



AGILE BEDROCK



Scrum Manager[®]
Uncovering better ways

Agile Bedrock

Gestión ágil de equipos del conocimiento con asistencia inteligente

Versión 1.1 (2026)

Scrum Manager ®

Fecha de la versión: septiembre 2026.

Autor de la versión: equipo de Scrum Manager®.

Ilustración portada: Ana Andrés Soria.

Uncovering Better Ways SLU es la editora y propietaria de los derechos de distribución, que libera en los términos de la licencia Creative Commons Atribución – No Comercial 4.0 Internacional (CC BY-NC 4.0). Los formadores y centros oficiales de Scrum Manager quedan licenciados bajo los términos CC BY 4.0 para su actividad formativa.

Derechos registrados en Safe Creative. N° de registro: 2609126968904

Índice de contenidos

Índice de contenidos.....	3
Prefacio	6
Introducción	8
La espiral del conocimiento	8
La agilidad como respuesta a la complejidad.....	9
Gestión predictiva y gestión evolutiva	10
El <i>Manifiesto Ágil</i>	12
Origen de scrum	13
Síntesis del marco scrum tradicional	14
La cuarta ola: la asistencia inteligente como antítesis	15
Qué cuestiona la antítesis y qué consolida	17
Parte I — Fundamentos del empirismo	19
El empirismo: transparencia, inspección, adaptación	19
El valor como norte	20
Personas, interacción y conocimiento tácito.....	21
La asistencia inteligente: espectro y principios de integración.....	22
Parte II — Funciones del equipo ágil	23
La función de visión y decisión de valor	23
La función de ejecución y validación.....	24
La función de facilitación y mejora del sistema	25
La función de orquestación humano-asistencia	26
Parte III — Arquitecturas de ritmo.....	27
Ritmo iterativo.....	27
Ritmo continuo	28
Doble canal.....	29
Transiciones y elección de ritmo.....	30
Parte IV — Prácticas de trabajo.....	31
Inventario de necesidades	31
Tablero de trabajo	32
Incrementos y criterios de terminación	33
Sincronizaciones del equipo.....	34
Estimación y predicción	35

Métricas y radiadores	36
Parte V — Contexto y decisión	37
La asimetría estructural: software y no software	37
Diagnóstico del equipo	38
Elección de arquitectura de ritmo	40
Integración de asistencia: niveles y guardarraíles	41
El campo de posibilidades	42
Parte VI — Principios y valores	43
Principios	43
Valores	45
Apéndice	46
Glosario	46
Referencias bibliográficas	49
Información, recursos y comunidad profesional	50
Mejora continua y control de calidad	50

守破離

Shu Ha Ri

El aprendizaje de cualquier habilidad tiene tres etapas:

Shu: se elige una técnica y, asumiendo que es correcta, se intenta imitar.

Ha: se coleccionan más técnicas.

Ri: se experimenta e inventan nuevas técnicas mezclando, combinando y modificando.

Las técnicas de etapa shu son aplicables en general. Las técnicas de etapa ri solo funcionan en casos concretos, y requieren de conocimiento experto para saber cuándo y cómo aplicarlas.

«No puedes ganar en una industria competitiva utilizando técnicas shu.»

— Alistair Cockburn

Prefacio

Esta guía forma parte del cuerpo de conocimiento troncal de Scrum Manager®. Es contenido que se incorpora al marco de formación y certificación para cubrir el vacío que abre la evolución actual: la irrupción de la asistencia inteligente modifica la ecuación de producción de valor en las empresas del conocimiento y, con ella, los criterios con que un profesional decide cómo trabajar.

La velocidad a la que gira hoy la espiral del conocimiento (la sucesión de tesis, antítesis y síntesis que veremos en la introducción) exige conocer la base de referencia desde la que cada profesional puede componer su propia síntesis. Eso es lo que ofrece esta guía. No es un modelo cerrado, sujeto a la obsolescencia de cada giro, sino los fundamentos y los criterios profesionales que permiten adaptarse a medida que el contexto cambia: el sustrato firme (el *bedrock*) sobre el que cada profesional construye su desarrollo y su evolución.

El documento cumple una doble función. Sirve como **guía** para los equipos y organizaciones que necesitan diseñar su forma de trabajar en este escenario, partiendo del empirismo y adaptándose a su contexto, ya desarrollen *software* o no, ya empleen asistencia generativa o agentes autónomos, ya necesiten un ritmo iterativo o continuo. Y es, a la vez, el **temario de referencia** de la certificación troncal Agile Bedrock de Scrum Manager, para los profesionales que deseen acreditar su conocimiento actualizado.



[Certificación Scrum Manager](#)

La guía no prescribe un marco único. Fiel a la postura agnóstica de Scrum Manager®, ofrece un inventario de conocimiento para que cada profesional desarrolle su propio criterio. Se dirige a quienes ejercen, o aspiran a ejercer, la función de facilitación y mejora en equipos del conocimiento; el nombre de esa función importa menos que su propósito, que es lograr que el equipo entregue valor de forma empírica, sostenible y transparente. Los recursos de Scrum Manager para seguir formándose después de la guía se recogen en el apéndice.

Al hablar de agilidad se manejan anglicismos y términos que adquieren un significado concreto dentro de la gestión. La guía sigue tres convenciones:

- Las palabras inglesas van en cursiva la primera vez que aparecen y en redonda a partir de entonces.

- Los conceptos que cuentan con una traducción asentada se emplean traducidos, con el término original indicado entre paréntesis la primera vez.
- Los nombres de los conceptos del oficio, como *scrum*, *sprint* o *product owner*, se escriben en minúscula; conservan sus mayúsculas los nombres propios y la Definición de Hecho (*Definition of Done*, DoD).

Introducción

Esta introducción sirve para saber de dónde venimos. Presenta la espiral del conocimiento que da sentido a toda la guía, repasa el origen de la gestión ágil y del marco scrum que llegó a ser su estándar, y describe la antítesis que hoy lo pone a prueba: la asistencia inteligente. El resto de la guía sirve para comprender hacia dónde se está configurando la próxima tesis.

La espiral del conocimiento

Conviene empezar por el marco que da sentido a todo lo que sigue. El conocimiento no avanza en línea recta ni por acumulación, sino describiendo una espiral. Nonaka y Takeuchi, los investigadores que dieron nombre a scrum, lo formularon apoyándose en una idea dialéctica: el conocimiento progresa mediante tesis, antítesis y síntesis.

Una **tesis** es una respuesta válida a una situación, construida con el conocimiento y las circunstancias de su momento. Funciona, se extiende y se convierte en el modo establecido de hacer las cosas. Con el tiempo aparecen circunstancias nuevas que ponen en evidencia sus límites y contradicciones. Esa es la **antítesis**, que más que un error de la tesis es la señal de que el contexto ha cambiado. Del enfrentamiento entre ambas surge una **síntesis**, una comprensión nueva que reconcilia lo que seguía siendo válido de la tesis con lo que la antítesis ha revelado. La síntesis no da la victoria a una de las dos ni se limita a sumarlas.

Es en este momento que la tesis y antítesis anteriores se reconcilian y trascienden. No obstante, con el tiempo, incluso la síntesis resultará tendenciosa en algún aspecto. Servirá entonces como tesis para un nuevo movimiento dialéctico, y así el proceso continúa de forma zigzagueante en espiral.

— Nonaka, 2004

Esa es la clave. La síntesis, lejos de ser un punto de llegada, es el inicio del siguiente giro: cada síntesis se convierte en la tesis de un nuevo ciclo, que encontrará su propia antítesis y dará lugar a una nueva síntesis. El conocimiento avanza así, en espiral, sin detenerse.

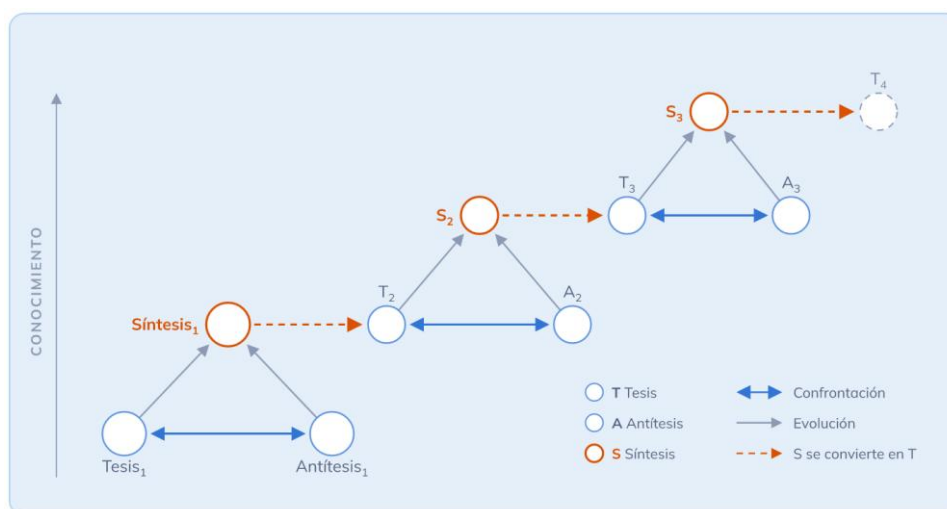


Figura 1. La espiral del conocimiento de Nonaka y Takeuchi: la síntesis reconcilia tesis y antítesis y se convierte en la tesis del siguiente giro.

Esta guía se sitúa en un punto concreto de esa espiral. La gestión ágil fue, en su día, una síntesis. La asistencia inteligente es la antítesis que hoy la pone a prueba. Y lo que la guía recoge es **la síntesis que se está formando ahora mismo**, la reconciliación, todavía en curso, entre el cuerpo de conocimiento ágil que sigue siendo válido y las circunstancias nuevas que la inteligencia artificial ha introducido. Quien la estudie debe leerla con esa conciencia: aprende el estado actual de una evolución que continuará, y no un destino final.

La agilidad como respuesta a la complejidad

La gestión ágil surgió como antítesis de un modelo anterior, al que nos referiremos con frecuencia: la gestión de proyectos predictiva. Ambas tienen sus virtudes y resultan más útiles en industrias distintas. La predictiva se centra en planificar, calcular un presupuesto y marcar plazos. Si el proyecto se termina en la fecha acordada, sin exceder el coste y con todas las funcionalidades del plan inicial, se considera un éxito.

Esa estrategia sirve en entornos estables, con productos que son resultado de una atención escrupulosa a procesos y protocolos. La **gestión predictiva** es fruto de la Revolución Industrial. Viene de la construcción, de la automoción, de la fábrica. Si lo que el cliente busca es una casa, esta debe construirse sólida, segura, conforme a las necesidades de quienes la habitarán y, en el caso ideal, dentro de plazo y presupuesto.

Pero buena parte de lo que se produce hoy no se parece a una casa, porque puede ser abstracto: una película, una aplicación móvil, una campaña de *marketing*, un informe jurídico. Con esos productos se pueden probar cosas durante el desarrollo, comprobar de forma empírica qué funciona, ajustar sobre la marcha, partir de un esbozo mínimo e ir creciendo. El escenario cambia, y una funcionalidad que parecía esencial al principio puede quedar obsoleta para la fecha de entrega.

Para competir en estas industrias se necesita capacidad de responder rápido en escenarios inciertos, sin requisitos estables de partida, con clientes que necesitan empezar a usar el producto cuanto antes y mejorarlo de forma continua. Son las **empresas del conocimiento**, las que desarrollan productos o servicios basándose en el conocimiento más que en las herramientas y los procesos. Su entorno se parece muy poco al que originó la gestión predictiva.

Gestión predictiva y gestión evolutiva

Desde los años ochenta se han acumulado tantos modelos y prácticas para gestionar proyectos que el panorama puede resultar abrumador. Conviene un mapa que los ordene. Toda la gestión de proyectos puede entenderse cruzando tres conceptos (desarrollo, trabajo y conocimiento) con dos modelos de gestión, la gestión predictiva y la evolutiva.

Desarrollo

El desarrollo del proyecto puede ser completo o incremental.

- **En el desarrollo completo**, la descripción de lo que se desea obtener está disponible y detallada al inicio y sirve de base para estimar. Con el plan inicial se planifican la agenda y los recursos, y durante la ejecución se gestiona el cumplimiento de lo previsto.
- **En los desarrollos incrementales**, esa descripción está incompleta al inicio y se complementa y evoluciona durante el desarrollo. Admite dos tácticas diferentes: el **incremento continuo**, que logra un flujo continuo de entrega según se completan las partes, y el **incremento iterativo**, que mantiene la producción a un ritmo fijo mediante tiempo prefijado (*timeboxing*), y que es el modo del marco estándar de scrum, donde cada iteración se llama sprint.

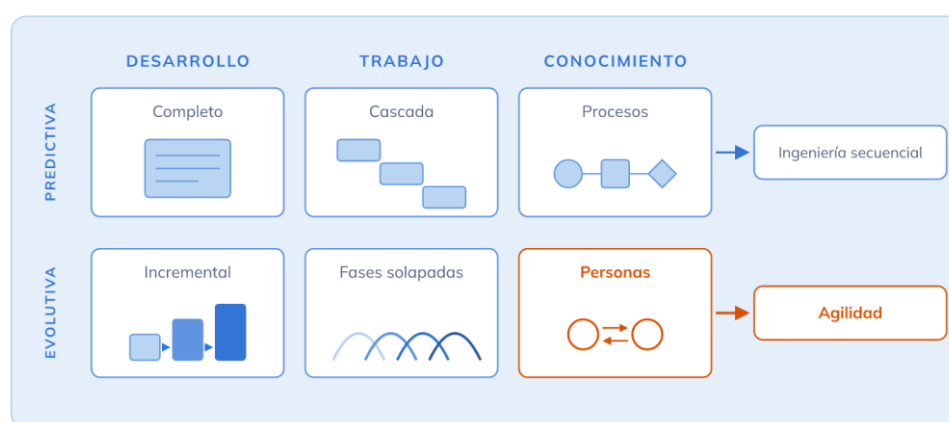


Figura 2. El mapa de la gestión de proyectos: el acento marca el extremo ágil, con el conocimiento residiendo en las personas.

Conviene retener esta doble táctica del desarrollo incremental, iterativo y continuo, porque será una de las bisagras de toda la guía.

Trabajo

La forma de trabajar puede ser secuencial («en cascada»), encadenando fases una tras otra, o concurrente, solapándolas en el tiempo.

Conocimiento

El conocimiento puede residir en los procesos (conocimiento explícito, donde la calidad depende del proceso y la tecnología) o en las personas (conocimiento tácito, donde la calidad depende de la experiencia, la motivación y el talento de quien trabaja).

El conocimiento tácito es personal, específico del contexto, y por tanto difícil de formalizar y comunicar. El conocimiento explícito o «codificado», por otro lado, es conocimiento que puede transmitirse con lenguaje formal y sistemático.

— Nonaka, 1995

Los dos modelos de gestión

Ingeniería secuencial

La combinación de desarrollo completo, trabajo en cascada y conocimiento en los procesos da lugar a la ingeniería secuencial, también llamada gestión predictiva. Su objetivo es ofrecer resultados predecibles: desarrollar el producto previsto sin exceder plazo ni recursos.

Gestión evolutiva

La combinación de desarrollo incremental, fases solapadas y conocimiento en las personas da lugar a la gestión evolutiva, cuyo objetivo es entregar pronto un producto mínimo viable e incrementar su valor de forma continua. Puede llevarse a cabo con producción basada en procesos (**ingeniería concurrente**) o con producción basada en personas (**agilidad**). La distinción importa, porque sin ella se confunde con agilidad la simple aplicación de las reglas de scrum o el mero uso de un tablero *kanban*.

Ingeniería concurrente	Agilidad
Emplea recursos propios de la gestión ágil: solapamiento de fases, equipos multidisciplinares e iteraciones frecuentes de mejora.	Reduce o elimina tareas administrativo-burocráticas que no aportan valor al producto o al sistema de desarrollo.
Se centra en la calidad de los procesos.	Propia de las empresas del conocimiento. Se centra en el conocimiento tácito de las personas, en la cultura y el talento.

Quien se centra en seguir el proceso, aunque use iteraciones y tableros, hace ingeniería concurrente; la agilidad exige confianza en las capacidades del equipo y autogestión real. Esta diferencia, en apariencia sutil, será central en toda la guía.

El Manifiesto Ágil

Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar: a los individuos y su interacción, por encima de los procesos y las herramientas; el software que funciona, por encima de la documentación exhaustiva; la colaboración con el cliente, por encima de la negociación contractual; la respuesta al cambio, por encima del seguimiento de un plan. Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.

— *Manifiesto por el Desarrollo Ágil de Software, 2001*

En febrero de 2001, diecisiete profesionales del software se reunieron en Salt Lake City. Tenían algo en común: eran críticos con los modelos de producción basados en procesos, que consideraban demasiado pesados y rígidos. De allí salieron el término «métodos ágiles» y los cuatro postulados del *Manifiesto*, junto con doce principios que los desarrollan.

Los individuos y su interacción, por encima de los procesos y las herramientas. Es el postulado más importante. Los procesos ayudan y las buenas herramientas mejoran la eficiencia, pero hay tareas que requieren talento y personas motivadas que lo aporten. En una producción basada en procesos se busca que la calidad sea consecuencia de estos; en el desarrollo ágil, los procesos son solo un soporte para guiar el trabajo.

El producto que funciona, por encima de la documentación exhaustiva. La documentación necesaria es útil, pero su relevancia debe ser menor que la del producto. Un documento de requisitos cerrado y minucioso antes de empezar suele ser una pérdida de tiempo.

La colaboración con el cliente, por encima de la negociación contractual. El objetivo de un proyecto ágil no es controlar la ejecución para que se cumpla el plan inicial, sino proporcionar de forma continua el mayor valor posible.

La respuesta al cambio, por encima del seguimiento de un plan. Los valores de la gestión ágil son la anticipación y la adaptación, frente a la planificación y el control de la gestión ortodoxa.

Los doce principios que acompañan a estos postulados serán la piedra de toque cuando lleguemos a la antítesis. Conviene retener en particular estos:

- la entrega temprana y continua de valor;
- dar la bienvenida al cambio;
- entregar con frecuencia, en periodos breves;
- construir en torno a individuos motivados;
- la conversación directa como forma más eficaz de comunicación;
- el software que funciona como medida del progreso;
- el ritmo sostenible;

- la excelencia técnica y la simplicidad;
- los equipos que se autoorganizan;
- la reflexión periódica para ajustar la conducta.

Origen de scrum

Scrum es un modelo de desarrollo ágil que se caracteriza por cuatro rasgos:

- equipos autónomos y autogestionados, que comparten conocimiento y aprenden juntos;
- una estrategia de desarrollo incremental, en lugar de la planificación completa;
- la calidad basada en el conocimiento tácito y la creatividad de las personas, no en los procesos;
- el solapamiento de las fases del desarrollo, en lugar de encadenarlas en cascada.

La palabra viene del rugby, donde «scrum» designa la formación en la que ambos equipos empujan juntos por el balón. El término saltó a la gestión en el Japón de los años ochenta, cuando Nonaka y Takeuchi estudiaron las empresas de manufactura que obtenían los mejores resultados de innovación y tiempo de salida al mercado (Takeuchi y Nonaka, 1986). Lo que observaron fue una forma de trabajar en equipos autogestionados, con las fases solapadas y el conocimiento compartido, que compararon con el avance de los jugadores de rugby en formación.

A partir de 1995, Ken Schwaber presentó en la conferencia OOPSLA una metodología de desarrollo de software basada en un entorno scrum (Schwaber, 1995). No existe una autoridad que determine qué es scrum y qué no: el marco ha cambiado y seguirá cambiando con las aportaciones de la comunidad. El espíritu original se mantiene. Las prácticas deben ayudar a los equipos a autogestionarse y a mantener un flujo de avance continuo, produciendo resultados de forma iterativa y frecuente. Scrum Manager® usa el término en ese sentido original, y no como un reglamento cerrado de ceremonias.

Síntesis del marco scrum tradicional

Durante más de dos décadas, el marco scrum se estableció como el estándar de referencia del desarrollo ágil iterativo. Fue, en términos de la espiral, una síntesis lograda: reconcilió la necesidad de entregar pronto con la de mantener el control empírico del avance. Conviene conocerlo, no para imitarlo ciegamente, sino para entender qué problemas resolvía y cuáles de ellos siguen vigentes ahora que esa síntesis se ha convertido en la tesis que la asistencia inteligente pone a prueba.

Roles tradicionales

- El product owner o propietario del producto, única persona responsable de maximizar el valor.
- El *development team* o equipo de desarrollo, los profesionales autogestionados y multifuncionales que entregan el incremento.
- El *scrum master*, facilitador del marco, responsable de que scrum se entienda y aplique correctamente.

Artefactos tradicionales

- La pila del producto (*product backlog*), lista priorizada y emergente de todo lo que el producto necesita.
- La pila del sprint (*sprint backlog*), el subconjunto seleccionado para un sprint junto con el plan para entregarlo.
- El incremento, la suma de los elementos terminados sobre el valor de los anteriores.

Eventos tradicionales

- El sprint, iteración de duración fija que contiene a los demás eventos.
- La planificación, que decide qué se hará y cómo.
- El scrum diario, sincronización breve.
- La revisión, que inspecciona el incremento.
- La retrospectiva, que reflexiona sobre el proceso.

Este marco funcionó, y sigue funcionando, en miles de equipos. Pero se diseñó en un contexto donde las personas éramos los únicos actores en la producción de valor. La asistencia inteligente obliga a preguntarse si estos roles, artefactos y eventos siguen siendo la forma más eficiente de empirismo, o si en algunos contextos se han convertido en un ritual que simula control mientras la realidad fluye por otros cauces.

La cuarta ola: la asistencia inteligente como antítesis

Desde 2022, la irrupción de la inteligencia artificial generativa y, desde 2024, el auge de los agentes de asistencia autónomos han alterado la velocidad relativa entre la decisión humana y la ejecución asistida. Más que una moda tecnológica pasajera, es un cambio en la ecuación de producción de valor que obliga a releer el marco ágil desde sus raíces. En los términos de la espiral del conocimiento, es la antítesis que pone a prueba la tesis que scrum había llegado a ser.

La pregunta ya no es si la asistencia inteligente va a cambiar el trabajo, porque ya lo está haciendo. La pregunta es si ese cambio erosionará el empirismo (la capacidad de inspeccionar y adaptar a partir de la realidad) o si la asistencia se usará para potenciarlo.

En esta guía la asistencia inteligente se entiende como un **espectro**, no como un interruptor. En un extremo está la **asistencia generativa**, con herramientas que proponen borradores, resúmenes, sugerencias de código o de contenido, mientras la persona decide, edita y valida. En el otro, los **agentes autónomos**, sistemas que ejecutan tareas completas, desde escribir código hasta redactar documentos, con mínima intervención humana en la ejecución. Un equipo de marketing que genera borradores de campaña está en un polo; un equipo de software que delega la escritura de microservicios a un agente está en el otro. El marco de trabajo debe poder contener ambos sin perder su esencia.

Esta antítesis tiene un efecto que conviene comprender antes de seguir, porque ordena todo el análisis posterior: la asistencia inteligente actúa como un **amplificador**.

La asistencia amplifica, no corrige

La inteligencia artificial no es, por sí sola, ni la solución ni la ruina de la gestión ágil. Lo que hace es magnificar las condiciones que ya existen en el equipo. Donde hay buenas prácticas, cultura de calidad y fundamentos sólidos, la asistencia refuerza el flujo de trabajo y acelera la entrega de valor. Donde hay procesos frágiles, deuda técnica y coordinación rota, la asistencia expone y agrava esos problemas, que se manifiestan ahora con más fuerza y más deprisa. La misma herramienta produce resultados opuestos según el estado de partida del equipo.

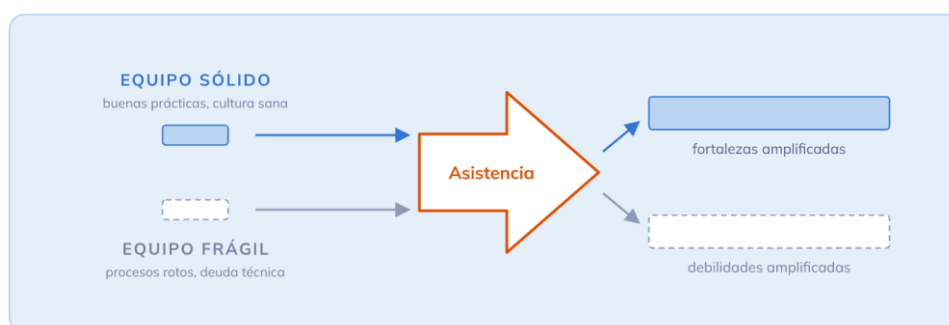


Figura 3. La asistencia amplifica las condiciones de partida: potencia a los equipos sólidos y agrava las debilidades de los frágiles.

El mecanismo es comprensible. La asistencia permite generar trabajo mucho más rápido, lo que tiende a producir lotes mayores, más difíciles de revisar y más propensos a introducir defectos, salvo que existan sistemas de control que contengan ese volumen (pruebas automatizadas, revisión disciplinada, ciclos de *feedback* rápidos). Sin esa base, la velocidad acumula problemas a mayor ritmo en lugar de mejorar el resultado. La asistencia, en suma, no arregla equipos; los amplifica.

Esta constatación tiene una lectura tranquilizadora respecto a los fundamentos, y es la razón por la que esta guía los coloca en su primera parte. Si la asistencia amplifica lo que ya existe, los mecanismos que permiten ver pronto qué está pasando y corregir a tiempo (la transparencia, la inspección frecuente, la adaptación) ganan valor con la inteligencia artificial en lugar de perderlo. El empirismo, lejos de ser una víctima de la antítesis, es precisamente lo que distingue a un equipo que aprovecha la amplificación de otro que la sufre.

El cuello de botella se desplaza

Hay un segundo efecto, más conceptual, que explica por qué la asistencia transforma los equipos. En todo trabajo del conocimiento conviven dos cosas distintas: decidir qué hay que hacer y ejecutar lo decidido. Durante toda la historia de la gestión ágil, el cuello de botella estuvo casi siempre en la ejecución. Escribir el código, redactar el documento o producir la pieza costaba tiempo y esfuerzo humano, y por eso el trabajo se organizaba en torno a ese esfuerzo, se estimaba y se medía.

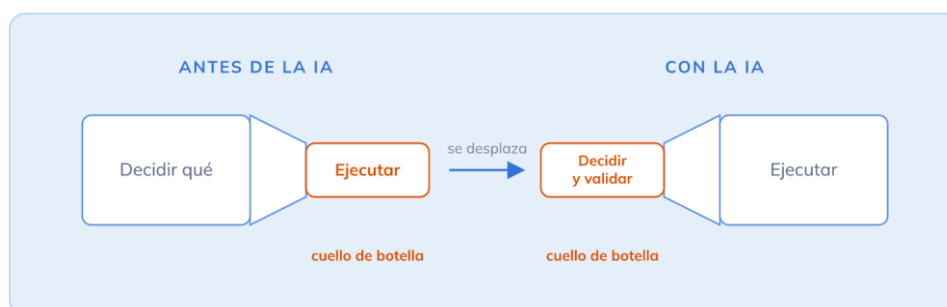


Figura 4. El desplazamiento del cuello de botella: al abaratare la ejecución, el punto estrecho del trabajo pasa de producir a decidir y validar.

Este desplazamiento explica una transformación ya visible en los equipos que más han incorporado asistencia: tienden a hacerse más pequeños. Donde antes hacía falta un equipo numeroso con roles especializados, empieza a bastar un grupo reducido capaz de poseer la visión, el diseño, la construcción y la validación, apoyado en herramientas que ejecutan la parte mecánica. Y como en el trabajo del conocimiento el freno real ha estado siempre en la velocidad de la decisión más que en el número de manos, los equipos más pequeños deciden antes y con menos burocracia.

La consecuencia para el marco es incómoda pero clara. La maquinaria que existía para coordinar la ejecución de muchas personas pierde peso, mientras que todo lo dedicado a decidir bien y a validar lo gana.

Qué cuestiona la antítesis y qué consolida

Antes de entrar en la síntesis conviene separar con claridad sobre qué actúa la antítesis, porque la asistencia inteligente no afecta por igual a todo el marco. Cuestiona ciertas formas concretas de trabajar, sobre todo en entornos de flujo, y a la vez consolida (cuando no amplifica) los fundamentos que dan sentido a la agilidad. Esta separación es, en sí misma, el primer paso de la síntesis.

Lo que la antítesis cuestiona son formas concretas, no principios, y lo hace con más fuerza en los entornos que se inclinan hacia el flujo continuo que en los iterativos, de manera muy marcada en el carril de asistencia del desarrollo de software. Cuatro formas dejan de ser evidentes:

- La iteración de duración fija como opción única, cuando una parte del trabajo se produce en minutos.
- La batería completa de ceremonias celebradas siempre del mismo modo, cuando el equipo es diminuto o cuando una parte del trabajo la ejecutan agentes.
- El punto de historia como unidad universal de medida, que no encaja para medir el trabajo de una máquina.
- El equipo grande con roles muy especializados, que se vuelve innecesario allí donde la ejecución se abarata.

Ninguna de estas formas desaparece; siguen siendo válidas en muchos contextos, sobre todo fuera del software. Pero ninguna puede darse ya por supuesta para todos.

Lo que la antítesis consolida, y a menudo amplifica, es todo lo que siempre ha sido fundamento antes que ceremonia, y que la síntesis debe seguir preservando:

- la entrega temprana y continua de valor;
- la transparencia sobre el estado real del trabajo;
- la inspección frecuente del resultado y de la forma de trabajar;
- la adaptación rápida ante lo que se descubre;
- la centralidad del juicio humano, que decide qué es valioso, valida la calidad y asume la responsabilidad.

Por encima de todo, la antítesis consolida el primer postulado del *Manifiesto*, que la asistencia vuelve más literal que nunca en lugar de derogarlo: los individuos y su interacción por encima de los procesos y las herramientas. La asistencia es la herramienta más poderosa que ha entrado nunca en el campo, y por eso mismo recuerda que una herramienta, por sí sola, no aporta conocimiento tácito ni asume responsabilidad.

La síntesis que desarrolla esta guía consiste, precisamente, en reconstruir la forma de trabajar sobre lo que se consolida, soltando sin nostalgia lo que la antítesis ha dejado obsoleto en cada contexto. Las dos primeras partes establecen los fundamentos que se sostienen; las partes III a V reconstruyen, sobre ellos, las arquitecturas, las prácticas y los criterios de decisión.

Parte I — Fundamentos del empirismo

Las raíces que no cambian, aunque las ramas muten

Las dos primeras partes de esta guía desarrollan lo que la antítesis consolida: los fundamentos sobre los que cualquier síntesis tiene que apoyarse. Son anteriores a cualquier ritmo, práctica o herramienta, y sobreviven a todos ellos.

El empirismo: transparencia, inspección, adaptación

El empirismo es la base innegociable de cualquier desarrollo ágil. No es una técnica, sino una actitud ante la realidad. Afirma que el conocimiento válido proviene de la experiencia y de la observación de lo que sucede, y no de la planificación teórica.

Scrum se describe a sí mismo como un marco empírico porque se apoya en tres pilares:

- **Transparencia.** Los aspectos significativos del trabajo deben ser visibles para quienes responden del resultado. La visibilidad alcanza a lo que se está haciendo, a los problemas que surgen y a las decisiones que se toman. Sin transparencia no hay inspección posible.
- **Inspección.** Quienes participan en el trabajo deben inspeccionar con frecuencia los artefactos y el avance hacia el objetivo, para detectar desviaciones indeseadas. La inspección es una mirada crítica y constructiva, lejos de cualquier control policial.
- **Adaptación.** Si se detecta que algo se desvía de los límites aceptables, debe ajustarse cuanto antes para minimizar desviaciones mayores.

Estos tres pilares no son negociables, y son indiferentes al ritmo de trabajo. Da igual que el equipo trabaje en iteraciones de dos semanas, en flujo continuo o con agentes que despliegan varias veces al día. Si no hay transparencia sobre qué se construye, inspección frecuente de si eso genera valor y adaptación rápida cuando no lo genera, no hay agilidad. Hay caos veloz o burocracia lenta.

La amplificación que introduce la asistencia inteligente hace estos pilares más necesarios todavía, porque cuanto más rápido se produce, antes hay que poder ver si lo producido sirve.

El valor como norte

El objetivo de la gestión ágil no es entregar más rápido, sino entregar **más valor, más temprano**. La velocidad es un medio; el valor, el fin. Esta distinción, que siempre fue importante, se vuelve crítica en un entorno donde la asistencia puede multiplicar la producción, porque producir mucho no equivale a aportar valor y a veces solo añade volumen, o incluso ruido, al producto.

El valor significa cosas distintas en contextos distintos:

- para un cliente de software, una funcionalidad que resuelve un problema real;
- para un departamento legal, un contrato revisado que reduce el riesgo sin demorar la operación;
- para un equipo de marketing, una campaña que conecta con el público en el momento adecuado.

La gestión predictiva mide el éxito por el cumplimiento del plan (si se llegó a tiempo, si no se excedió el presupuesto, si se entregó lo acordado). La gestión ágil lo mide por el impacto: si algo cambió para mejor, si se aprendió algo que no se sabía, si el equipo es capaz de adaptarse. Mantener el valor como norte es lo que impide que la velocidad se convierta en un fin en sí mismo.

Personas, interacción y conocimiento tácito

El *Manifiesto* coloca a los individuos y su interacción por encima de los procesos y las herramientas. No es un romanticismo, sino una observación económica. En el trabajo del conocimiento, la calidad del resultado depende más del conocimiento tácito de las personas que de la calidad de los procesos.

Un proceso puede indicar cómo redactar un contrato, pero solo una persona con experiencia jurídica y conocimiento del cliente sabe si ese contrato protege de verdad los intereses del negocio. Un agente de asistencia puede generar el borrador en segundos; la decisión de si ese borrador es adecuado sigue siendo humana. Esta es la línea que la asistencia no cruza: ejecuta y propone, pero no aporta conocimiento tácito ni asume responsabilidad.

Las interacciones directas siguen siendo la forma más rica de comunicar información compleja, por sus matices, su tono y su retroalimentación inmediata. Con equipos distribuidos y asistencia digital no siempre son posibles, pero el principio permanece: buscar la interacción más rica que el contexto permita.

Y los equipos autogestionados siguen siendo el corazón de la agilidad. Necesitan dirección, como cualquier equipo, pero esa dirección debe venir del conocimiento del trabajo y no de la jerarquía. Cuando quien no hace el trabajo decide cómo debe hacerse, se pierde la capacidad de adaptación local.

La asistencia inteligente: espectro y principios de integración

La asistencia inteligente no es una herramienta más, como lo fueron en su día las hojas de cálculo o los tableros digitales. Aquellas ayudaban a gestionar el trabajo; la asistencia puede realizar parte del trabajo. Por eso conviene integrarla con criterio, entendiéndola como el espectro que vimos en la introducción (—«La cuarta ola: la asistencia inteligente como antítesis»), que va de la asistencia generativa, que propone para que la persona decida, a los agentes autónomos, que ejecutan tareas completas bajo supervisión.

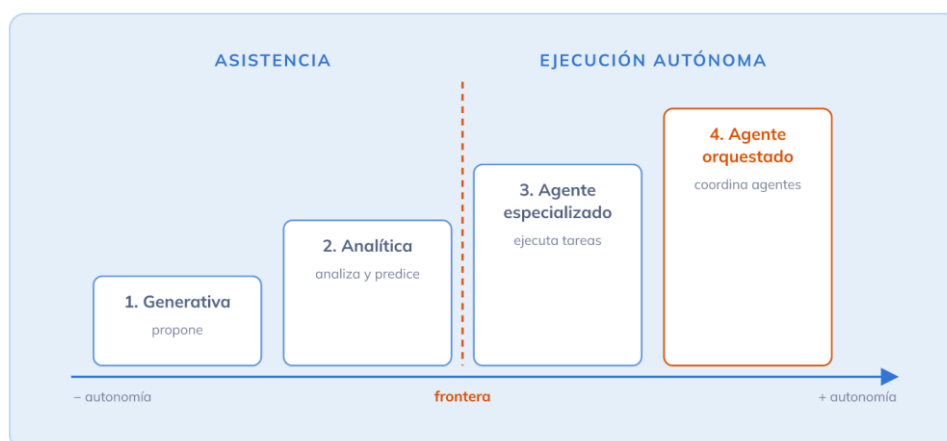


Figura 5. El espectro de la asistencia inteligente: la frontera entre el segundo y el tercer nivel, de la asistencia a la ejecución autónoma, es la que más condiciona el modo de trabajo.

Sea cual sea el punto del espectro, su integración en un equipo ágil se rige por cuatro principios:

- **El empirismo manda sobre la velocidad.** La asistencia se adopta si refuerza la capacidad de inspeccionar y adaptar, no porque acelere. Velocidad sin empirismo es solo caos más rápido.
- **La responsabilidad es indivisible.** Si un agente genera un trabajo que falla, la responsabilidad es del equipo humano que lo validó. La asistencia no puede ser el chivo expiatorio.
- **La transparencia se extiende a la asistencia.** El equipo debe saber qué decisiones y qué producciones fueron humanas y cuáles asistidas. Si no se puede inspeccionar la procedencia de una decisión, no hay empirismo sobre ella.
- **La interacción humana sigue siendo el eje.** La asistencia no sustituye la conversación entre quien define el valor y quien lo construye; sustituye la transcripción, la repetición y la ejecución mecánica.

Parte II — Funciones del equipo ágil

Quién hace qué, y por qué

En la era de la asistencia inteligente, los títulos importan menos que las funciones. Un equipo ágil necesita cubrir cuatro funciones esenciales, que pueden ejercer una o varias personas y que en algunos casos pueden estar asistidas por sistemas autónomos. La responsabilidad final de cada función, en todo caso, es humana.

El desplazamiento del cuello de botella hacia la decisión, descrito en la introducción (—«La cuarta ola: la asistencia inteligente como antítesis»), recorre toda esta parte. Las funciones que deciden y validan ganan peso, las que ejecutaban lo pierden, y aparece una función nueva que el marco tradicional no contemplaba.

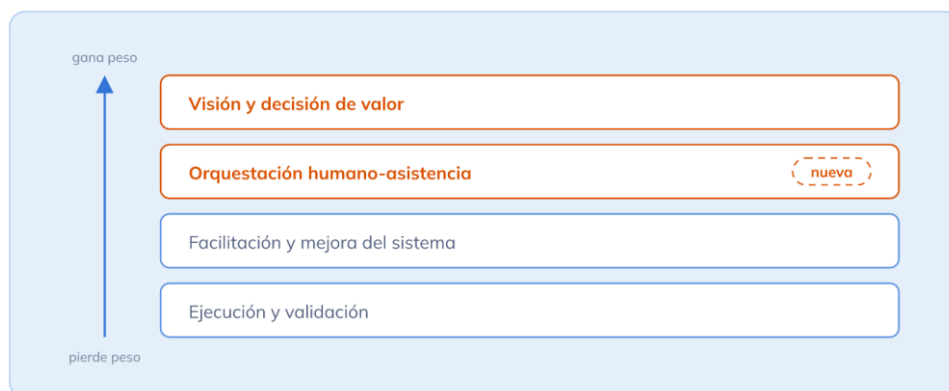


Figura 6. Las cuatro funciones del equipo ágil: destacadas, las que ganan peso con el desplazamiento del cuello de botella, incluida la función de orquestación, que es nueva.

La función de visión y decisión de valor

Propósito

Definir qué problema se resuelve, para quién y por qué importa. Maximizar el valor entregado al cliente y al negocio.

Es la función que tradicionalmente ejercía el product owner. Pero no es un título, sino una responsabilidad. En un equipo legal puede recaer en el abogado *senior* que entiende el riesgo del negocio; en marketing, en quien conoce al cliente; en software, en quien entiende el mercado.

Responsabilidades clave

- Mantener un inventario de necesidades priorizado, transparente y comprensible para todo el equipo.
- Tomar decisiones de priorización basadas en valor, riesgo y dependencias.
- Comunicar la visión de modo que el equipo pueda tomar decisiones tácticas alineadas.

- Validar que lo entregado resuelve el problema planteado, más allá de que simplemente «funcione».

Con asistencia inteligente

La asistencia puede analizar datos de uso, feedback de clientes y tendencias de mercado para informar la priorización, y generar borradores de historias o de hoja de ruta. Pero la decisión de qué problema resolver sigue siendo humana, porque implica juicio de negocio, tolerancia al riesgo y visión estratégica que la máquina no posee. Esta es la función que más se refuerza con el desplazamiento del cuello de botella, porque cuando ejecutar se abarata, decidir bien qué construir se convierte en la actividad más valiosa y más escasa. El perfil evoluciona de gestor de una lista a estrategia del valor.

La función de ejecución y validación

Propósito

Construir el incremento de producto o servicio y asegurar que cumple los criterios de calidad y de terminación.

Es la función que tradicionalmente ejercía el equipo de desarrollo. Pero «desarrollar» no es solo programar, sino crear, y eso abarca redactar, diseñar, analizar, codificar, probar y verificar.

Responsabilidades clave

- Seleccionar del inventario de necesidades el trabajo abordable en el siguiente ciclo o tramo de flujo.
- Descomponer, estimar y autoasignar el trabajo.
- Construir el incremento con la calidad acordada.
- Validar que el incremento cumple los criterios de terminación antes de entregarlo.

Con asistencia inteligente

En software, los agentes pueden generar código, pruebas, documentación o infraestructura, y las personas pasan a ser **curadores y validadores** que revisan, comprenden, integran y garantizan. En los equipos no software, la asistencia acelera la producción de borradores, análisis o diseños, y las personas validan el juicio profesional, además de la forma. La autoasignación y la estimación siguen siendo humanas, porque dependen del conocimiento tácito del equipo sobre su propia capacidad y contexto. El conocimiento técnico, lejos de sobrar, se vuelve más necesario: hace falta más juicio que nunca para validar lo que una máquina produce deprisa.

La función de facilitación y mejora del sistema

Propósito

Lograr que el equipo trabaje sin fricciones, remover impedimentos y mejorar de forma continua el sistema de trabajo.

Es la función que tradicionalmente ejercía el scrum master. Pero no es un guardián de ritos, sino un facilitador del empirismo y un optimizador del flujo de valor.

Responsabilidades clave

- Facilitar las sincronizaciones del equipo para que sean eficientes y productivas.
- Identificar y remover los impedimentos que frenan el flujo de valor.
- Proteger al equipo de las interrupciones externas que rompen el ritmo.
- Fomentar una cultura de mejora continua, donde se inspecciona y adapta el sistema de trabajo.
- Enseñar principios ágiles más que imponer prácticas.

Con asistencia inteligente

La asistencia puede automatizar la recogida de métricas, la generación de resúmenes de reuniones o la detección de patrones de bloqueo. Pero la facilitación de conflictos, el acompañamiento de las personas y la decisión de cambiar el sistema de trabajo siguen siendo humanas, porque requieren empatía, intuición y criterio. El facilitador debe vigilar, además, que la asistencia no convierta las reuniones en un trámite: que se celebren porque el equipo las necesita, en lugar de porque una herramienta las programe.

La función de orquestación humano-asistencia

Propósito

Decidir qué trabajo se delega a la asistencia, qué se reserva a las personas y cómo se valida lo que la asistencia produce.

Esta función es nueva. No existía en el scrum tradicional porque no había sistemas que ejecutaran trabajo. Hoy es imprescindible en cualquier equipo que use asistencia más allá del corrector ortográfico, y el desplazamiento del cuello de botella la ha vuelto, en muchos equipos de software, la más crítica de las cuatro: cuando lo escaso pasa de escribir a validar, quien orquesta esa validación ocupa el centro del sistema.

Responsabilidades clave

- Definir los **guardarraíles** de la asistencia: qué puede hacer por su cuenta, qué requiere aprobación, qué queda prohibido.
- Mantener la **transparencia algorítmica**, de modo que el equipo sepa qué decisiones o producciones provienen de sistemas asistidos.
- Validar periódicamente la calidad de lo que la asistencia produce, en especial cuando los modelos subyacentes cambian, un fenómeno conocido como **deriva del modelo** (*model drift*) que puede degradar en silencio resultados que antes eran fiables.
- Gestionar el **carril asistido** en los equipos de doble canal, asegurando que la velocidad de la máquina no desconecte al humano.

Quién la ejerce

No exige necesariamente una persona dedicada. En equipos pequeños recae en el facilitador o en un perfil técnico de referencia; en equipos de software maduros puede ser un rol especializado; en equipos no software suele asumirla el profesional senior que usa la asistencia y conoce sus límites. Lo que no admite es quedar sin dueño. Cuando nadie orquesta, la asistencia deja de ser una herramienta del equipo y empieza a marcar el ritmo por su cuenta.

Parte III — Arquitecturas de ritmo

Cómo marca el tiempo el equipo

Sobre los fundamentos de las partes anteriores se reconstruye ahora la forma concreta de trabajar. El primer elemento configurable es el ritmo. Su elección, lejos de ser una preferencia estética, es una decisión que depende del contexto del equipo, de la naturaleza del trabajo y del nivel de asistencia empleado. Recuérdese que las dos tácticas de la gestión evolutiva, la iterativa y la continua, son ambas legítimas (→ «Gestión predictiva y gestión evolutiva»): la cuestión no es cuál es más ágil, sino cuál se ajusta mejor a cada caso.

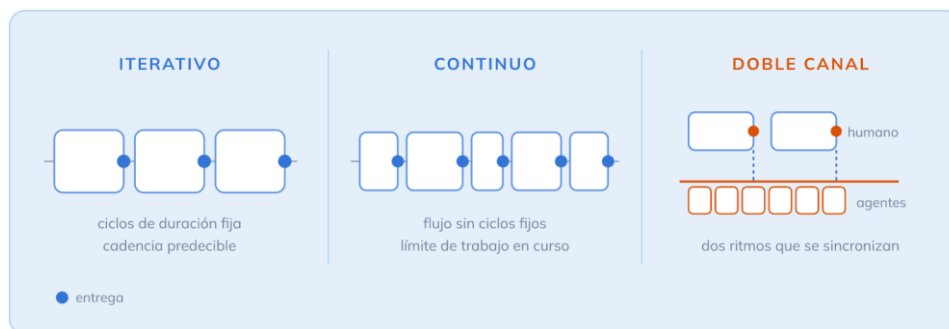


Figura 7. Las tres arquitecturas de ritmo: el doble canal, destacado, combina un carril humano iterativo con un carril asistido continuo.

Ritmo iterativo

El ritmo iterativo divide el trabajo en ciclos de duración fija (iteraciones o sprints), al final de los cuales se produce un resultado tangible y se realiza una sincronización completa del equipo.

Características

- **Tiempo prefijado.** El tiempo es fijo; el alcance es negociable. Si no se puede terminar todo, se reduce el alcance, no se alarga el tiempo.
- **Sincronización completa.** Al final de cada ciclo el equipo se reúne para revisar el incremento, recibir feedback y adaptar el plan.
- **Cadencia predecible.** Los interesados externos pueden confiar en que habrá un momento de revisión cada cierto número de semanas.

Cuándo es apropiado

- Equipos que aprenden a trabajar juntos y necesitan la disciplina de los límites.
- Entornos donde los requisitos cambian con frecuencia y conviene un punto de inspección regular.
- Equipos no software donde la coordinación con departamentos externos (legal, clientes, reguladores) exige hitos predecibles.

- El **carril humano** de los equipos de doble canal.

El ritmo iterativo es el modo donde el marco tradicional descrito en la introducción sigue siendo válido casi en su totalidad (—«Síntesis del marco scrum tradicional»). La asistencia entra aquí sobre todo como apoyo dentro de cada ciclo, sin alterar su lógica.

Ritmo continuo

El ritmo continuo no divide el trabajo en ciclos. El flujo es constante: una unidad de trabajo entra, se procesa, se valida y se entrega. No hay entrega «al final del sprint»; hay entrega cuando está lista.

Características

- **Sistema de arrastre** (*pull*). El trabajo se toma del sistema cuando hay capacidad, en lugar de empujarlo.
- **Límites de trabajo en curso** (*WIP*). Se limita de forma explícita cuántas cosas pueden estar en marcha a la vez, lo que evita la multitarea y expone los cuellos de botella.
- **Puertas de calidad** (*quality gates*). Cada etapa del flujo tiene criterios claros de entrada y salida, y nada avanza sin cumplirlos.
- **Métricas de flujo**. Tiempo de ciclo (*cycle time*), tiempo de espera (*lead time*), *throughput* y tasa de defectos.

Cuándo es apropiado

- Trabajo homogéneo y predecible, donde cada unidad se parece a la anterior.
- Equipos maduros, con alta disciplina técnica y capacidad de entrega continua.
- El **carril asistido** de los equipos de doble canal, donde los agentes operan sin sincronización humana constante.
- Equipos de soporte, mantenimiento u operaciones, donde el trabajo llega de forma impredecible.

El control es innegociable. Aquí la lección del amplificador es crítica (—«La cuarta ola: la asistencia inteligente como antítesis»). Si la asistencia magnifica el volumen y el principal riesgo es la inestabilidad, un flujo continuo sin control se vuelve peligroso: mucho producido, poco revisado, defectos acumulados. Por eso el ritmo continuo solo es seguro sobre una base de prácticas maduras (límites estrictos de trabajo en curso, pruebas automatizadas, integración continua, revisión disciplinada, puertas de calidad).

El límite de trabajo en curso deja de ser una técnica de gestión visual y pasa a ser el mecanismo que impide que la velocidad desborde la capacidad de revisión.

Doble canal

El doble canal (*dual track*) es una arquitectura en la que dos modos de trabajo coexisten y se sincronizan solo en puntos específicos. Es la respuesta que mejor conserva el valor del marco cuando en un mismo equipo conviven personas y agentes que ejecutan trabajo con autonomía, porque sus ritmos naturales son incompatibles y forzarlos al mismo compás los degrada a ambos.

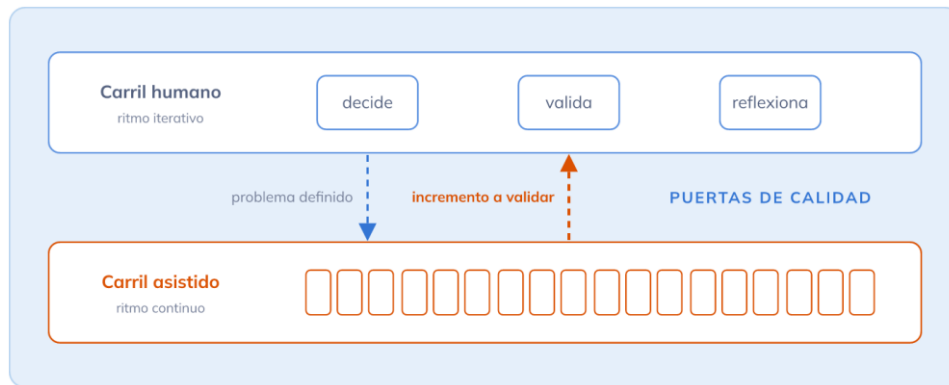


Figura 8. El doble canal en detalle: el carril humano alimenta al asistido con problemas definidos y valida lo que este devuelve, en las puertas de calidad.

Puntos de sincronización

El carril humano alimenta al asistido con problemas bien definidos y criterios de aceptación claros. El carril asistido devuelve al humano incrementos terminados para su validación. La sincronización ocurre en las puertas de calidad, no en cada paso. Ese punto de validación (donde el trabajo del carril asistido entra en el producto) es el lugar más crítico de todo el sistema, y es donde la Definición de Hecho adquiere un peso que antes no tenía (—«Incrementos y criterios de terminación»).

Riesgos del doble canal

- **Desconexión.** El carril asistido va tan rápido que el humano deja de validar, y la deuda técnica o de contenido se acumula de forma invisible.
- **Trámite vacío.** El carril humano mantiene reuniones formales mientras el asistido ya entregó hace días.
- **Pérdida de aprendizaje.** Si la asistencia lo resuelve todo, el equipo humano deja de aprender y se vuelve dependiente.

La gestión fina del doble canal es todavía territorio de experimentación más que de práctica asentada, y debe abordarse con el espíritu del ri: conocimiento experto para saber cuándo y cómo aplicarlo.

Transiciones y elección de ritmo

Ningún ritmo es intrínsecamente superior. La elección depende de un diagnóstico honesto del equipo y la organización.

Factor	Iterativo	Continuo	Doble canal
Tipo de trabajo	Complejo, variable, creativo	Homogéneo, predecible	Mixto
Madurez técnica	Media	Alta	Muy alta
Nivel de asistencia	Bajo a medio	Medio a alto	Muy alto (agentes)
Coordinación externa	Frecuente, predecible	Mínima	Puntos definidos
Tamaño del equipo	De 3 a 9 personas	Flexible	2 o más humanos, N agentes
Sector típico	No software, software inmaduro	Software maduro, operaciones	Software con agentes

Cómo transitar

- **De iterativo a continuo.** Requiere madurez técnica, automatización de pruebas y cultura de calidad. No se puede saltar.
- **De continuo a iterativo.** Puede ser necesario cuando el equipo crece, el trabajo se vuelve más complejo o la coordinación externa exige hitos.
- **Hacia el doble canal.** Solo cuando el equipo tiene claras las funciones de orquestación y validación (—«La función de orquestación humano-asistencia»).

El ritmo debe servir al empirismo, no al revés. Si el ritmo elegido impide que el equipo inspeccione y adapte, es el ritmo equivocado, por moderno que parezca.

Parte IV — Prácticas de trabajo

Las piezas configurables del sistema

Las prácticas que siguen son los componentes con los que cada equipo arma su sistema concreto. Lejos de ser obligaciones, son piezas que cumplen una función. Entre un ritmo y otro cambia su forma; su propósito se mantiene.

Inventario de necesidades

El inventario de necesidades (la pila del producto, en el vocabulario tradicional) es la fuente priorizada de todo lo que aporta valor. Es un documento vivo, que refleja lo que se conoce en cada momento y evoluciona a medida que se aprende.

Su propósito es doble: ordenar el trabajo según el valor y servir de herramienta de comunicación de la visión a todo el equipo. Lo mantiene la función de visión y decisión de valor (→«La función de visión y decisión de valor»), y su característica esencial es la priorización, con lo más valioso y urgente en primer lugar.

Con asistencia inteligente

La asistencia puede analizar feedback, datos de uso o tendencias para sugerir necesidades, y generar borradores de historias o especificaciones. Pero la decisión de incluir algo en el inventario, y su prioridad, sigue siendo humana: la asistencia informa, no decide. El inventario debe hacer visible, cuando sea relevante, si una necesidad la identificó una persona o la propuso un sistema. Lejos de ser un exceso de celo, es empirismo. Si el equipo no puede inspeccionar la fuente de una idea, no puede adaptar su criterio de selección.

Tablero de trabajo

El tablero es la representación visual del trabajo en curso. Su propósito es hacer transparente el estado del flujo, identificar cuellos de botella y facilitar la autoorganización.

En ritmo iterativo suele ser un tablero con columnas (por hacer, en curso, en revisión, terminado) donde cada elemento es una unidad de trabajo del ciclo actual y los desarrolladores se autoasignan moviendo tarjetas. **En ritmo continuo** es más complejo. Incluye todas las etapas del flujo, con límites de trabajo en curso explícitos en cada columna; si una columna alcanza su límite, el equipo deja de empujar trabajo hacia ella y ayuda a vaciarla.

Con asistencia inteligente

Los agentes pueden actualizar el tablero según su estado, y conviene incluir una columna de validación humana como puerta de calidad obligatoria. La función de orquestación (—«La función de orquestación humano-asistencia») debe poder ver de un vistazo cuánto trabajo ejecutan las personas y cuánto la asistencia. Un tablero que «miente» (porque la asistencia actualiza estados sin que el humano lo sepa) deja al equipo ciego justo donde más necesita ver.



Figura 9. El tablero con validación humana: la IA acumula trabajo en la columna «Generado», pero nada pasa a «Terminado» sin atravesar la validación humana.

Incrementos y criterios de terminación

Un incremento es una porción de valor terminada, probada y lista para que el cliente la use. No es un «ochenta por ciento hecho»: es «hecho».

La Definición de Hecho es un acuerdo que el propio equipo establece, en lugar de una imposición externa. Debe ser clara, compartida y verificable. En software suele incluir código escrito, pruebas superadas, revisión por pares, documentación actualizada y despliegue. Fuera del software puede incluir borrador revisado, aprobación del cliente, cumplimiento normativo y publicación.

Con asistencia inteligente

Esta práctica gana un peso que antes no tenía. Cuando la asistencia produce más y más deprisa, la DoD se convierte en la barrera que separa «se ha generado mucho» de «se ha entregado algo que aporta valor». Es una distinción decisiva. Un mayor volumen no equivale a un mayor valor, y un incremento generado sin validación puede incluso añadir ruido al producto (código que nadie entiende, contenido que diluye el mensaje, documentación que estorba).

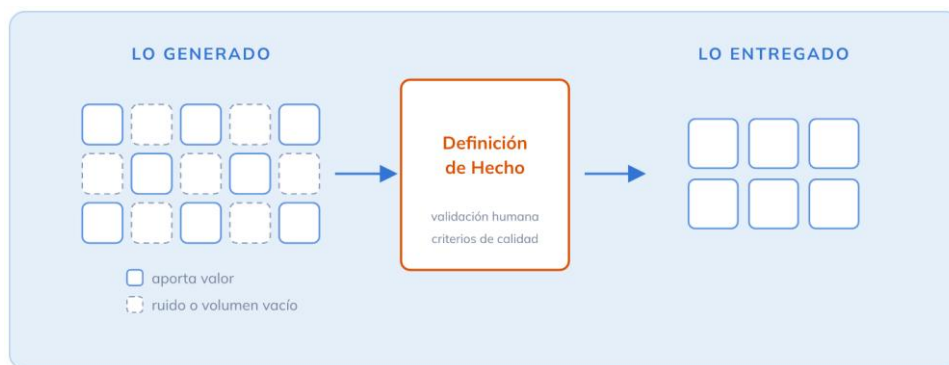


Figura 10. La Definición de Hecho como frontera: filtra el volumen generado y solo deja pasar lo que de verdad aporta valor.

Por eso la Definición de Hecho debe incluir de forma explícita la **validación humana** de cualquier trabajo generado o asistido: si un agente escribió el código, una persona debe revisarlo; si un agente redactó el documento, una persona debe verificar los hechos y el tono. La asistencia puede automatizar las pruebas, pero no puede firmar la terminación. La firma es humana.

Sincronizaciones del equipo

Las sincronizaciones son reuniones con un propósito empírico: alinear, inspeccionar, adaptar. No son rituales obligatorios; si una sincronización no aporta valor, se modifica o se suprime. Su forma cambia con el ritmo.

Planificación

- **En ritmo iterativo**, al inicio de cada ciclo: decide qué se aborda y cómo, en una conversación entre la función de visión y la de ejecución, y no consume más del diez por ciento de la duración del ciclo.
- **En ritmo continuo**, la planificación de ciclo se sustituye por el refinamiento continuo del inventario: cuando una unidad está lista para entrar en el flujo, se discute brevemente quién la aborda.
- **En doble canal**, el carril humano planifica qué problemas alimentarán al carril asistido, que necesita especificaciones claras más que planificación.

Sincronización diaria

- **En ritmo iterativo**, una reunión breve (quince minutos como máximo) que funciona como alineación táctica más que como informe de estado: el equipo inspecciona el tablero y ajusta el plan de las próximas horas.
- **En ritmo continuo** puede ser menos frecuente, o una simple revisión visual del tablero. La regla es sincronizar cuando hay desviación, no por reloj.
- **En doble canal**, el carril humano se sincroniza y el asistido se monitoriza: la función de orquestación revisa los registros de los agentes.

En todos los casos conviene que la reunión se centre en el trabajo más que en las personas, una evolución que la comunidad ágil ya recomendaba antes de la asistencia y que esta acelera.

Revisión del resultado

- **En ritmo iterativo**, al final del ciclo: el equipo muestra el incremento a los interesados y recibe feedback, en una conversación sobre el producto.
- **En ritmo continuo** no hay revisión de ciclo; cada incremento se revisa tan pronto como está listo, y el feedback es continuo.
- **En doble canal**, el carril asistido entrega de forma continua al humano, que valida antes de llevar nada al cliente.

Reflexión sobre el sistema

- **En ritmo iterativo**, la retrospectiva al final del ciclo: el equipo reflexiona sobre cómo trabajó, más que sobre qué construyó.
- **En ritmo continuo**, la reflexión es periódica o se dispara ante una anomalía: si el flujo se rompe, se para y se reflexiona.
- **En los modos con asistencia intensa**, esta reflexión incorpora un objeto nuevo: **la depuración del propio sistema de trabajo con la asistencia**.

Cuando un agente produce algo inservible, la mejora consiste en revisar y reescribir las instrucciones, los controles y los criterios con que opera la herramienta, en lugar de corregir a una persona. La retrospectiva amplía así su alcance. Abarca ya la dinámica humana y también la configuración del trabajo automatizado, junto con la pregunta incómoda de si la asistencia se está usando para evitar conversaciones difíciles o si la validación humana se ha vuelto un mero trámite.

Estimación y predicción

La estimación ágil busca **comprensión compartida** más que precisión absoluta. Cuando el equipo estima, lo valioso no es el número que sale, sino la conversación que genera sobre complejidad, dependencias y riesgos.

Unidades relativas

Los **puntos de historia** son una medida relativa de esfuerzo, complejidad e incertidumbre; cada equipo define su escala y no son comparables entre equipos. Las **tallas de camiseta** (XS, S, M, L, XL) sirven cuando la precisión numérica genera debates estériles. Los «**Ricitos de Oro**» (demasiado grande, demasiado pequeño, adecuado) ayudan a detectar unidades que conviene descomponer o agrupar.

Sin estimación (#NoEstimates)

En equipos maduros con flujo continuo, la estimación puede volverse innecesaria: si el trabajo es homogéneo y el tiempo de ciclo es estable, la predicción se hace por estadística de flujo en lugar de por estimación puntual. Más que una moda, es una consecuencia de la madurez; un equipo que no puede predecir por flujo todavía necesita estimar para aprender.

Con asistencia inteligente

Los sistemas pueden predecir el esfuerzo a partir de históricos, pero ese resultado es un dato que informa, y el veredicto sigue siendo del equipo. La estimación humana sigue siendo valiosa porque obliga a conversar, alinear expectativas y descubrir riesgos ocultos. El facilitador debe evitar que la asistencia acorte esa conversación: si un algoritmo dice «ocho puntos» y el equipo acepta sin discutir, se ha perdido el aprendizaje, que era lo único que la estimación buscaba.

Y cuando el trabajo lo ejecuta un agente, el punto de historia pierde su sentido, porque fue concebido para medir el esfuerzo relativo de una persona. Se recurre entonces a otras unidades, como el coste computacional, y sobre todo a métricas de calidad y estabilidad.

Métricas y radiadores

Las métricas deben servir al empirismo, no al control jerárquico, ni a la competencia entre equipos, ni al trámite vacío.

En ritmo iterativo

- La **velocidad** (puntos completados por ciclo) sirve para predecir; si se usa para presionar, comparar equipos o exigir más, se corrompe.
- El **gráfico de avance** (*burndown*) muestra el trabajo pendiente y detecta desviaciones tempranas.
- El **gráfico de producto** (*burnup*) muestra el trabajo acumulado hacia una meta mayor.

En ritmo continuo

- El **tiempo de espera** mide desde que una necesidad entra hasta que se entrega.
- El **tiempo de ciclo** mide desde que una unidad se activa hasta que termina.
- El **throughput** cuenta las unidades completadas por unidad de tiempo.
- La **tasa de defectos** es la proporción de trabajo que requiere rehacerse.

Radiadores de información

Gráficos visibles, actualizados por los propios equipos, que muestran el estado real. En equipos asistidos deben incluir la procedencia del trabajo (humano, asistido o mixto), en coherencia con la transparencia algorítmica. La métrica más importante mide cuánto de lo producido genera valor para el cliente, más que cuánto se produce. Medir es costoso: cada métrica debe justificarse respondiendo a cómo ayuda a inspeccionar y adaptar. Si la respuesta es «para que la dirección vea que se trabaja», se trata de un instrumento de control antes que de una métrica ágil.

Parte V — Contexto y decisión

Cómo diseñar el sistema de trabajo del equipo

Esta parte no prescribe. Ofrece un método para que el profesional diagnostique su contexto y tome decisiones informadas sobre cómo combinar todo lo anterior. Y empieza por la distinción que más condiciona cualquier decisión: si el equipo desarrolla software o no.

La asimetría estructural: software y no software

No es un factor más entre varios. Es el eje que organiza todo el diseño del sistema de trabajo, porque determina cuánto y cómo afecta la asistencia inteligente al equipo. Tratarlo como un detalle lleva a aplicar a un equipo recomendaciones pensadas para el otro, que es el error más común de esta época.

En el desarrollo de software, la disrupción es máxima, por una razón concreta: el producto coincide con aquello en lo que la asistencia es más competente. La inteligencia artificial escribe código, lo prueba, lo refactoriza y lo documenta, y los agentes pueden encadenar esas tareas con autonomía. Por eso es en software donde aparecen los casos extremos (equipos diminutos que multiplican su capacidad de entrega, ciclos que pasan de semanas a horas) y donde la presión sobre el marco iterativo es más intensa. Es el terreno natural del ritmo continuo, del doble canal y de la función de orquestación.

En el resto del trabajo del conocimiento, el impacto es real pero de otra naturaleza. En marketing, asuntos legales, recursos humanos, finanzas o consultoría, la asistencia entra como una herramienta potente que automatiza tareas concretas (redactar borradores, revisar documentos, analizar datos), pero rara vez cuestiona el marco. Las razones son estructurales:

- la coordinación con clientes, tribunales, reguladores o dirección impone ritmos que no se comprimen a voluntad;
- el producto no es ejecutable de principio a fin por un agente;
- el juicio profesional permanece en el centro de cada entrega.

Para estos equipos, el ritmo iterativo sigue siendo el punto de partida válido, y la asistencia se integra dentro de él sin desbordarlo.

La consecuencia práctica es que **no existe una respuesta única**, y que el primer paso de cualquier diagnóstico es situarse con honestidad en este eje. Lo que sigue debe leerse siempre a través de esta distinción.

Diagnóstico del equipo

Antes de elegir ritmo, prácticas o herramientas, conviene responder con honestidad a cinco preguntas.

- **¿Qué tipo de trabajo hacemos?** Si es complejo, variable y creativo, tiende al ritmo iterativo. Si es homogéneo, predecible y repetible, tiende al continuo. Si es mixto (parte creativo, parte mecánico), tiende al doble canal.
- **¿En qué sector estamos?** Es la pregunta de la asimetría anterior. En software, la asistencia ha alcanzado un nivel que cuestiona el tiempo prefijado y permite a los agentes operar en flujo continuo, aunque la validación, la decisión de valor y la coordinación externa sigan siendo iterativas. Fuera del software, la asistencia se integra como herramienta dentro de un marco que sigue siendo iterativo.
- **¿Cuál es nuestra madurez técnica?** ¿Hay automatización de pruebas, integración continua y despliegue continuo, requisitos para el flujo continuo? ¿Es baja la deuda técnica o de proceso, sabiendo que la asistencia amplifica las debilidades? ¿Se sabe medir y actuar sobre métricas de flujo?
- **¿Cuál es nuestra cultura organizativa?** ¿Confía la dirección en los equipos para autoorganizarse, sin lo cual ningún ritmo funciona? ¿Se usan las métricas para mejorar o para controlar? ¿Hay disposición a experimentar y a fallar?
- **¿Qué nivel de asistencia usamos?** ¿Generativa, con impacto menor en el ritmo? ¿Analítica, con impacto medio? ¿Agentes autónomos, con impacto mayor, que empujan hacia el doble canal o el flujo continuo?

El diagnóstico honesto importa más que la respuesta sofisticada. Una organización que adopta asistencia sin guardarraíles claros suele ver que la velocidad aumenta pero la estabilidad y la satisfacción del cliente no. Es de nuevo el amplificador en acción (—«La cuarta ola: la asistencia inteligente como antítesis»): si el equipo es frágil, la asistencia lo hace más frágil, más rápido.

Cruzar las dos preguntas más determinantes (el tipo de producto y el nivel de asistencia) da un mapa que orienta sin encorsetar.

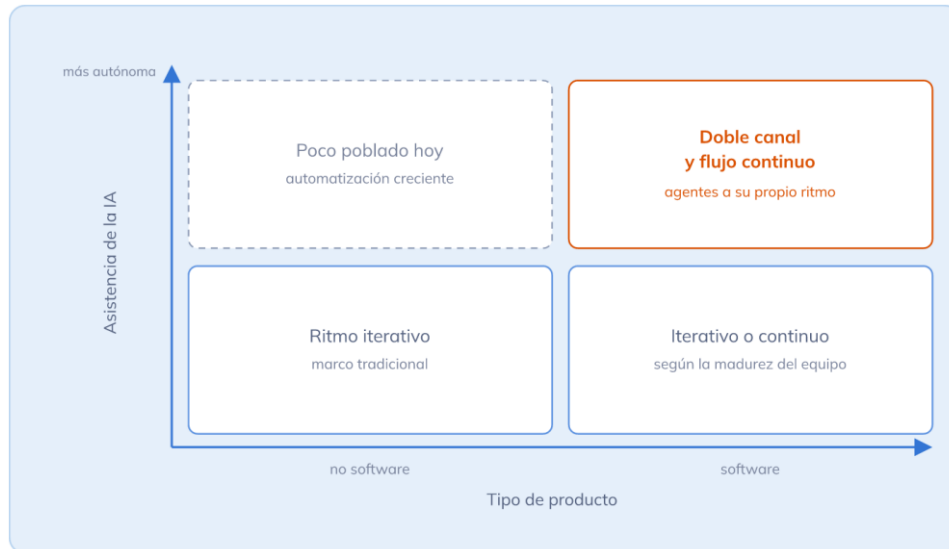


Figura 11. El mapa de decisión del equipo: el acento marca el cuadrante donde el cambio es más profundo, el del software con agentes autónomos.

La mayoría de los equipos no software se sitúan hoy en la mitad inferior, y para ellos el ritmo iterativo sigue siendo un excelente punto de partida. La mayoría de los equipos de software se mueven hacia la derecha y hacia arriba, y para ellos la elección del modo de trabajo es una decisión viva. El mapa no asigna destinos fijos: un equipo puede ocupar posiciones distintas según el proyecto, y desplazarse con el tiempo.

Elección de arquitectura de ritmo

A partir del diagnóstico, el equipo puede optar por una de estas arquitecturas.

A. Ritmo iterativo puro

Ciclos de una a tres semanas con planificación, sincronización diaria, revisión y reflexión. Apropiado para equipos no software, equipos de software en formación, equipos con alta coordinación externa o con asistencia generativa. Aporta disciplina, predecibilidad y alineación; su riesgo es la rigidez y el trámite vacío si las reuniones se vuelven formales.

B. Ritmo continuo puro

Flujo con límites de trabajo en curso, sin sprints, con entrega cuando algo está listo. Apropiado para equipos de software maduros, equipos de soporte u operaciones, o trabajo homogéneo. Aporta flexibilidad, menor coste de coordinación y velocidad real; su riesgo es la falta de reflexión periódica, la procrastinación sin tiempo prefijado y la pérdida de alineación estratégica.

C. Doble canal

Carril humano iterativo (planificación, validación, reflexión) más carril asistido continuo (ejecución, pruebas, despliegue). Apropiado para equipos de software con agentes autónomos, investigación con simulaciones masivas o contenido con generación asistida. Maximiza a la vez la velocidad de la asistencia y el juicio humano; su riesgo es la desconexión entre carriles, el trámite vacío en el carril humano y la pérdida de validación.

D. Híbrido adaptativo

El equipo alterna entre iterativo y continuo según la fase del producto o del proyecto: iterativo en el descubrimiento y el diseño, continuo en el escalado y el mantenimiento. Apropiado para equipos en transición u organizaciones con varios productos en fases distintas. Aporta flexibilidad máxima; su riesgo es la confusión, la falta de consistencia y el coste de cambiar de modo.

No hay una arquitectura correcta, sino una adecuada al contexto actual. Y ese contexto cambia, por lo que la elección debe revisarse de forma periódica en lugar de consagrarse.

Integración de asistencia: niveles y guardarraíles

La asistencia se integra por niveles, cada uno con sus propios guardarraíles, en lugar de activarse de golpe. El principio que los recorre es uno: cuanto mayor es la autonomía de la asistencia, mayor debe ser la rigurosidad de los guardarraíles humanos. La confianza no es ciega, sino calibrada, verificable y reversible.



Figura 12. Niveles de asistencia y guardarraíles: a medida que crece la autonomía de la máquina, el control humano se intensifica.

Nivel	Qué hace la asistencia	Guardarraíles
1. Generativa	Propone borradores, resúmenes, sugerencias. La persona decide, edita y valida.	El humano revisa siempre. No se acepta la salida como terminada sin validación.
2. Analítica	Analiza datos, detecta patrones, predice tendencias.	El humano entiende la lógica de la predicción. No se decide solo porque el algoritmo lo sugiera.
3. Agente especializado	Ejecuta tareas completas dentro de un dominio acotado.	Puertas de calidad humanas obligatorias. Registros transparentes. Reversión inmediata.
4. Agente orquestado	Varios agentes coordinan objetivos complejos con mínima intervención humana.	Gobernanza formal. Un humano responsable de cada agente. Auditoría periódica. Parada manual.

Los dos primeros niveles son asistencia; los dos últimos, ejecución autónoma. La frontera entre el segundo y el tercero es la que más pesa, porque es donde el trabajo deja de ser «humano con ayuda» y adquiere un ritmo propio que hay que coordinar.

El campo de posibilidades

En 2026, el debate sobre el futuro de la agilidad no tiene una respuesta única. Conviven tres posturas, que el profesional debe conocer como coordenadas de un mapa más que como verdades cerradas. Vistas a la luz de la espiral del conocimiento (—«La espiral del conocimiento»), son tres lecturas distintas de la misma síntesis en formación.

El empirismo sigue siendo el núcleo. Para una parte de la comunidad, los pilares de la agilidad (transparencia, inspección, adaptación, entrega de valor, personas sobre procesos) son más valiosos que nunca. La asistencia acelera la complejidad, pero no la elimina, y los guardarraíles empíricos importan más en un mundo veloz, porque la velocidad sin dirección es solo caos más rápido. Desde esta lectura, la asistencia es una herramienta más dentro del marco, que cambia la velocidad de ejecución sin tocar la lógica de coordinación.

El marco evoluciona desde dentro. Otra parte propone integrar la asistencia en actividades concretas (estimación potenciada, planificación asistida, facilitación automatizada, funciones redefinidas para equipos híbridos). Los roles mutan y los artefactos se adaptan, pero los pilares empíricos permanecen.

El marco tradicional pierde sentido con agentes autónomos. Una minoría influyente sostiene que los marcos ágiles se diseñaron para un mundo donde las personas éramos el único cuello de botella. Si la asistencia elimina ese cuello, los ciclos, las reuniones diarias y las retrospectivas resultan innecesarios para la máquina, y proponen flujos continuos, doble canal o lenguajes de patrones que reemplazan los marcos completos. Desde esta lectura, los marcos tradicionales completaron su misión para la era previa, y ahora distintas inteligencias en el equipo necesitan distintos ritmos de coordinación.

Estas tres posturas no son tres respuestas entre las que elegir, sino tres tensiones de una misma síntesis que aún se está formando. Al profesional de la agilidad no le corresponde defender a ultranza ninguna de ellas. Le corresponde conocer las tres coordenadas, diagnosticar en qué tipo de equipo y organización trabaja, y ayudar a su equipo a experimentar de forma consciente, manteniendo el empirismo como faro. Esa es, en el fondo, la actitud *ri*: construir con criterio el propio modo de trabajo en lugar de aplicar una receta. Y, fiel a la espiral, hacerlo sabiendo que la síntesis que hoy se construya será mañana la tesis que una nueva antítesis vuelva a poner a prueba.

Parte VI — Principios y valores

Las raíces bajo el hielo

Si las arquitecturas, las prácticas y las funciones son piezas configurables, los principios y los valores son el criterio que dice si la configuración resultante sigue siendo ágil. Son lo que la antítesis consolida, no lo que cuestiona, y por eso cierran la guía: son el suelo firme sobre el que se construye cualquier síntesis, presente o futura.

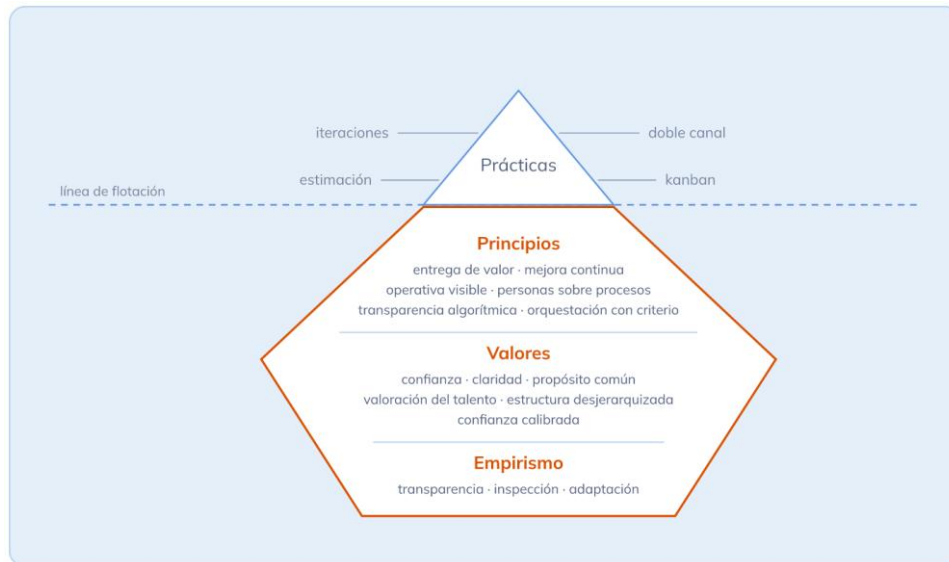


Figura 13. Las raíces bajo el hielo: las prácticas son la punta visible que cambia con cada giro de la espiral; los principios, los valores y el empirismo son la masa que las sostiene.

Principios

Los principios son el soporte de las prácticas. Sin ellos, las herramientas son solo ruido.

- **Entrega de valor.** Entrega temprana y continua de valor al cliente, para lo que es necesario que este colabore con el equipo y se comparta y comprenda su visión.
- **Mejora continua.** Reflexión frecuente sobre los métodos de trabajo, cuestionando su efectividad y adaptándolos. El mismo esfuerzo autocrítico se aplica a la mejora de los productos y servicios.
- **Desarrollo iterativo, incremental o continuo.** El producto no se construye según un plan inicial cerrado, sino arrancando desde un mínimo viable sobre el que se añaden incrementos, o fluyendo de forma continua.
- **Ritmo de trabajo sostenible.** Alcanzar un ritmo que evite tanto la expansión del trabajo hasta llenar el tiempo disponible como la presión de descubrir los retrasos demasiado tarde. La velocidad que aporta la asistencia no debe convertirse en una expectativa de sobreesfuerzo permanente.
- **Atención continua a la excelencia.** Empleo de técnicas que garanticen la calidad y permitan detectar errores pronto. Este principio gana importancia cuando la

asistencia genera más trabajo más deprisa: sin control maduro, el volumen acelera la aparición de defectos en lugar de prevenirla.

- **Operativa visible.** La información se comparte con claridad para facilitar la colaboración, identificar impedimentos pronto y permitir que todo el equipo conozca el estado del trabajo y aporte ideas.
- **Cadencia y sincronización.** Predecir la frecuencia de reuniones y entregas, y sincronizar el trabajo de diferentes equipos o carriles. El doble canal es, en el fondo, un problema de cadencia: coordinar dos ritmos distintos (—«Doble canal»).
- **Personas sobre procesos.** La inteligencia colectiva del equipo, su conocimiento tácito, es responsable de la calidad del producto. Los procesos y la asistencia son ayudas, nunca sustitutos. Lejos de debilitarse, este principio se vuelve el más decisivo en la era de la asistencia.

A estos principios clásicos, la entrada de la asistencia inteligente añade dos que antes no hacía falta enunciar.

- **Transparencia algorítmica.** La transparencia clásica se ocupaba del estado del trabajo; esta se ocupa de su procedencia. El equipo debe poder saber qué decisiones y producciones fueron humanas y cuáles asistidas, y qué guardarraíles y límites de autonomía rigen la asistencia. No se trata de entender el funcionamiento interno de la máquina, sino de poder inspeccionar el origen de lo que se produce. Si no se puede inspeccionar de dónde viene una decisión, no hay empirismo posible sobre ella.
- **Orquestación con criterio.** La máquina ejecuta; la persona juzga. Orquestar no es operar, sino decidir qué se delega, qué se valida y qué se discute, sin que la cadena de responsabilidad humana quede nunca oculta. De aquí se sigue una regla práctica: lo que la asistencia produce y ningún humano puede explicar ni asumir no debería llegar al cliente.

Valores

La cultura de la empresa es la suma de sus características organizativas y de gobernanza, que pueden acelerar o frenar el desarrollo de la agilidad.

- **Asertividad.** Capacidad de expresar lo que se piensa de forma directa, honesta y respetuosa.
- **Valoración del talento.** Reconocer que el valor del producto depende del conocimiento tácito de las personas.
- **Claridad.** Transparencia en la información, en las decisiones y en los objetivos.
- **Confianza.** Base de la autoorganización. Sin confianza no hay equipo, solo supervisión.
- **Estructura desjerarquizada.** Las decisiones se toman donde está el conocimiento, no donde está el rango.
- **Propósito común.** El equipo comparte un objetivo que trasciende los intereses individuales.
- **Confianza calibrada.** Ni adopción ciega de la asistencia ni rechazo por desconfianza, sino la capacidad de evaluar cuándo amplifica las capacidades del equipo y cuándo expone sus debilidades. La confianza en la máquina se construye igual que la confianza en un compañero: con transparencia, inspección y adaptación.

Estos principios y valores son las raíces que permanecen y dan soporte a las nuevas prácticas. Ese es el sentido último de una formación que no busca enseñar un modelo, sino la capacidad de construir el propio: el campo de scrum de cada equipo, en cada momento de la espiral.

Apéndice

Glosario de los términos de la guía, referencias bibliográficas, recursos y comunidad profesional de Scrum Manager, y control de calidad de la formación.

Glosario

Adaptación: tercer pilar del empirismo: ajustar cuanto antes lo que se desvía de los límites aceptables, para minimizar desviaciones mayores.

Agente autónomo: extremo del espectro de la asistencia inteligente en el que un sistema ejecuta tareas completas con mínima intervención humana, bajo supervisión.

Amplificador: efecto de la asistencia inteligente sobre un equipo: magnifica sus condiciones de partida, para bien en los equipos sólidos y para mal en los frágiles.

Antítesis: circunstancias nuevas que ponen en evidencia los límites de una tesis y señalan que el contexto ha cambiado. En esta guía, la asistencia inteligente es la antítesis de la gestión ágil establecida.

Asistencia generativa: extremo del espectro de la asistencia inteligente en el que las herramientas proponen borradores, resúmenes o sugerencias mientras la persona decide, edita y valida.

Asistencia inteligente: nombre que la guía da a los sistemas de inteligencia artificial que participan en el trabajo, en un espectro que va de la asistencia generativa a los agentes autónomos.

Carril asistido: en el doble canal, el modo de trabajo en flujo continuo en el que los agentes ejecutan, prueban y despliegan, y devuelven incrementos para su validación.

Carril humano: en el doble canal, el modo de trabajo iterativo en el que las personas planifican, validan y reflexionan, y alimentan al carril asistido con problemas bien definidos.

Confianza calibrada: valor que añade la asistencia inteligente: evaluar cuándo la asistencia amplifica las capacidades del equipo y cuándo expone sus debilidades, sin adopción ciega ni rechazo por desconfianza.

Conocimiento explícito: conocimiento codificado, que puede transmitirse con lenguaje formal y sistemático; reside en los procesos.

Conocimiento tácito: conocimiento personal y específico del contexto, difícil de formalizar y comunicar; reside en las personas y sostiene la calidad en el trabajo del conocimiento.

Definición de Hecho (DoD): acuerdo del equipo, claro, compartido y verificable, que fija cuándo un incremento está terminado; con asistencia incluye la validación humana de todo lo generado.

Deriva del modelo: degradación silenciosa de los resultados de la asistencia cuando cambian los modelos subyacentes; exige validar periódicamente lo que produce.

Doble canal: arquitectura de ritmo en la que coexisten un carril humano iterativo y un carril asistido continuo, sincronizados en las puertas de calidad y no en cada paso.

Empirismo: base de cualquier desarrollo ágil: el conocimiento válido proviene de la experiencia y la observación. Se apoya en la transparencia, la inspección y la adaptación.

Empresas del conocimiento: las que desarrollan productos o servicios basándose en el conocimiento de las personas más que en las herramientas y los procesos.

Espiral del conocimiento: modelo de Nonaka y Takeuchi: el conocimiento avanza por tesis, antítesis y síntesis, y cada síntesis es la tesis del siguiente giro.

Función de ejecución y validación: la que construye el incremento y asegura que cumple los criterios de calidad y de terminación; con asistencia, sus personas curan y validan.

Función de facilitación y mejora del sistema: la que logra que el equipo trabaje sin fricciones, remueve impedimentos y mejora de forma continua el sistema de trabajo.

Función de orquestación humano-asistencia: función nueva que decide qué se delega a la asistencia, qué se reserva a las personas y cómo se valida lo producido.

Función de visión y decisión de valor: la que define qué problema se resuelve, para quién y por qué importa, y maximiza el valor entregado al cliente y al negocio.

Gestión evolutiva: combinación de desarrollo incremental, fases solapadas y conocimiento en las personas; entrega pronto un producto mínimo viable e incrementa su valor de forma continua.

Gestión predictiva: combinación de desarrollo completo, trabajo en cascada y conocimiento en los procesos (ingeniería secuencial); mide el éxito por el cumplimiento del plan.

Guardarraíles: límites que definen qué puede hacer la asistencia por su cuenta, qué requiere aprobación y qué queda prohibido; se endurecen a medida que crece su autonomía.

Incremento: porción de valor terminada, probada y lista para que el cliente la use.

Ingeniería concurrente: gestión evolutiva con producción basada en procesos: usa recursos de la gestión ágil, pero se centra en la calidad de los procesos, no en las personas.

Inspección: segundo pilar del empirismo: examinar con frecuencia los artefactos y el avance hacia el objetivo para detectar desviaciones indeseadas.

Inventario de necesidades: nombre que la guía da a la pila del producto (product backlog): fuente priorizada y viva de todo lo que aporta valor.

Límite de trabajo en curso: cantidad máxima de elementos que pueden estar en marcha a la vez en una etapa del flujo (WIP); evita la multitarea y expone los cuellos de botella.

Puertas de calidad: criterios explícitos de entrada y salida de cada etapa del flujo; nada avanza sin cumplirlos. En el doble canal, puntos de sincronización entre carriles.

Puntos de historia: medida relativa de esfuerzo, complejidad e incertidumbre, propia de cada equipo; pierden su sentido cuando el trabajo lo ejecuta un agente.

Radiador de información: gráfico visible, actualizado por el propio equipo, que muestra el estado real del trabajo; en equipos asistidos, también su procedencia.

Ritmo continuo: arquitectura de ritmo sin ciclos: el trabajo entra, se procesa, se valida y se entrega cuando está listo, con límites de trabajo en curso y puertas de calidad.

Ritmo iterativo: arquitectura de ritmo que divide el trabajo en ciclos de duración fija, con un resultado tangible y una sincronización completa del equipo al final de cada uno.

Shu ha ri: las tres etapas del aprendizaje de una habilidad: imitar una técnica, coleccionar técnicas y crear las propias con criterio experto.

Síntesis: comprensión nueva que reconcilia lo que seguía siendo válido de la tesis con lo que la antítesis ha revelado; no suprime ni suma, y se convierte en la tesis del siguiente giro.

Tablero de trabajo: representación visual del trabajo en curso que hace transparente el estado del flujo; con asistencia incluye una columna de validación humana.

Tesis: respuesta válida a una situación, construida con el conocimiento y las circunstancias de su momento, que se convierte en el modo establecido de hacer las cosas.

Tiempo prefijado: técnica del ritmo iterativo (timeboxing) en la que el tiempo es fijo y el alcance negociable: si no se puede terminar todo, se reduce el alcance.

Transparencia: primer pilar del empirismo: los aspectos significativos del trabajo son visibles para quienes responden del resultado.

Transparencia algorítmica: principio nuevo: el equipo debe poder saber qué decisiones y producciones fueron humanas y cuáles asistidas, y qué guardarraíles rigen la asistencia.

Referencias bibliográficas

El diagnóstico de la antítesis y el inventario de propuestas de la síntesis se apoyan en el estado del conocimiento profesional y académico a mediados de 2026. La guía prioriza deliberadamente los principios y las dinámicas de fondo sobre las herramientas concretas, que cambian con rapidez. Entre las fuentes de contexto empleadas están el informe DORA 2025 sobre desarrollo asistido por IA, los informes anuales sobre el estado de la agilidad, los análisis sobre la evolución de los roles ágiles y el achicamiento de los equipos, y las propuestas de ciclo de vida y de marcos aumentados con IA surgidas en la comunidad.

Scrum Manager® mantiene una postura agnóstica y ofrece el inventario de conocimiento disponible para que cada profesional desarrolle, con criterio, su propio modelo de trabajo. Las obras que la guía cita se relacionan a continuación, en orden alfabético.

Beck, Kent; Beedle, Mike; van Bennekum, Arie; Cockburn, Alistair; Cunningham, Ward; Fowler, Martin; Grenning, James; Highsmith, Jim; Hunt, Andrew; Jeffries, Ron; Kern, Jon; Marick, Brian; Martin, Robert C.; Mellor, Steve; Schwaber, Ken; Sutherland, Jeff y Thomas, Dave (2001). *Manifiesto por el Desarrollo Ágil de Software*, agilemanifesto.org.

Cockburn, Alistair (2006). *Agile Software Development: The Cooperative Game* (2.ª ed.), Addison-Wesley.

DORA (2025). *State of AI-assisted Software Development 2025*, Google Cloud.

Nonaka, Ikujiro y Takeuchi, Hirotaka (1995). *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press.

Nonaka, Ikujiro y Toyama, Ryoko (2004). «Knowledge Creation as a Synthesizing Process». En Hirotaka Takeuchi e Ikujiro Nonaka (eds.), *Hitotsubashi on Knowledge Management*, pp. 91–124. Singapore: John Wiley & Sons (Asia).

Schwaber, Ken (1995). «SCRUM Development Process», *OOPSLA'95 Workshop on Business Object Design and Implementation*, ACM.

Takeuchi, Hirotaka y Nonaka, Ikujiro (1986). «The New New Product Development Game», *Harvard Business Review*, 64(1), Harvard Business Publishing.

Información, recursos y comunidad profesional

Scrum Manager mantiene un ecosistema de recursos abiertos y de comunidad para seguir aprendiendo después de esta guía:

[Club Agile →](#)[Skill Arena →](#)[Academy →](#)[BoK/Wiki →](#)[Comunidad →](#)[Tiras de cómic →](#)[Acerca de la certificación Scrum Manager® →](#)[Preguntas frecuentes sobre cursos y exámenes Scrum Manager® →](#)

Síguenos en  [LinkedIn](#)

Mejora continua y control de calidad

El control de calidad de Scrum Manager se basa en las valoraciones de sus estudiantes. Con tu opinión nos ayudas a mantener el nivel de nuestros materiales, cursos, centros y profesores.

Si has participado en una actividad de formación auditada por Scrum Manager®, te rogamos y agradecemos que valores la calidad de la formación recibida. La información que se recoge es anónima. Puedes enviarnos tus comentarios desde el «Área de miembros»: <https://scrummanager.com>