

# Un Sprint de Dos Semanas para el Conocimiento

*De Agile al Pensamiento de Diseño*

---

**Jessica Talisman, MLS**

21 de abril de 2026 · Intentional Arrangement

En 1986, Hirotaka Takeuchi e Ikujiro Nonaka publicaron «The New New Product Development Game» en *Harvard Business Review*, describiendo cómo Honda, Canon y Fuji-Xerox organizaban sus equipos para fabricar copiadoras y automóviles. Jeff Sutherland leyó ese artículo, lo combinó con los principios de manufactura lean y la programación orientada a objetos, y en 1993 lo adaptó al desarrollo de software en Easel Corporation. Ken Schwaber formalizó el método en la conferencia OOPSLA de 1995. En 2001, diecisiete desarrolladores de software se reunieron en un albergue de esquí en Snowbird, Utah, y firmaron el Manifiesto Ágil, declarando que estaban «descubriendo mejores maneras de desarrollar software.» No conocimiento ni significado. Un marco de manufactura aplicado al software.

El propio lenguaje del Manifiesto Ágil es explícito sobre su alcance, y sin embargo la industria ha aplicado la metodología ágil a prácticamente todo lo que existe en un entorno tecnológico — incluyendo el trabajo del conocimiento, los sistemas semánticos y ahora la inteligencia artificial. Y en el mismo aliento, organizaciones y expertos colman la industria con discursos sobre innovación y el surgimiento de un nuevo paradigma en las formas de trabajo. Sin embargo, esas formas de trabajo permanecen encerradas dentro de las metodologías ágiles y Scrum, diseñadas para la línea de producción y la base de código — no para el trabajo profundamente humano, recursivo y desordenado de construir significado a partir de la complejidad.

## **Scrum es un calendario de manufactura**

Volvamos al artículo de Takeuchi y Nonaka, cuyo título — «The New New Product Development Game» — describe cómo los equipos de producto en empresas japonesas desarrollaban bienes físicos como la copiadora FX-3500 y el Honda City. Los autores recurren a la metáfora del rugby porque el equipo «intenta recorrer la distancia como una unidad, pasándose el balón de un lado al otro.» La unidad de entrega es un prototipo funcional de un artefacto material. El ciclo de retroalimentación es una máquina que funciona o falla.

El libro de Sutherland de 2014, *Scrum: The Art of Doing Twice the Work in Half the Time*, refuerza esto con casos de estudio militares y de manufactura como la producción del caza F-86 y Toyota. El libro reconstruye el argumento sobre la misma premisa: que aquello que se construye tiene partes discretas, inspeccionables y verificables que, una vez terminadas, están terminadas.

La Guía Scrum de 2020 define el Incremento como «un peldaño concreto hacia el Objetivo del Producto», uno que es «aditivo respecto a todos los Incrementos anteriores y rigurosamente verificado, asegurando que todos los Incrementos funcionen en conjunto.» Estas palabras tienen sentido si se está construyendo una aplicación móvil, con una pantalla de inicio de sesión y una página de configuración. Pero para los sistemas de conocimiento, estas palabras son destructivas, porque los componentes de una arquitectura semántica no son aditivos de la manera en que Scrum prescribe. Un vocabulario controlado no es un subcomponente de una taxonomía del mismo modo en que una pantalla de inicio de sesión lo es de una aplicación. Una taxonomía sin un control de vocabulario disciplinado como fundamento no funciona bien, o directamente no funciona. No se puede entregar media taxonomía y llamarla un Incremento.

Scrum asume problemas dóciles que pueden enmarcarse en el tiempo, estimarse con planning poker, descomponerse en historias de usuario, rastrearse en un gráfico de velocidad y declararse resueltos frente a una lista de verificación. Si tan solo el conocimiento fuera así de dócil. La organización del conocimiento está muy lejos de serlo. Con más de cincuenta años de antigüedad, en 1973 Horst Rittel y Melvin Webber advirtieron que aplicar el paradigma de la ingeniería de manera uniforme a todos los problemas estaba «condenado al fracaso.» Y sin embargo aquí estamos, aplicando un calendario de manufactura y sus metodologías al conocimiento, esperando que el conocimiento quepa en los confines de viejas formas de trabajar — un artefacto de la Tercera Revolución Industrial impuesto sobre las exigencias de la Cuarta.

En 2016, Klaus Schwab escribió sobre este patrón histórico. La Primera Revolución Industrial, que comenzó alrededor de 1760, nos dio el vapor y la producción mecánica. La Segunda, a finales del siglo XIX, nos dio la electricidad y la cadena de montaje — el mundo que produjo Toyota, la manufactura lean y la lógica de producción que Scrum aún lleva como ethos central. La Tercera, que comenzó en la década de 1960, nos dio los semiconductores, la computación mainframe y finalmente internet — el mundo en que nació Agile, por desarrolladores de software, para el desarrollo de software. La Cuarta Revolución Industrial, argumentó Schwab, es fundamentalmente diferente en su naturaleza, pues está definida por la fusión de la inteligencia artificial, los sistemas ciberfísicos y las tecnologías del conocimiento que difuminan las fronteras entre los mundos físico, digital y biológico.

Los problemas de esta era no son necesariamente problemas de ingeniería con requisitos estables. Estamos lidiando con preguntas como de qué manera anclar la inteligencia artificial en el significado, cómo organizar el conocimiento en evolución y cómo construir sistemas capaces de explicarse a sí mismos. Estos son problemas retorcidos que exigen

nuevas epistemologías y marcos para la resolución de problemas. Agile puede haber sido la metodología correcta para la Tercera Revolución Industrial, pero ya no estamos en ella. Los marcos no se han puesto al día, y el coste de ese rezago se agrava cada vez que una organización impone los principios de la Tercera Revolución Industrial al trabajo de la Cuarta.

En otras palabras, esto no es un problema de producto ni de plataforma — es el desafío de aprender a abrazar nuevas formas de trabajar. Es un problema de infraestructura gobernado por la epistemología: una filosofía fundamental sobre el conocimiento.

### **Por qué el conocimiento no puede ejecutarse en sprints**

Una ontología es, según la definición fundacional de Thomas Gruber, «una especificación de una conceptualización.» Una conceptualización es un todo, no partes de un todo. Especificar la mitad produce incoherencia lógica, no un producto mínimamente viable. Por ejemplo, la norma ANSI/NISO Z39.19 para la construcción de vocabularios controlados exige resolver la sinonimia, la homonimia y el alcance antes de que los términos entren en producción, porque cualquier inconsistencia se propaga hacia abajo en cada uso posterior. SKOS, el estándar del W3C para codificar sistemas de organización del conocimiento, requiere que las relaciones jerárquicas formen estructuras coherentes con cierre transitivo, y OWL 2 exige que los axiomas de clase sean satisfacibles. Estas son precondiciones necesarias para que los sistemas lógicos razonen y expresen significado.

En la práctica, he observado organizaciones intentando entregar sistemas de conocimiento de forma ágil en dos patrones, y ninguno termina bien. En el primero, el equipo celebra stand-ups, ceremonias y retrospectivas. El taxonomista trae tickets y añade quince términos al vocabulario del producto. Los tickets se cierran y el gráfico de velocidad sube. Nada se integra, porque la integración es la parte difícil y no cabe en dos semanas, de modo que va al backlog. Seis meses después, el vocabulario es una colección de términos con alcances inconsistentes, la ontología tiene clases huérfanas y el grafo de conocimiento — si existe — no puede responder consultas de manera confiable. Pero el burndown chart es impecable y el liderazgo está ciegamente satisfecho porque las casillas fueron marcadas.

En el segundo patrón, el equipo toma prestado de otras metodologías sin dejar de llamarlo ágil. La taxonomista conduce análisis de dominio y entrevistas con partes interesadas durante varias semanas — eso es etnografía. Realiza card sorting y modelado conceptual — eso es investigación en diseño. Itera con expertos en la materia — eso es diseño participativo. Finalmente entrega un modelo coherente. El ritmo del sprint fue irrelevante para el trabajo. Martin Fowler denominó este fenómeno en su conferencia magistral de 2018: «faux-agile» — el Complejo Industrial Ágil.

Ron Jeffries, otro signatario original del Manifiesto Ágil, les dijo a los desarrolladores ese mismo año, 2018, que abandonaran la palabra por completo. Dave Thomas declaró en 2014 que «Ágil» se había convertido en «un imán para cualquiera con puntos que exponer, horas que facturar o productos que vender.» Cuando tres de los diecisiete autores de un marco dicen que ha sido vaciado de contenido, uno tiene que preguntarse si

Agile es apropiado para todo el trabajo tecnológico y si ha llegado el momento de revisar las formas de trabajar.

### **La innovación no cabe en un backlog**

El problema más profundo es que Agile y Scrum nunca fueron diseñados para inspirar o fomentar la innovación. Fueron diseñados para entregar requisitos conocidos con eficiencia creciente. Todo el aparato de la Guía Scrum — product backlog, sprint planning, Definición de Hecho — presupone que el problema ha sido formulado por un product owner capaz de escribirlo en una historia de usuario. Pero el trabajo del conocimiento, como el diseño, es retorcido. La primera propiedad de un problema retorcido según Rittel y Webber es que «No existe una formulación definitiva.» La segunda propiedad establece que no hay regla de parada. Y la quinta propiedad ilumina que cada solución es una «operación de un solo disparo» porque no hay oportunidad de aprender por ensayo y error sin consecuencias.

Dada la naturaleza de los problemas retorcidos, parecería que Agile está mal equipado para enfrentarlos. Y por la misma razón, mal equipado para la naturaleza dinámica del conocimiento, donde las infraestructuras de conocimiento nunca son operaciones de un solo disparo, requieren reglas de parada y desafían sin lugar a dudas las formulaciones definitivas.

Por lo tanto, la organización del conocimiento en un dominio empresarial es retorcida exactamente en este sentido. El problema de cómo representar «cliente» en el modelo semántico de una empresa de telecomunicaciones no tiene una respuesta correcta preexistente. Su formulación depende de quién hace qué con esa representación, y eso cambia a medida que se construye. El alcance del vocabulario se desplaza conforme se entrevista a las partes interesadas. La taxonomía se reorganiza al encontrar casos límite. Los axiomas de la ontología se tensan o aflojan cuando el razonador descubre contradicciones. No hay que equivocarse — esto no se pliega al principio ágil de bienvenida a los requisitos cambiantes, porque los requisitos se constituyen *a través* del trabajo. El problema está en construcción, evolucionando frecuentemente, durante su resolución.

Las organizaciones que gestionan todo el trabajo a través de Scrum están, en efecto, optimizando para la previsibilidad en dominios que exigen exploración. Los resultados inhiben la innovación y adoptan la apariencia de productividad. Los gráficos de velocidad suben mientras la arquitectura intelectual subyacente se aplanan en entregas fragmentadas, desconectadas del continuo que es el conocimiento.

### **El pensamiento de diseño pertenece al pantano**

Donald Schön trazó la distinción de que «En la variada topografía de la práctica profesional, hay una meseta alta y firme donde los profesionales pueden hacer uso efectivo de la teoría y la técnica basadas en la investigación, y hay unas tierras bajas pantanosas donde las situaciones son “enredos” confusos incapaces de solución técnica.» Agile vive en la

meseta mientras que el trabajo del conocimiento vive en el pantano. El término de Schön para describir cómo los profesionales competentes navegan el pantano fue «reflexión-en-acción» — la recalibración iterativa y tácita que ocurre mientras el trabajo transcurre.

El pensamiento de diseño, en el linaje de Herbert Simon, Richard Buchanan y Schön, es la metodología del pantano. Asume que el problema está en construcción. Sus cinco fases — empatizar, definir, idear, prototipar, probar — no son un flujo lineal sino un proceso recursivo e iterativo que espera que la definición del problema cambie a medida que la comprensión se profundiza.

La investigación en el *Handbook of Human-Centered Artificial Intelligence* confirma esta alineación — integrar el pensamiento de diseño en el desarrollo de IA hace que los sistemas sean más centrados en el usuario, iterativos e impactantes, enfatizando la gobernanza de datos, la transparencia, la explicabilidad y la mitigación del sesgo. La Tepper School of Business de Carnegie Mellon University imparte ahora un programa dedicado a pensamiento de diseño con IA, argumentando que la combinación aborda «problemas complejos y sistémicos» que los marcos lineales no pueden gestionar con éxito. La práctica de diseño empresarial de IBM, desarrollada a lo largo de dos décadas, se ha aplicado en compromisos con clientes precisamente porque mantiene la experiencia humana en el centro mientras la tecnología evoluciona por debajo — el pensamiento de diseño es «atemporal», mientras que la tecnología está «fecha».

Un estudio de 2026 en *Scientific Reports* examinó la relación entre el pensamiento de orden superior, el uso de chatbots de IA generativa y la creatividad en ingeniería, encontrando que el pensamiento de orden superior tenía una importancia significativamente mayor para los resultados creativos que el uso de herramientas de IA por sí solo — reforzando que la profundidad cognitiva, no la velocidad de las herramientas, impulsa la innovación. La investigación en las *Proceedings of the Design Society* argumenta además que la IA generativa está desplazando el rol del diseñador de ejecutor a configurador de paradigmas, haciendo que el pensamiento de diseño sea más esencial a medida que la IA gestiona las tareas de producción mientras los humanos deben guiar el encuadre, la ética y el significado.

Para ser claros, estos marcos no son distintos sabores de la misma metodología. Scrum y el pensamiento de diseño representan epistemologías completamente diferentes. Scrum asume que el problema te fue entregado, mientras que el pensamiento de diseño asume que el problema debe encontrarse y lidiarse con él.

## **¿Es la IA una herramienta de conocimiento?**

Esto nos lleva a la pregunta que la industria evita y que yo, por mi parte, llevo tiempo deseando formular. ¿Es la IA una herramienta de conocimiento? ¿Cómo se clasifica?

Un modelo de lenguaje de gran escala genera secuencias plausibles de texto. Volviendo al artículo de Bender, Gebru, McMillan-Major y Mitchell mencionado anteriormente, los autores definieron un modelo de lenguaje como un sistema para ensamblar formas

lingüísticas según patrones probabilísticos, «sin ninguna referencia al significado.» Sin referencia al significado, dado que los LLMs sin configuración adicional carecen de anclaje externo. Optimizan para la plausibilidad, no para la verdad, y por lo tanto, como sistemas autónomos, pueden clasificarse mejor como herramientas de procesamiento de información.

Se convierte en una herramienta de conocimiento únicamente cuando se ancla en una capa semántica — vocabularios controlados, taxonomías, ontologías, grafos de conocimiento — que proporciona referencia y procedencia. Las propias medidas correctivas de la industria y el mercado asociado confirman esto como un hecho. La generación aumentada por recuperación, los grafos de conocimiento para LLMs y lo que el discurso reciente denomina «ingeniería de contexto» son, bajo el empaquetado, la infraestructura semántica que debería haberse construido primero. Pero fue añadida a posteriori para prevenir alucinaciones que estaban prácticamente garantizadas desde el momento en que el modelo fue desplegado sin ella.

El artículo sobre RAG de 2020 señaló este problema, identificando que la memoria paramétrica es opaca y no puede actualizarse, y que la recuperación desde fuentes estructuradas es el modo en que la procedencia se captura y codifica. Pero parece que la industria tiene un problema de memoria, o el mensaje llegó demasiado pronto para que la mayoría pudiera escucharlo.

Irónicamente, la procedencia es un elemento central de la gestión del conocimiento y el trabajo ontológico, pues constituye un principio fundacional de la disciplina y su espacio de problemas. Sin procedencia, el conocimiento no puede existir. Las organizaciones están descubriendo ahora que la necesitan, porque sus sistemas generativos no pueden explicarse a sí mismos. Llámalo linaje, llámalo trazabilidad. Pero si construyes para el conocimiento, llámalo procedencia, porque la procedencia abarca trazas, linaje, citaciones, lo temporal y lo espacial.

De modo que si la IA generativa es una herramienta de procesamiento de información, sostengo que para que la IA se convierta en una herramienta de conocimiento debe haber procedencia y lógica descriptiva. Y para ser francos, nadie llega a la procedencia haciendo sprints — esta se arquitecta y se sostiene.

## **El conocimiento es una espiral, no un sprint**

La ironía no pasa desapercibida en este momento. El mismo Nonaka que coescribió el artículo de 1986 que inspiró Scrum desarrolló el modelo SECI de creación de conocimiento en 1995, describiendo el conocimiento como una espiral entre formas tácitas y explícitas, que avanza a través de la socialización, la externalización, la combinación y la internalización.

La espiral no tiene fin, como es propio de las espirales, emblema de la vida. Cada rotación crea nuevo conocimiento que se convierte en el sustrato de la siguiente. No existe una Definición de Hecho porque la definición de lo que se está haciendo está siendo cons-

truida en sí misma. Scrum tomó la estructura de equipo del primer artículo de Nonaka y descartó su siguiente década de trabajo porque no encajaba limpiamente en Scrum. Los mecanismos del trabajo del conocimiento, como Nonaka argumentó durante una década, son la metáfora, la habilidad tácita, el aprendizaje por proximidad, el espacio compartido y la paciente conversión de intuiciones en modelos y de modelos de vuelta en práctica.

El Ontology Pipeline™ refleja esta espiral. Se comienza con un vocabulario controlado — un proceso disciplinado que implica acuerdos sobre lo que significan los términos. Se madura hacia una taxonomía introduciendo jerarquía y estructurando los términos en relaciones padre-hijo. Posteriormente, la taxonomía se madura hacia un tesoro añadiendo relaciones asociativas y sinonimia, codificadas en SKOS. El tesoro es el trabajo base a partir del cual se desarrolla una ontología, incorporando semántica formal mediante OWL y RDF. Se arquitecta un grafo de conocimiento, ensamblando componentes que incluyen los activos de conocimiento desarrollados a través del marco iterativo del Ontology Pipeline™.

Cada etapa prepara el terreno para la siguiente y las etapas no pueden saltarse ni eludirse. No se puede entregar el grafo de conocimiento sin una ontología, y la ontología debe ser lógicamente consistente y coherente para dotar de sentido a los datos. Un grafo de conocimiento sin control de vocabulario arriesga convertirse en un mentiroso convincente o en fallar al trabajar con lenguaje natural. Porque los activos de conocimiento son dimensionalmente interdependientes, el conocimiento es efectivamente una espiral, incapaz de ser contenida por la lógica de la manufactura.

## **El patrón y el remedio**

Las organizaciones presupuestan para plataformas — Confluence, SharePoint, suscripciones a LLMs, bases de datos vectoriales, un proveedor de RAG. Pero no presupuestan para la lógica descriptiva y el conocimiento, quizás porque el problema no se resuelve con una sola herramienta y resulta demasiado amorfo en un mundo centrado en los datos y en Agile. El vocabulario controlado se recorta a menos que genere valor inmediato para la base de datos. Con frecuencia, la taxonomía se recorta porque no hace una buena demostración y su valor reside en la fontanería, entregando su impacto con el tiempo. El resultado es una pila siempre creciente de documentos con un chatbot pegado en la entrada, respondiendo preguntas con confianza en la voz de un sistema que no tiene referencia para ninguno de los términos dispersos a lo largo de los documentos.

Entonces el sistema alucina. Se contradice a sí mismo entre sesiones. Presenta como vigente una política desactualizada y muestra mensajes privados de Outlook de empleados cuyas cuentas fueron desactivadas. He visto esto en la vida real más veces de las que puedo contar. Y la percepción es peor cuando una herramienta de proveedor es responsable de revelar chats y transacciones de correo privadas. Habitualmente, el patrocinador ejecutivo pregunta por qué la inversión no ha funcionado, ignorando la evidencia de los errores, como la exposición de comunicaciones privadas. Trabajo de un día cualquiera,

cuando los burndown charts y la velocidad marcan el precedente de las formas de trabajar y los sistemas de recompensa.

Los ejecutivos ignoran el hecho de que el sistema nunca fue anclado en el conocimiento. Nunca lo fue porque anclar requiere el trabajo lento, costoso y disciplinado de la organización del conocimiento — trabajo que no cabe en un sprint, que no hace una buena demostración y que no genera un gráfico de velocidad. La organización elige eludir la inversión en el aparato de conocimiento descriptivo porque las líneas son demasiado difusas para definirse claramente según el vocabulario de Agile. Porque el conocimiento es infraestructura, no un producto ni una plataforma. El conocimiento es la gasolina del depósito, el combustible del significado, el material que justifica el razonamiento lógico.

El pensamiento de diseño le da al trabajo del conocimiento un hogar. Es cómodo con la ambigüedad. Trata el descubrimiento del problema como un igual a la resolución del problema. Desde luego no exige un incremento entregable cada catorce días. En cambio, el pensamiento de diseño exige que entiendas lo que estás construyendo antes de construirlo — y que sigas revisando esa comprensión a medida que el trabajo avanza. La biblioteconomía y la ciencia de la información, con más de un siglo de estándares para la organización del conocimiento, proporcionan los métodos mientras el pensamiento de diseño proporciona la epistemología. Juntos ofrecen lo que Agile nunca pudo — un marco adecuado a la naturaleza del conocimiento mismo.

¿Un sprint de dos semanas para entregar qué? Si estás construyendo sistemas de conocimiento, dos semanas no producirán nada coherente. El conocimiento no se construye rápido y se moldea a la fuerza. Se construye con cuidado, con procedencia, gobernanza y paciencia. El conocimiento no es un producto que pueda manufacturarse y no — no se puede comprar conocimiento estándar que encaje perfectamente con un dominio u organización. El conocimiento es matizado, específico a cada organización, dominio y persona.

En el espíritu del argot del capital de riesgo: el conocimiento es el foso, y ese foso no es Ágil.

---

*Jessica Talisman, MLS* es Ingeniera Semántica, Arquitecta de Información y estratega de infraestructuras de conocimiento con más de 25 años de experiencia. Es fundadora del Ontology Pipeline™ y publica *Intentional Arrangement* en Substack.