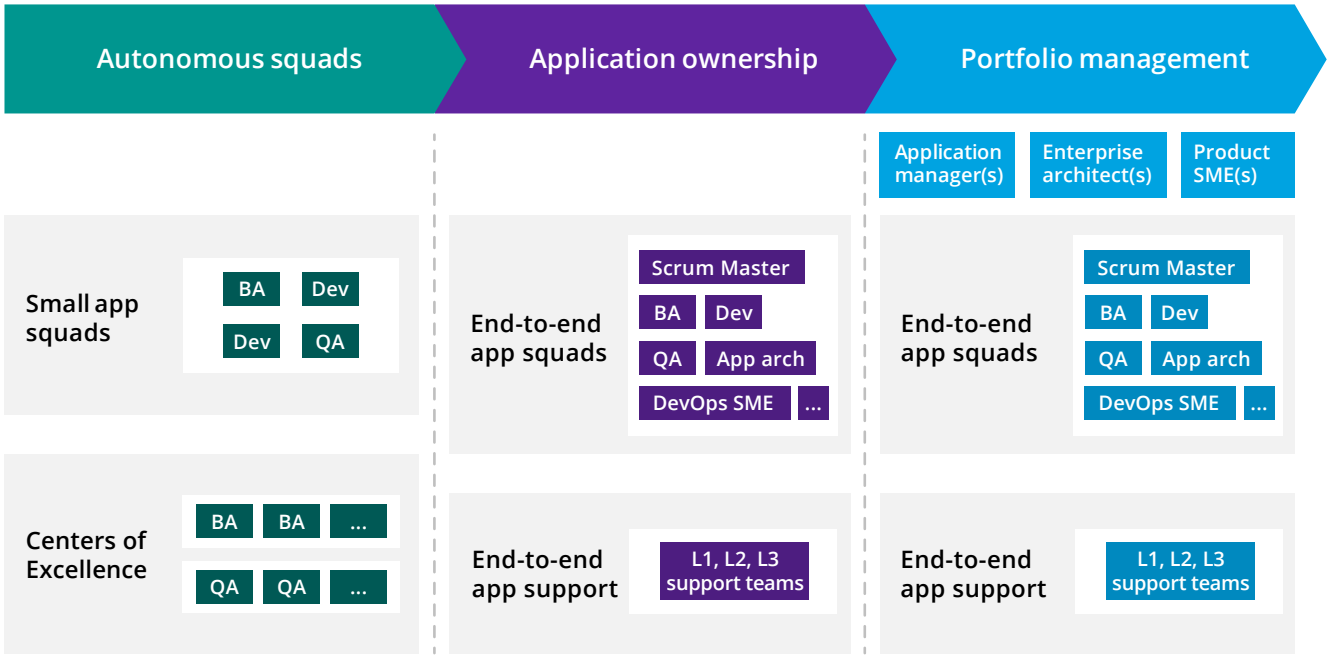
- Knowledge capture
- Knowledge risk management
- Continuous improvement



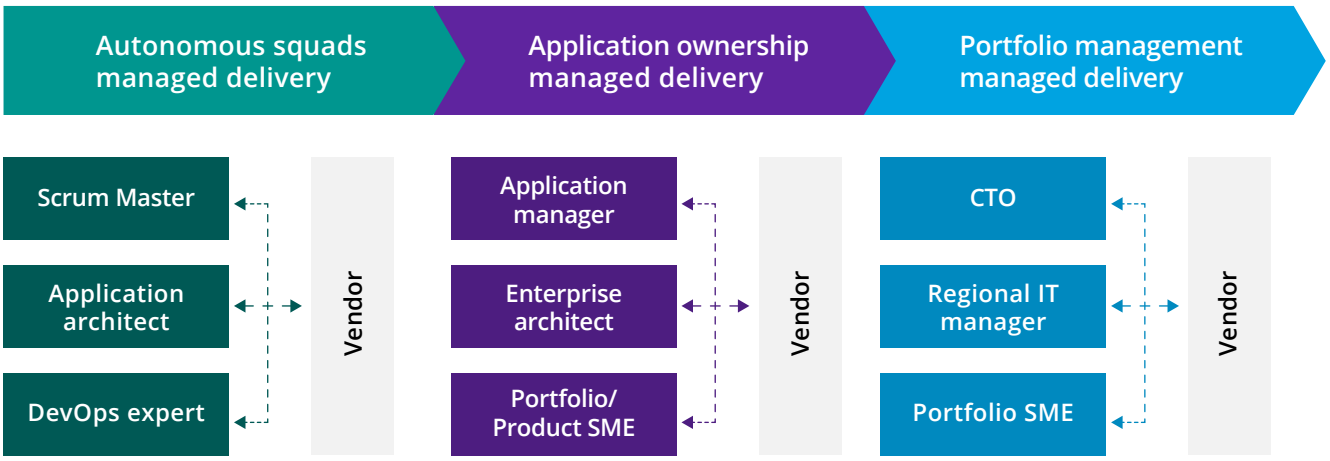
Innovation and transformation model

- Innovation and transformation scope
- Partner incentivization

Team composition



Vendor team formation



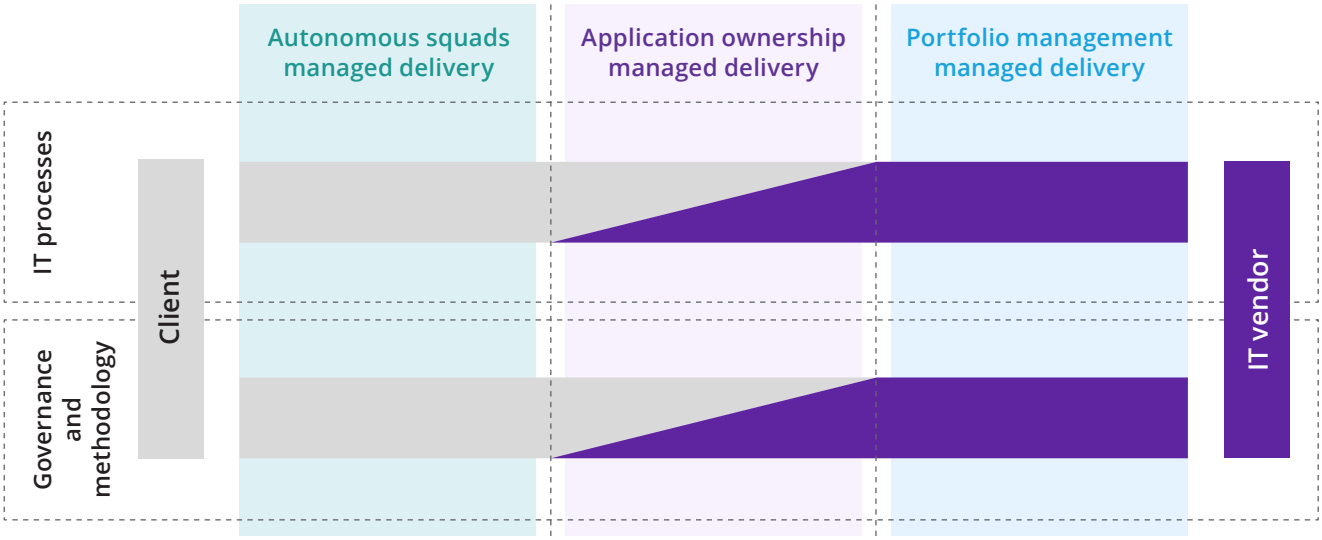
Client-vendor interaction

Role	Autonomous squads	Application ownership	Portfolio management
CTO	Client	Client	Client
Regional management	Client	Client	Client
Portfolio expertise	Client	Client	Client
Application management	Client	Client	Vendor
Product expertise	Client	Client	Vendor
Scrum Master	Client	Vendor	Vendor
Architect	Client	Vendor	Vendor
DevOps	Client	Vendor	Vendor
Development	Vendor	Vendor	Vendor
Analysis	Vendor	Vendor	Vendor
QA	Vendor	Vendor	Vendor

Roles and responsibilities

Delivery model	Key characteristics
Autonomous squads	• Client gives vendor independent but limited roles • Squad structure skewed towards specific roles such as business analysts, testers, developers and support analysts, rather than all-encompassing • Scrum master role performed by client not vendor • Vendor responsibilities restricted to analysis, development, testing and support of the carved-out scope • Key responsibilities such as management of end-to-end project and program deliverables, business stakeholders, scope and product backlog, schedule, and feature prioritization are with the client • Vendor given independence to manage squads, prioritize and allocate work in the defined scope, own delivery risks and report status
Application ownership	• Vendor has independent and all-encompassing roles for the specific applications in scope • Squad structure may include key roles such as scrum master, business analyst, application architect, developer and tester • Vendor responsibilities involve end-to-end managed delivery for that application group, including analysis, development, testing or support • Vendor responsible for managing project deliverables, associated business stakeholders, scope and scrum backlogs, schedule, effort and quality for in-scope applications • Client responsible for end-to-end program and portfolio deliverables, business stakeholders, scope and product backlog, architecture decisions, schedule and feature prioritization • Vendor given independence to manage their scope of applications, squads, prioritize and allocate work in the defined scope, own delivery risks and report status • Usually, client deploys portfolio or application managers to fulfill responsibilities and interface with rest of the firm, including the business
Portfolio management	• Vendor has independent and all-encompassing roles for that portfolio • Different forms of squad and team structure may be required, such as business-vertical aligned and shared-services aligned based on technology, service, support, infrastructure and other scope of work • All portfolio, program, project, functional, technical, process, development, maintenance, support, transformation and enabling roles in vendor scope • Vendor responsibilities involve end-to-end managed delivery for that portfolio • Vendor responsible for managing interface with teams and stakeholders, including business and other vendors • Client still provides business-user-community roles, chief technology officer role, plus architecture guidelines, global or regional IT portfolio manager and other enterprise-level roles • Vendor given complete independence to manage portfolio and associated risks

Ways of working



Level of client influence

Delivery model	Key characteristics
Autonomous squads	• Mainly aligned to client's defined development and support methodologies • Limited project management by vendor for defined scope and squads. No program management in scope • Mostly, vendor deploys a project manager to represent their squads • Client's governance structure and methodology followed
Application ownership	• Aligned to client's defined development and support methodologies, although vendor might introduce custom processes for scope of applications • Project management by vendor for defined scope of applications and squads, while overall program management scope is with client • Vendor deploys a project or program manager to represent their squads in most cases • Governance structure and methodology harmonized between best practices of the vendor and client
Portfolio management	• Vendor has a significant say in defining ways of working, and performs a harmonization exercise with the client's processes • All project, program and portfolio management by vendor with minimal effort required from the client • Governance structure and methodology is mainly vendor-defined and normally harmonized with client's methodology

Delivery performance model

Metrics	Autonomous squads	Application ownership	Portfolio management
Focus area	Narrow squad level	Wider app level	Portfolio level
Schedule	✓	✓	✓
Quality	✓	✓	✓
Efficiency		✓	✓
Cost		✓	✓
CSAT			✓
Governance		✓	✓
Knowledge		✓	✓
Risk mitigation			✓
Innovation and transformation			✓
SLAs/Service credits	✓	✓	✓

Comparison of metric attributes

Delivery model	Key characteristics
Autonomous squads	• Metrics focused on schedule and quality • Service credits implemented as a risk-mitigation measure • Vendor contributes to overall engagement performance. Reporting owned by the client
Application ownership	• Metrics focused on schedule, quality and efficiency • Service credits implemented as a risk-mitigation measure • Vendor owns the engagement performance reporting for their scope of applications
Portfolio management	• Variety of metrics employed with holistic focus on schedule, quality, efficiency, cost, governance, knowledge, client satisfaction, risk mitigation, innovation and transformation • Service credits implemented as a risk-mitigation measure • Vendor owns end-to-end portfolio performance reporting for all key stakeholders from the client

Pricing models

Commercial models	Autonomous squads	Application ownership	Portfolio management
Fixed price	✓	✓	Portfolio level
Squad-based pricing	✓	✓	✓
Feature-based pricing		✓	✓
Story-point-based pricing		✓	✓
Gain-share models			✓
Benefits	Fixed for engagement	Evolving over engagement duration	Evolving over engagement duration. Gain-share models can bring further benefits

Compare pricing options

Delivery model	Key characteristics
Autonomous squads	• Fixed price for a defined scope of work and duration • Squad-based price for a specific set and count of roles for a specified duration • Fixed benefits for client based on up-front agreements and rolling benefits over engagement duration rarely in scope
Application ownership	• Fixed price for a defined scope of work and duration • Squad-based monthly price for each logical squad • Feature- or story-point-based monthly price for group of applications in scope • Evolving benefits from efficiencies gained over duration of engagement
Portfolio management	• Outcome-based pricing model contingent upon defined outcomes at project, program or portfolio level • Squad-based monthly price for each logical squad • Feature- or story-point-based monthly price for logical business functionality, product or service • Shared gain models to favor client and incentivize vendor for delivering benefits from transformation beyond base commitments • Evolving benefits from efficiencies gained over duration of engagement

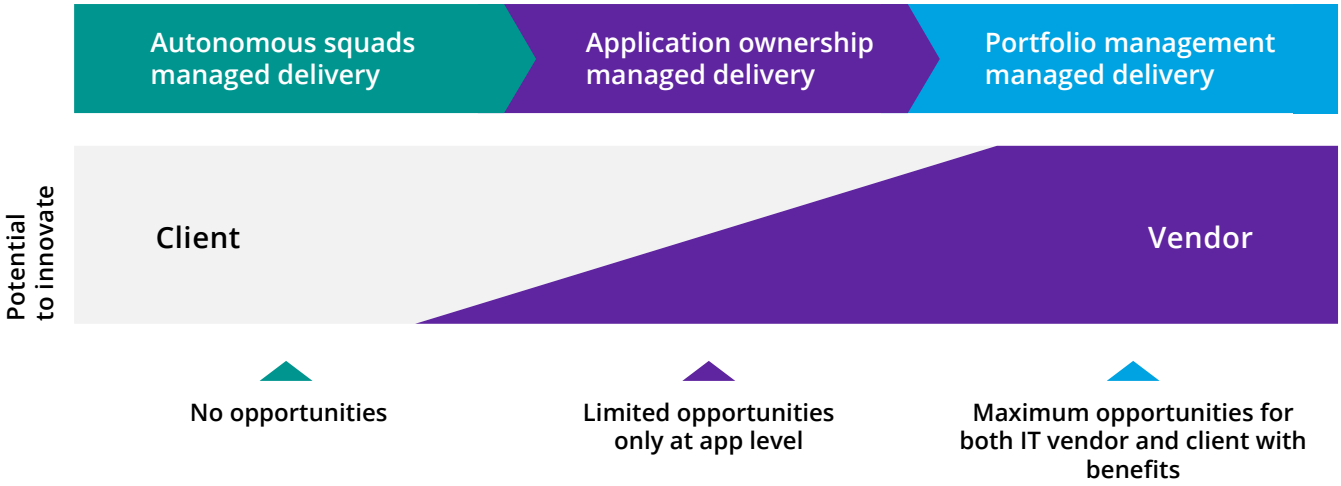
Knowledge management framework

Autonomous squads managed delivery		Application ownership managed delivery		Portfolio management managed delivery	
Client	**Vendor**	**Client**	**Vendor**	**Client**	**Vendor**
Overall knowledge management scope All SMEs	Limited squad-level knowledge scope	Product and portfolio knowledge management scope Product and portfolio SMEs	Wider app-level knowledge scope and management App SMEs	None	Apps, product and portfolio knowledge management scope Apps, product and portfolio SMEs

Client-vendor responsibilities

Delivery model	Key characteristics
Autonomous squads	• Selective knowledge capture limited to scope of vendor • Overall knowledge risk and management is client's responsibility • Vendor expected to manage knowledge and risks for their carved-out scope, and demonstrate continuous improvements
Application ownership	• Relevant knowledge capture for applications in vendor scope as well as required knowledge on ecosystem of interfacing applications • Vendor responsible for knowledge risk, management and continuous improvement for in-scope applications • Client assumes overall knowledge risk and management responsibility for the portfolio
Portfolio management	• Vendor responsible for portfolio-wide knowledge capture, risk management and continuous improvement

Innovation and transformation model



Client-vendor opportunities

Delivery model	Key characteristics
Autonomous squads	• Not in scope
Application ownership	• Limited innovation and transformation scope within the applications managed by vendor • Scope for incentivizing vendor to demonstrate higher-than-committed benefits with additional innovation effort
Portfolio management	• Significant innovation and transformation scope in the portfolio managed by vendor • Significant scope for incentivizing vendor to demonstrate higher-than-committed benefits with additional innovation effort

Choosing the right model

Autonomous squads managed delivery

For clients:

- Experimenting with a managed delivery model prior to sharing a significant scope of work with vendors
- Looking to form centers of excellence for specific skills and roles (provides a reliable supply of niche skills)
- Aiming to introduce a managed delivery approach in small scale applications or systems

Application ownership managed delivery

For clients:

- Transferring responsibility and risks for a subset of business functionality to vendor, and refocusing corresponding in-house expertise on more strategic business functions
- Driving efficiencies in a specific group of applications representing a logical business scope
- Iteratively developing a specific set of solutions from a larger portfolio of features
- Setting up an independent maintenance and support team for a logical group of independent applications

Portfolio management managed delivery

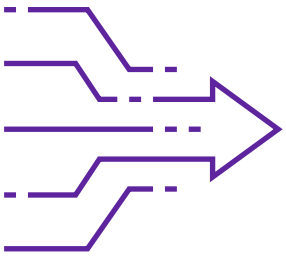
For clients:

- Transferring responsibility and risks for an entire portfolio of applications, products or services and refocusing corresponding in-house expertise on other strategic platforms
- Driving portfolio-wide transformation and efficiencies for multiple business clusters within a vertical, or shared services across verticals
- Iteratively developing solutions and products for a business portfolio
- Embarking on vendor-consolidation initiatives
- Setting up an independent maintenance and support team for a complete business portfolio

Other managed delivery models

You might come across other models too:

- A hybrid of the three models involving customized roles and responsibilities, segregation between client and vendor, unique benefits models and a combination of services
- Cross-portfolio managed-services models for a specific catalogue of services like: Level 1 support, Quality Assurance as-a-Service, Program Management as-a-Service, DevOps as-a-Service and so on



In summary

Client benefits	Staff augmentation	Managed delivery		
		Autonomous squads	Application ownership	Portfolio management
Knowledge risk mitigation	Medium	High	Medium	Low
Organization change risk mitigation	None	High	Medium	Low
Influence on ways of working	High	High	Medium	Low
Simplified interfacing of client and vendor teams	Low	Low	Low	High
Diverse pricing models for better benefits	None	Low	Medium	High
Scope for transformation and value addition	None	Low	Low	High
Objectivity of delivery performance and continuous improvements	None	Low	Medium	High

Benefits vary across the models, so the client simply chooses the best fit.

Pros and cons

Autonomous squads

Good:

- Excellent model for experimenting with organizational change during the adoption of small-scale managed delivery
- Minimal organizational changes and knowledge risks
- Complete control over ways of working

Less than ideal:

- Limited cost and efficiency benefits
- Greater share of roles and responsibilities requires more governance effort
- Increased complexity from aligning in-house and cross-vendor teams
- Not much scope for innovative pricing models
- No transformation

Application ownership

Good:

- Flexible pricing provides greater benefits
- Enhanced benefits from efficiencies
- Less governance effort, more vendor accountability
- Increased control over ways of working
- Tighter management of key business, technology and process knowledge retained in-house at portfolio level

Less than ideal:

- Greater (but manageable) client-change effort
- Vendor's managed-delivery-ownership benefits might be reduced by architecture plus key SME and portfolio management responsibilities being retained in-house
- More complex because of the need to align in-house and cross-vendor teams
- Limited scope for innovation and transformation because the vendor owns delivery for only a subset of the portfolio

Portfolio management

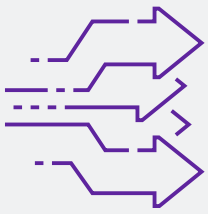
Good:

- Flexible pricing provides significant benefits
- Benefits from efficiencies
- Minimal governance effort with extensive vendor accountability
- Greatly reduced complexity from aligned portfolio-level ownership and scope with vendor
- Portfolio-level innovation and transformation

Less than ideal:

- Minimal control on ways of working
- Largely depends on the vendor for knowledge
- Major client-change effort which most likely needs third-party management

In summary, the client needs to choose the managed delivery model that's the best fit, or a hybrid of models based on their risk tolerance, past experience, benefits and transformation requirements, level of comfort with vendors, client readiness or similar. Luxoft has partnered extensively with many clients, advising on the selection of engagement models and, subsequently, implementing those models.



About **the author**

Balaji Venkatramani
Head of APAC Delivery Solutions,
Luxoft India

Balaji is a senior director with Luxoft India, heading the Digital Delivery Strategy and Solutions for BCM APAC. He has over 21 years' experience in the IT industry. Balaji has driven large-scale technology solutions and transformation initiatives in Silicon Valley technology companies as well as service partnerships with global financial clients. He specializes in large-scale knowledge transitions, transformations, Agile, DevOps, big data and analytics, cloud and program management.

About Luxoft

Luxoft is the design, data and development arm of DXC Technology, providing bespoke, end-to-end technology solutions for mission-critical systems, products and services. We help create data-fueled organizations, solving complex operational, technological and strategic challenges. Our passion is building resilient businesses, while generating new business channels and revenue streams, exceptional user experiences and modernized operations at scale.

luxoft.com