



La ventaja del contexto empresarial en la era agéntica

El futuro digital de las organizaciones no está en **programar agentes**, sino **enseñarles a operar la compañía**

Cuando el código se vuelve abundante, la ventaja competitiva migra hacia el contexto y la capacidad de delegar trabajo con control.

Strategia | Thought Leadership
AGO 2026

Resumen ejecutivo

La inteligencia artificial está reduciendo de forma radical el costo de producir software. Generar código, pruebas, documentación e integraciones será cada vez más una actividad dirigida y verificada por humanos, pero ejecutada en gran medida por máquinas. Esto no significa que la ingeniería desaparezca. Significa que su centro de gravedad se mueve desde escribir sintaxis hacia especificar intención, diseñar arquitectura, aportar contexto, validar comportamiento y asumir responsabilidad.

Para la empresa, la consecuencia es más profunda: el nuevo cuello de botella ya no será programar un agente, sino lograr que ese agente entienda cómo opera realmente la organización. Un agente genérico puede redactar, resumir o consultar. Un agente empresarial debe reconocer objetivos, políticas, excepciones, relaciones, niveles de autoridad, fuentes confiables, riesgos, lenguaje, prioridades y criterios de éxito. Debe saber no solo qué hacer, sino por qué, con qué evidencia, hasta dónde puede actuar y cuándo debe detenerse.

Ese aprendizaje no se resuelve con un "prompt" ni con cargar documentos en una base de conocimiento. Requiere convertir la inteligencia organizacional, especialmente la que vive en conversaciones, hábitos y juicio experto, hacia un modelo operativo que pueda ser compartido por personas y agentes. En la práctica, entrenar ese sistema exige más trabajo de negocio, psicología, diseño organizacional basado en buenas prácticas y gobierno que programación.

La oportunidad económica es significativa. Experimentos con IA generativa han observado mejoras de 14% en productividad de soporte al cliente, 40% menos tiempo en tareas profesionales de escritura, 25.1% menos tiempo en tareas de consultoría dentro de la frontera de capacidad y 55% menos tiempo en una tarea acotada de programación. Sin embargo, la misma evidencia advierte que el valor no es automático: en una tarea gerencial fuera de la frontera de capacidad, los usuarios de IA fueron 19% menos propensos a llegar a la respuesta correcta; y un estudio de 2025 encontró que desarrolladores experimentados trabajando sobre repositorios grandes y maduros tardaron 19% más con herramientas de IA.

La IA Agéntica multiplica la capacidad operativa cuando el trabajo está diseñado para que el agente reciba contexto suficiente, actúe dentro de una frontera conocida, aprenda de la retroalimentación y escale el juicio que no debe automatizar.

Por eso, la inversión decisiva de los próximos años **no será solamente tecnológica**. Será la construcción de **capacidad operativa delegada: procesos que combinan ejecución agéntica con dirección humana, permisos explícitos, trazabilidad, métricas y aprendizaje continuo**.

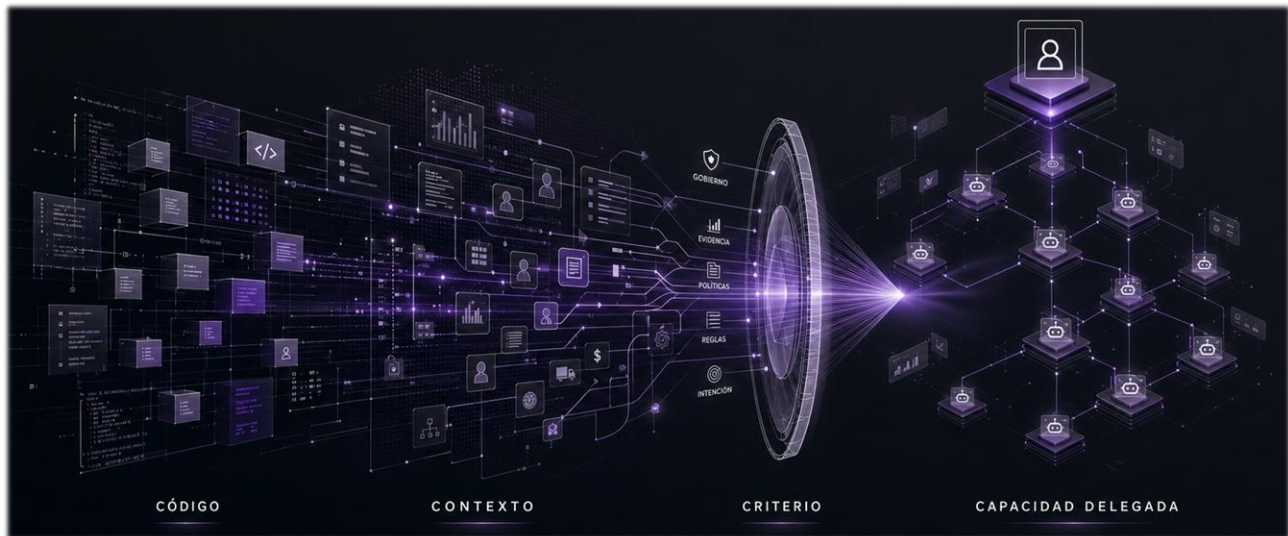
Desde el punto de vista de StrategIA, esta transición redefine la unidad de transformación. El objetivo ya no es desplegar agentes ni automatizar tareas individuales. Es rediseñar el modelo operativo para mover trabajo, decisiones y aprendizaje hacia una fuerza laboral híbrida que responda a KPI, políticas y límites de autoridad.



Gustavo Justicia C.
CEO, StrategIA

Ejecutivo experto en transformación empresarial e inteligencia artificial, con más de 16 años de experiencia en consultoría de negocio e implementación tecnológica en Latinoamérica y Estados Unidos.

Nuestra visión del futuro cercano de las organizaciones



1. La generación de código se convertirá en una capacidad abundante

La IA ya acelera tareas de desarrollo acotadas y reduce el esfuerzo cognitivo asociado al trabajo repetitivo. Nos encontramos en la era de transición. Antes era necesario tener equipos muy grandes de programación para generar software. En el futuro cercano más del 90% del código será generada por IA. Pero generación no equivale a ingeniería completa. Definir el problema, comprender el sistema, administrar dependencias, validar seguridad y operar el cambio siguen siendo responsabilidades de diseño y juicio.

2. El contexto organizacional será el recurso escaso

Un modelo puede conocer prácticas universales. No conoce automáticamente cómo se aprueba una excepción en una compañía, qué cliente requiere cuidado especial, qué dato es legalmente utilizable, qué regla informal evita un error o quién debe intervenir cuando dos objetivos entran en conflicto.

3. Entrenar agentes se parecerá más a desarrollar profesionales que a programar funciones

El agente necesita demostración, práctica situada, feedback, límites de autoridad y exposición gradual a casos más complejos. Esa lógica se aproxima al aprendizaje cognitivo y al desarrollo de modelos mentales compartidos, no al despliegue tradicional de software.

4. El modelo operativo humano-agente debe diseñarse como un sistema sociotécnico

La tecnología y la organización no pueden optimizarse por separado. Procesos, roles, datos, incentivos, confianza, controles y plataformas deben rediseñarse juntos. El objetivo no es reemplazar personas ni "añadir bots", sino crear equipos híbridos capaces de producir más valor con mejor control.

5. El tiempo liberado por la programación se reasignará a contexto, entrenamiento y control

El trabajo no desaparece, cambia de naturaleza. Menos tiempo se dedicará a transformar manualmente especificaciones en código. Más tiempo se invertirá en hacer explícito el conocimiento, definir la intención, preparar escenarios, evaluar resultados, gobernar permisos y mejorar el sistema.

Nuestro punto de vista: de agentes desplegados a capacidad operativa delegada

La conversación de mercado suele comenzar en la tecnología: modelos, prompts, herramientas, memory y orquestadores. Pero la pregunta no es qué puede hacer un agente en abstracto, sino **qué parte del modelo operativo puede delegarse de manera confiable para mover un KPI**.

StrategIA Principles: Principios de un Modelo Operativo Agéntico Exitoso

1. Permitir la dirección humana mientras la ejecución es autónoma

La autonomía agéntica es un medio para ampliar capacidad, no un fin. Las personas conservan propósito, accountability y criterio sobre las decisiones materiales. Los agentes ejecutan, coordinan y aprenden dentro de una frontera explícita.



2. El proceso end-to-end es la unidad de diseño

Automatizar una actividad aislada puede acelerar una cola y empeorar el flujo completo. El diseño debe seguir la creación de valor a través de funciones, sistemas y handoffs, y ubicar a humanos y agentes donde su contribución sea superior.



3. Conectar desde la ambición hasta la operación

La transformación debe recorrer una secuencia disciplinada. Cada etapa aumenta la madurez del agente y de la organización que lo entrena.



4. Gobernar la autoridad y el control agéntico

Evaluar el apetito de riesgo y traducirlo a permisos verificables. Define qué puede observar, recomendar, preparar, aprobar o ejecutar el agente; qué evidencia necesita; y cómo se revoca la autonomía.



5. Medir el impacto bajo métricas reales

Ejecutar una actividad tecnológica como cantidad de tokens utilizados, tickets cerrados, líneas de código, etc no equivale a generar valor. La dirección necesita ver productividad, calidad, riesgo, adopción, aprendizaje y capacidad delegada en una misma cadencia.



Para StrategIA el modelo operativo del futuro combina autonomía de ejecución con dirección humana, accountability clara y control basado en evidencia.

1. Cuando el código se vuelve abundante, la intención se vuelve estratégica

Una parte importante del costo de la transformación digital ha estado ligada a traducir necesidades de negocio a instrucciones que una máquina pudiera ejecutar. Esa traducción exigía analistas, arquitectos, programadores, testers, integradores y ciclos prolongados de entrega.



Ahora con la IA generativa ese ciclo es cada vez más corto. En un experimento controlado de GitHub con **95 desarrolladores profesionales**, el grupo que utilizó Copilot completó una tarea acotada de programación **55% más rápido**. El resultado muestra que la producción de código repetible ya puede acelerarse materialmente.

Al mismo tiempo, un estudio de METR con 16 desarrolladores experimentados, **246 issues reales y repositorios de más de un millón de líneas encontró una desaceleración de 19% cuando se permitía usar IA.**^[5] La aparente contradicción nos señala que la IA funciona mejor cuando el problema está acotado y el contexto necesario cabe en la interacción, pero su

ventaja cae cuando el trabajo depende de conocimiento profundo de un sistema maduro, estándares implícitos, arquitectura histórica y costos elevados de revisión.



Por eso, hablar de programación “prácticamente automatizada” requiere precisión. Es razonable anticipar un estado en el que casi todo el código de aplicación sea generado por IA. No es razonable confundir ese estado con software autónomamente correcto.

La producción de sintaxis puede acercarse a la automatización total; **la ingeniería seguirá requiriendo especificación, arquitectura, evaluación, seguridad, operación y accountability.**

La unidad de valor deja de ser la línea de código. Pasa a ser la calidad de la intención ejecutable y debemos resolver las siguientes preguntas:

1. ¿Qué resultado de negocio se busca?
2. ¿Qué restricciones no pueden violarse?
3. ¿Qué fuentes constituyen evidencia?
4. ¿Cómo se reconoce una excepción?
5. ¿Qué decisiones puede tomar el agente?
6. ¿Qué comportamiento exige aprobación humana?
7. ¿Cómo se mide el éxito y cómo se revoca la autonomía?

La nueva disciplina empresarial será convertir estas respuestas en un entorno donde personas y agentes puedan operar de manera coherente.

La ingeniería de mayor valor se desplazará desde la sintaxis hacia la arquitectura de intención, contexto, evaluación y control.

2. Las organizaciones más allá de la documentación

Las organizaciones suelen imaginar que “dar contexto” a un agente consiste en conectar políticas, manuales y bases de datos. Esa es una base necesaria, pero insuficiente.



Nonaka describió la creación de conocimiento organizacional como un diálogo continuo entre conocimiento tácito y explícito. Las personas originan conocimiento; la organización lo articula, combina y amplifica.

El reto agéntico puede leerse como una nueva vuelta de esa espiral: hacer visible suficiente conocimiento tácito para que el sistema pueda observarlo, practicarlo y reproducirlo, sin asumir que todo juicio puede codificarse.

Esto convierte el entrenamiento de agentes en un trabajo de descubrimiento organizacional. Se deben capturar no solo contenidos, sino también relaciones:

	Objetivo	↔	Proceso		El agente aprende a Interpretar ↓ Decidir ↓ Actuar ↓ Aprender
	Decisión	↔	Responsable		
	Dato	↔	Fuente y nivel de confianza		
	Excepción	↔	Criterio de escalamiento		
	Acción	↔	Permiso		
	Resultado	↔	KPI		
	Error	↔	Aprendizaje		

Un agente que recupera el documento correcto **puede seguir tomando una mala decisión** si no comprende cuál política prevalece, qué parte está vigente, qué nivel de riesgo es aceptable o quién tiene la última palabra.

La organización también vive en:

- la diferencia entre el proceso diseñado y el proceso ejecutado
- las excepciones que nunca llegaron al procedimiento
- la forma en que un experto reconoce un riesgo antes de poder explicarlo
- las relaciones entre clientes, proveedores, unidades y personas
- los trade-offs que la dirección acepta en una situación y rechaza en otra
- el lenguaje con el que la organización interpreta calidad, urgencia y valor
- la memoria de decisiones anteriores y las razones que las justificaron



“El entrenamiento real ocurre cuando el experto hace visible cómo interpreta señales, excepciones, políticas y trade-offs.”

3. La mina de oro agéntica: El Contexto Empresarial

La integración humano-agente no es únicamente un problema de arquitectura tecnológica, inteligencia artificial o un problema de procesamiento de lenguaje natural (NLP). Es un problema de coordinación, aprendizaje y confianza.

Modelos mentales compartidos

Los equipos efectivos desarrollan representaciones compatibles sobre la tarea, las funciones, los objetivos y la forma de coordinación. La investigación sobre modelos mentales compartidos muestra una relación positiva con los procesos y el desempeño del equipo. Para un equipo humano-agente, esto significa que el agente debe compartir una representación operativa suficiente:

- Qué objetivo colectivo está optimizando
- Cómo se distribuyen las responsabilidades
- Qué puede esperar de cada rol
- Qué señales indican que el plan dejó de ser válido
- Cómo coordinar sin solicitar instrucciones para cada decisión pequeña.

Un agente sin modelo mental compartido puede ser técnicamente competente y operacionalmente disruptivo.

Memoria transactiva

La memoria transactiva no exige que todos sepan todo. Exige que el equipo sepa quién sabe qué y confíe en ese mapa de especialización. En un sistema agéntico, la memoria no debe reducirse a “buscar en todos los documentos”. El agente necesita reconocer:

- Cuál sistema es la fuente oficial
- Qué experto posee conocimiento no documentado
- Qué agente especializado debe ser convocado
- Cuándo una respuesta requiere evidencia adicional
- Qué información está prohibida para su rol

Así, el agente se convierte en participante de una red de conocimiento, no en un supuesto oráculo.

Confianza calibrada

Lee y See mostraron que la confianza determina cómo las personas dependen de la automatización, especialmente en entornos complejos. La meta no es maximizar la confianza. Es calibrarla.

Si los colaboradores confían demasiado, aparece el sesgo de automatización: se acepta una recomendación incorrecta porque "la IA lo dijo". Si confían demasiado poco, aparece la aversión algorítmica: el sistema se ignora incluso cuando agrega valor. La confianza apropiada exige visibilidad sobre capacidad, límites, evidencia, incertidumbre y desempeño pasado.

Seguridad psicológica

Edmondson encontró que la seguridad psicológica se asocia con conductas de aprendizaje en equipos y que estas median el desempeño. Esto es especialmente relevante para agentes que aprenden de la organización. Si las personas temen revelar errores, cuestionar resultados o documentar cómo se resuelven realmente las excepciones, el agente será entrenado con una versión políticamente correcta pero operacionalmente incompleta de la compañía.

Entrenar agentes requiere una cultura donde sea seguro decir:

- "esta política no funciona en este caso"
- "la respuesta del agente parece convincente, pero está equivocada"
- "el proceso real se desvía del procedimiento"
- "no sabemos todavía cuál es la regla correcta"

La calidad del agente depende de la calidad de la verdad que la organización está dispuesta a mostrarle.

La adopción de agentes exige rediseñar las condiciones de aprendizaje del equipo, no solo capacitar en prompting.



La integración humano-agente es una disciplina de diseño del trabajo, no una capa tecnológica añadida al modelo actual.

4. Entrenar un agente empresarial no es un

Proponemos seis ciclos de entrenamiento iterativos y que se

Ciclo 1. Propósito y frontera de valor

Definir la misión del agente en términos de resultado de negocio, no de funcionalidades. Un agente de cuentas por cobrar no existe para "enviar correos"; existe para mejorar recuperación de caja respetando relaciones comerciales, políticas y riesgo.

Evidencia mínima: KPI, baseline, dueño del proceso, población objetivo, restricciones y criterio de éxito.

Ciclo 2. Contexto y modelo del trabajo

Construir la representación del proceso real: eventos, decisiones, sistemas, roles, datos, handoffs, políticas y excepciones. Incluir casos normales, casos difíciles y contradicciones.

Evidencia mínima: mapa de proceso, catálogo de decisiones, fuentes autorizadas, taxonomía de excepciones y glosario operativo.

Ciclo 3. Demostración del experto

Hacer visible el razonamiento que normalmente permanece implícito. El experto resuelve casos reales explicando señales, hipótesis, descartes y criterios. Este enfoque se alinea con el aprendizaje

cognitivo: modelar, acompañar, dar soporte, articular y reflexionar antes de retirar gradualmente la ayuda.

Evidencia mínima: trazas de decisión, ejemplos positivos y negativos, racionales, contraejemplos y patrones de escalamiento.

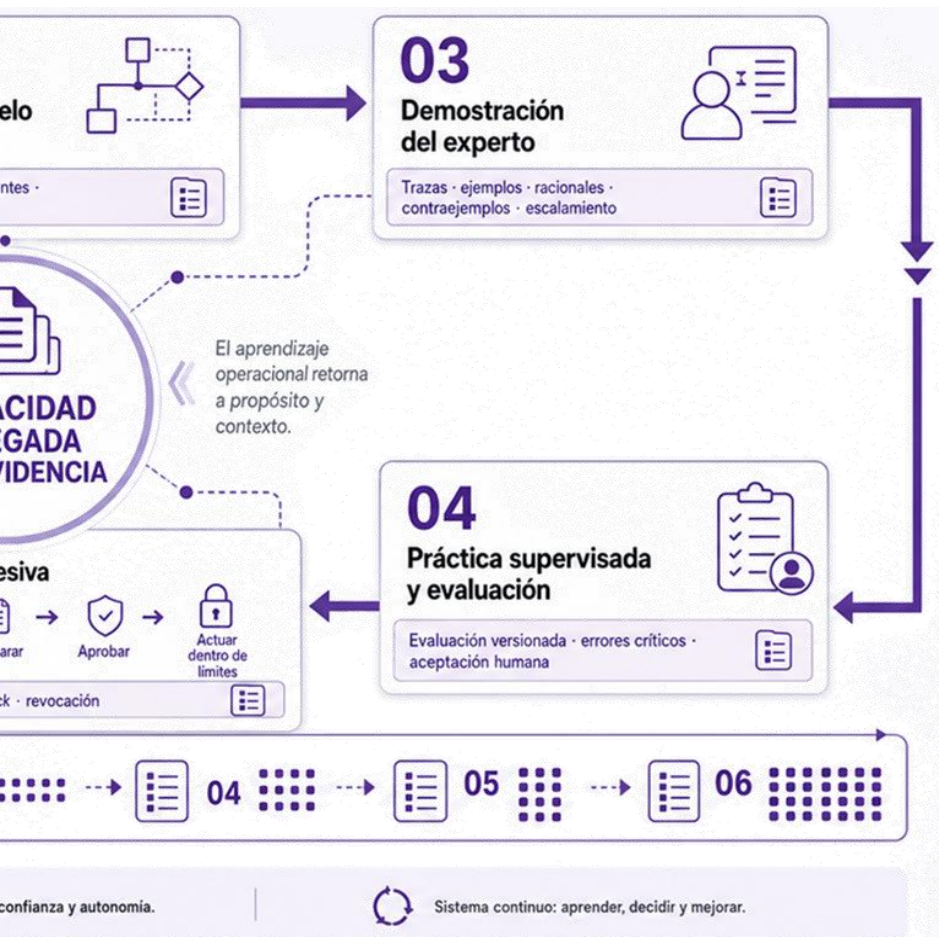


proyecto de desarrollo de software

mantienen activos durante toda la vida del agente.

Ciclo 4. Práctica supervisada y evaluación

Permitir que el agente observe, recomiende y simule antes de actuar. Evaluarlo en escenarios representativos y adversariales, no solo en "prompts" ideales. Medir exactitud, utilidad, política, evidencia, costo y comportamiento ante incertidumbre.



Evidencia mínima: conjunto de evaluación versionado, resultados por tipo de caso, errores críticos, tasa de escalamiento y aceptación humana.

Ciclo 5. Autoridad progresiva

Mover al agente por niveles de autonomía: **1. Observar, 2. Recomendar, 3. Preparar, 4. Validar aprobación, 5. Actuar dentro de límites.** La autorización debe depender de evidencia acumulada, no del entusiasmo por la tecnología.

Evidencia mínima: permisos, límites monetarios o de riesgo, segregación de funciones, trazabilidad, rollback y revocación.

Ciclo 6. Aprendizaje operativo

Cada interacción produce señal: corrección humana, excepción nueva, cambio de política, resultado de negocio o incidente. Esa señal debe alimentar una cadencia formal de revisión.

Evidencia mínima: log de decisiones, feedback clasificado, drift, cambios aprobados, impacto en KPI y backlog de mejora.

El "go-live" no termina el entrenamiento. Lo convierte en una capacidad operativa permanente.

6. Cuatro ejemplos donde el entrenamiento pesa más que la programación

Ejemplo 1. Cuentas por cobrar

Programar un agente que lea facturas, consulte el ERP y redacte recordatorios es relativamente sencillo. Entrenarlo para cobrar bien, exige entender promesas de pago, disputas, jerarquías de cliente, estacionalidad, tono, límites comerciales y cuándo priorizar caja frente a relación.

Valor potencial: más cobertura por analista, menor aging, mejor priorización y mayor consistencia.

Riesgo de mal entrenamiento: escalar un cliente estratégico por una deuda disputada o comprometer una condición comercial inexistente.

Ejemplo 2. Compras y proveedores

El agente puede preparar una orden o comparar cotizaciones. Para actuar con criterio necesita conocer categorías, tolerancias, proveedores críticos, conflictos, intercompany, segregación de funciones y excepciones de continuidad.

Valor potencial: más gasto analizado, ciclos más cortos, menor fuga de valor y mejor detección de anomalías.

Riesgo de mal entrenamiento: optimizar precio e introducir riesgo de abastecimiento, cumplimiento o calidad.

Ejemplo 3. Servicio al cliente

En un estudio de Brynjolfsson, Li y Raymond se sugiere que un asistente puede difundir prácticas de los mejores trabajadores y acelerar la curva de experiencia. El valor no proviene solo del modelo, sino de haber convertido patrones de expertos en orientación utilizable durante la conversación.

Valor potencial: más casos por hora, onboarding más rápido, experiencia más consistente y menor rotación.

Riesgo de mal entrenamiento: resolver rápido la pregunta equivocada, inventar una política o ocultar la incertidumbre.

Ejemplo 4. Fábrica de software

La IA puede producir componentes, pruebas y documentación a alta velocidad. Pero en sistemas maduros el contexto del repositorio, la arquitectura, las convenciones y el costo de revisión dominan el resultado. El estudio de METR ilustra que, sin suficiente ajuste al contexto, el tiempo ahorrado al escribir puede perderse revisando y corrigiendo.

Valor potencial: prototipado, modernización, pruebas, documentación y capacidad de entrega multiplicada.

Riesgo de mal entrenamiento: más volumen de código, más deuda técnica y una falsa sensación de velocidad.

El patrón común de los cuatro casos

El código necesario para consultar, generar, comparar o ejecutar puede producirse con rapidez. El esfuerzo dominante aparece en modelar decisiones, capturar excepciones, preparar evidencia, resolver contradicciones, definir autoridad y evaluar comportamiento. Ese trabajo es organizacional.

Priorizar procesos con valor y repetibilidad, pero también con un owner dispuesto a enseñar el trabajo real y asumir la evolución de la capacidad.

9. Preparando el modelo operativo humano-agente

Un agente aislado es una herramienta. Un sistema agéntico es una forma de operar. Para escalarlo, la empresa necesita una capa explícita que conecte intención, ejecución y control distribuido de forma clara:

Componentes del modelo

Dirección humana. Define propósito, prioridades, apetito de riesgo y criterios de valor. Conserva accountability sobre las decisiones materiales.

Dueños de proceso. Traducen la estrategia a flujos, decisiones, excepciones y KPIs. Son responsables del desempeño end-to-end, no solo de la adopción.

Agentes especializados. Ejecutan, analizan, coordinan o preparan decisiones dentro de una misión y permisos definidos.

Conocimiento operativo. Reúne políticas, datos, memoria de decisiones, taxonomías y conocimiento tácito externalizado.

Runtime y herramientas. Permiten consultar, decidir, actuar, comunicarse y registrar evidencia en los sistemas empresariales.

Control Agéntico. Determina qué puede observar, recomendar, aprobar o ejecutar cada agente, con qué controles y bajo qué condiciones de revocación.

Medición Oportuna. Conectar las actividades agénticas con resultados: valor, calidad, riesgo, tiempo, adopción y capacidad delegada.

La métrica principal no debe ser "número de agentes". Debe ser **capacidad operativa delegada**: cuánto trabajo de valor puede ejecutar el sistema de forma confiable, medible y reversible bajo dirección humana.

El caso de negocio de implementación de IA Agéntica está basado en capturar valor de flujo, no únicamente horas ahorradas o licencias consumidas.

10. Recomendaciones clave para la Alta Dirección

CEO

Definir dónde la capacidad agéntica puede cambiar la economía del negocio, no solo reducir costos. Patrocinar un modelo de responsabilidad claro y proteger el aprendizaje frente a resultados al corto plazo.

COO

Rediseñar flujos inicio-fin, eliminar colas y decidir qué capacidad puede delegarse. Evitar automatizar desperdicio o fragmentación.

CHRO

Convertir expertise en una capacidad institucional. Redefinir roles, incentivos y desarrollo para que enseñar, supervisar y mejorar agentes sea trabajo reconocido. Crear seguridad psicológica para revelar errores y conocimiento tácito.

CIO / CTO

Construir el runtime, las integraciones, la identidad, la observabilidad y la seguridad. Tratar el código generado como un componente del sistema, no como el resultado final.

Riesgo, Legal y Auditoría

Diseñar autoridad, evidencia y reversibilidad desde el inicio. Aplicar una lógica de riesgo proporcional y utilizar marcos como NIST AI RMF para gobernar, mapear, medir y gestionar el riesgo.

11. Como StrategIA puede ayudar a la transformación

Las organizaciones no deberían comenzar preguntando "¿qué agente podemos desplegar?". Deberían preguntar:

¿Qué capacidad operativa queremos delegar, qué debe aprender el agente para ejercerla y qué evidencia necesitamos antes de confiarle autoridad?

Trabajar en cinco decisiones clave:

1. **Entender claramente la unidad de inversión:** de casos de uso aislados a capacidades operativas delegadas con KPI, owner y decisión de escala.
2. **Nombrar dueños de entrenamiento:** responsables de contexto, criterio, evaluación y resultados, no solo de tecnología.
3. **Aplicar perspectiva empresarial:** conectar estrategia, diseño, operación, escala y transformación en una sola ruta de valor.
4. **Aplicar una capa de control:** ampliar permisos solo cuando el desempeño, la reversibilidad y el control lo justifican.
5. **Operar una capacidad de medición de valor:** integrar productividad, calidad, riesgo, aprendizaje y experiencia humana en una misma cadencia.

La empresa que domine esta disciplina podrá convertir experiencia escasa en capacidad reutilizable, ampliar el alcance de cada profesional y operar con una velocidad que el modelo tradicional no puede sostener.

¿Listo para realizar el salto?

Contáctanos.

Fuentes y referencias

- Brynjolfsson, E., Li, D. y Raymond, L. R. (2023). *Generative AI at Work*. NBER Working Paper 31161. DOI: 10.3386/w31161. <https://www.nber.org/papers/w31161>
- Noy, S. y Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654), 187–192. DOI: 10.1126/science.adh2586.
- Dell'Acqua, F., McFowland III, E., Mollick, E., Lifshitz-Assaf, H., Kellogg, K. C., Rajendran, S., Kraymer, L., Candelon, F. y Lakhani, K. R. (2023/2026). *Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of Artificial Intelligence on Knowledge Worker Productivity and Quality*. Harvard Business School Working Paper 24-013 y *Organization Science*. DOI: 10.1287/orsc.2025.21838. <https://www.hbs.edu/faculty/Pages/item.aspx?num=64700>
- Peng, S., Kalliamvakou, E., Cihon, P. y Demirel, M. (2023). *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot*. arXiv:2302.06590; resumen del experimento publicado por GitHub. <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>
- Becker, J. et al. / METR (2025). *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*. <https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/>
- Nonaka, I. (1994). A Dynamic Theory of Organizational Knowledge Creation. *Organization Science*, 5(1), 14–37. DOI: 10.1287/orsc.5.1.14.
- DeChurch, L. A. y Mesmer-Magnus, J. R. (2010). Measuring Shared Team Mental Models: A Meta-Analysis. *Group Dynamics: Theory, Research, and Practice*, 14(1), 1–14. DOI: 10.1037/a0017455.
- Lewis, K. (2003). Measuring Transactive Memory Systems in the Field: Scale Development and Validation. *Journal of Applied Psychology*, 88(4), 587–604. DOI: 10.1037/0021-9010.88.4.587.
- Lee, J. D. y See, K. A. (2004). Trust in Automation: Designing for Appropriate Reliance. *Human Factors*, 46(1), 50–80. DOI: 10.1518/hfes.46.1.50.30392.
- Edmondson, A. C. (1999). Psychological Safety and Learning Behavior in Work Teams. *Administrative Science Quarterly*, 44(2), 350–383. DOI: 10.2307/2666999.
- Collins, A., Brown, J. S. y Newman, S. E. (1989). Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing, and Mathematics. En L. B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*. Lawrence Erlbaum Associates.
- National Institute of Standards and Technology. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. DOI: 10.6028/NIST.AI.100-1. <https://www.nist.gov/itl/ai-risk-management-framework>



StrategIA

Intelligent **A**utonomy for the Enterprise

©2026 STRATEGIA AGILE SOLUTIONS, Inc.

All Rights reserved.

StrategIA is an AI-native consulting firm helping organizations transform how they operate, compete, and create value through emerging technologies. We combine business strategy, operating model design, advanced analytics, intelligent automation, and agentic AI to turn digital transformation into measurable business outcomes.

This publication does not constitute legal, financial, investment, tax, regulatory, or professional advice. The views and examples included are based on publicly available information and StrategIA's interpretation of emerging market trends. Organizations should evaluate their own context, risks, controls, and regulatory obligations before making business or technology decisions.

<https://siaas.ai>

