Malorean Corp.

Una novela para identificarse

Un panorama para comprender

Desarrolladores, IA y pequeña empresa : una historia (casi) real de transformación

Lo que viven realmente los equipos de un SaaS ERP frente a la IA hoy



LAURENT DESÉCHALLIERS

Malorean Corp

> *Hasta hace poco, convertir una idea en un prototipo podía llevar semanas. Hoy, una persona que cuenta con la ayuda de la inteligencia artificial puede conseguirlo en pocos días. El cambio es espectacular. Pero la verdadera transformación podría producirse en otro lugar: en nuestra forma de trabajar juntos.*

Laurent Deséchalliers

Nota sobre la traducción

Esta edición es una traducción automática, producida por una inteligencia artificial a partir del texto francés.
No ha sido revisada por un traductor humano. Las ideas, la estructura y los ejemplos son fieles al original; en cambio, algunos giros pueden sonar torpes, y ciertos pasajes conservarán la huella de la lengua de partida.
Es el precio que aceptamos para que esta obra exista en varios idiomas y no solo en francés. En caso de duda sobre una formulación, la edición francesa es la referencia.

1

Introducción

Lo que más me impresiona de la IA no es que sepa escribir código. Es la velocidad con la que hace posible algo que, hasta hace muy poco, habría requerido días o semanas de trabajo.

Una idea puede convertirse en un prototipo antes incluso de que termine la discusión sobre su viabilidad. Un desarrollador puede explorar una posibilidad sin esperar a que todo un equipo esté disponible. Alguien que no sabe programar puede llegar a construir una primera versión de lo que imagina. Nada de esto significa que un prototipo sea un producto fiable ni que la complejidad haya desaparecido. Pero el mero hecho de poder probar algo ya cambia nuestra percepción de lo que parecía posible.

Y nuestras empresas no fueron concebidas para un mundo en el que desarrollar una primera solución pudiera hacerse tan deprisa.

Se han organizado en torno a las competencias de las personas, el tiempo disponible, los relevos entre equipos y las decisiones necesarias para sacar adelante el

trabajo. Todos esos mecanismos tienen su razón de ser. Sin embargo, cuando cambian las capacidades de las personas, empiezan a surgir nuevas preguntas. ¿Sigue teniendo sentido trabajar en el mismo orden? ¿Esperar las mismas aprobaciones? ¿Reservar determinadas actividades a las mismas profesiones?

Estas preguntas no afectan solo a los desarrolladores. Los Product Managers, los diseñadores, los comerciales, los responsables de equipos y muchos otros también están descubriendo nuevas formas de trabajar. Algunos pueden hacer directamente parte de lo que antes pedían a sus compañeros. Otros ven cómo cambia la naturaleza de su actividad. Lo que parecía una simple cuestión de herramientas empieza a afectar a nuestra manera de colaborar y tomar decisiones.

Sería tentador anunciar una revolución cuyo desenlace ya conocemos. No creo que hayamos llegado a ese punto. Algunas empresas están probando nuevas formas de repartir el trabajo y las responsabilidades. Otras avanzan con más cautela, a veces por buenas razones. Hay prácticas prometedoras, pero todavía nos falta perspectiva para saber cuáles resistirán las exigencias del día a día. Y una solución que funciona en un equipo pequeño no tiene por qué ser adecuada para una

empresa más grande, más regulada o con una trayectoria más larga.

Por eso no he querido escribir un manual sobre cómo deben organizarse las empresas con la IA. Este libro es un intento de comprender qué está cambiando realmente: qué permiten hacer las herramientas, cómo reaccionan las empresas, qué ganan, qué pueden perder y qué cuestiones tendrán que resolver.

Para ello he elegido dos caminos que se cruzan.

El primero es la observación. Examinaremos empresas reales, prácticas que se han hecho públicas, estudios y experiencias. Intentaré distinguir lo que está demostrado de lo que sigue siendo una hipótesis, sin confundir un anuncio atractivo con un resultado probado.

El segundo es una historia. Seguiremos a Malorean Corp., una empresa ficticia cuyos empleados irán descubriendo cómo estas nuevas posibilidades transforman su trabajo. No tienen un plan para reinventar su organización. Tienen clientes, costumbres, urgencias, convicciones y, a veces, desacuerdos. La ficción no demuestra nada; nos permite observar estas transformaciones desde la perspectiva de quienes las viven.

Necesitamos ambas miradas. Los ejemplos reales nos ayudan a no confundir nuestras intuiciones con hechos. La historia nos recuerda que los cambios organizativos no ocurren solo en los organigramas: afectan a la confianza, las competencias, las responsabilidades y el lugar que ocupa cada persona.

En el fondo, hay una pregunta que recorrerá las páginas que siguen: **¿qué ocurre en una empresa cuando quienes trabajan en ella descubren que pueden hacer mucho más de lo que hasta ahora les permitía su profesión?**

Sobre el autor

Llevo varias décadas trabajando en el sector del software y actualmente soy Product Manager. Mi trayectoria me ha llevado del desarrollo y la arquitectura de sistemas a la organización de equipos y, después, a la gestión de producto, sus operaciones y su marketing.

Esta experiencia me sitúa en una posición privilegiada para observar la evolución de la inteligencia artificial y sus efectos en nuestro trabajo. Lo veo a diario en mi actividad profesional, pero también en las conversaciones con mis compañeros Product Managers. Compartimos experiencias, descubrimientos e inquietudes. Estas charlas me llevan a menudo a mirar más allá de las herramientas: ¿qué está cambiando realmente en nuestras profesiones, en nuestras decisiones y en nuestra forma de trabajar juntos?

A medida que observábamos estos cambios y conversábamos sobre ellos, surgían nuevas preguntas. Sentí la necesidad de tomar distancia, contrastar nuestras impresiones con lo que ocurre en otras empresas y compartir esa reflexión. Así nació este libro.

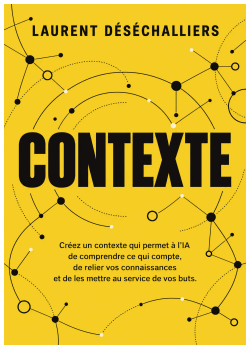
La escritura forma parte de mi trayectoria desde hace mucho tiempo. He colaborado con la prensa especializada y trabajado como revisor técnico para M2 Édition y Eyrolles. Hoy escribo en **malorean.net** sobre gestión de producto, IA, métodos de trabajo y sistemas de información.

También colaboro con **Proposition Marketing**, una revista independiente dedicada al marketing estratégico y operativo. Escribir para lectores del ámbito del marketing y de la dirección de empresas me anima a explicar las cuestiones de producto de forma accesible, sin dar por sentado que todos compartimos el mismo vocabulario.

Laurent Deséchalliers

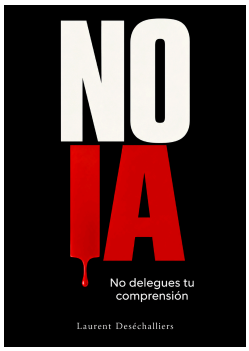
- **laurent@malorean.net**
- **https://malorean.net**

Del mismo autor



Le Contexte — Cómo construir un contexto que permita a la IA comprender lo que importa, conectar tus conocimientos y capitalizar lo que aprendes al servicio de tus objetivos.

Publicación: octubre de 2026.



NO IA — No delegues tu comprensión. Métodos sin IA para cuidar nuestra memoria, profundizar nuestra reflexión y ejercitar nuestro espíritu crítico. Prácticas para desarrollar nuestras propias capacidades y aprovechar mejor la IA sin delegarle nuestra comprensión.

Publicación prevista: finales de 2026.

Más información en **https://malorean.net**.

Agradecimientos

Quiero dar las gracias, en primer lugar, a los Product Managers con los que converso habitualmente. Contrastamos nuestros puntos de vista sobre la evolución de la gestión de producto, las profesiones relacionadas con el desarrollo de software y, en general, los efectos de la inteligencia artificial en el trabajo y la organización de las empresas. Estas conversaciones, a veces apasionadas, son una fuente constante de preguntas, replanteamientos e ideas nuevas. Algunas de esas reflexiones aparecen en las páginas de este libro. Gracias por vuestra curiosidad, por compartir vuestras experiencias y por vuestras ganas de debatir.

Y quiero dar las gracias también a Christine, mi mujer. Al terminar mi primer libro, **Le Contexte**, le prometí que haría una pausa. Al final, casi sin darme cuenta, empecé enseguida con este. Por suerte, esta segunda aventura nos ha quitado menos tardes y fines de semana que la anterior. Gracias, Christine, por tu paciencia y tu apoyo, incluso cuando mis reflexiones acaban convirtiéndose en otro libro.

Nota sobre los ejemplos

Las reflexiones y los ejemplos de este libro parten de más de treinta años de experiencia profesional en el sector informático. Proceden de situaciones vividas en proyectos y entornos profesionales diversos, y de numerosas conversaciones con product managers, CTO, engineering managers y otros profesionales de distintas empresas. Abarcan, por tanto, varias organizaciones y diferentes etapas de mi trayectoria.

Para poner estas situaciones al servicio del libro, las he modificado, generalizado o recompuesto. Los casos de uso, los datos numéricos y las métricas ilustran problemas recurrentes y lo que he aprendido de ellos. Son construcciones: ninguna describe a un cliente, un producto o una organización reales, ni refleja los datos, las prácticas o las posiciones de mi empleador.

Los análisis, métodos y opiniones que aparecen en este libro son personales y no comprometen a nadie más que a mí.

¿Se ha escrito este libro con IA?

Sí. En su totalidad, en colaboración con la inteligencia artificial.

Sé que hoy esta pregunta tiene una relevancia especial. Nos preguntamos si un artículo, una novela o un libro se ha escrito con IA. Hay incluso quien pasa los textos por detectores, como si fuera imprescindible separar lo que procede de una persona de lo que procede de una máquina.

Prefiero ser transparente: la IA ha participado en todas las etapas de este libro, desde la investigación documental hasta la redacción, pasando por la organización de las ideas, la comprobación de las referencias y la búsqueda de aspectos que se me pudieran haber escapado. Pero esta colaboración merece una explicación, porque por sí sola no aclara quién ha pensado el libro.

Durante mucho tiempo, para llevar a cabo un trabajo de esta envergadura, uno podía contar con personas que buscaran documentos, contrastaran ideas, revisaran textos, los estructuraran o les dieran forma. Hoy la IA me

permite disponer de parte de esas capacidades. Ejecuta tareas con rapidez, propone formulaciones, señala incoherencias y, a veces, plantea preguntas que no se me habían ocurrido. Me permite explorar mucho más de lo que podría abarcar por mi cuenta con el tiempo del que dispongo.

Para escribir este libro he utilizado un motor de Contexto basado en las tecnologías de ChatGPT, con Notion para conservar la información entre conversaciones. De este modo, las fuentes, las ideas, las decisiones y las versiones de trabajo se pueden guardar y recuperar de una conversación a otra. Ya presenté este enfoque en mi primer libro, **Le Contexte**. En él explico cómo un sistema de este tipo permite que la IA se convierta en una especie de supersecretario personal: no se limita a responder preguntas, sino que mantiene el hilo de los proyectos y me ayuda a impulsarlos a lo largo del tiempo.

Hay, sin embargo, una distinción esencial: **tener capacidad para ejecutar algo no significa asumir la responsabilidad de comprenderlo y elegir.** Las preguntas que guían este libro, la experiencia profesional que las alimenta, las interpretaciones, las decisiones y lo que finalmente decido defender o descartar son míos. La IA puede llevarme la contraria, cuestionar mis

planteamientos y ayudarme a expresar mejor una idea. No sustituye mi criterio ni mi responsabilidad como autor.

También distingo este libro de aquellas obras en las que la propia escritura constituye una parte fundamental de la creación. Cuando leemos a Shakespeare en una traducción de François-Victor Hugo, o a Cervantes o Cizia Zykë en su lengua original, la voz, el ritmo y el estilo importan tanto como lo que se cuenta. En una obra traducida, las decisiones del traductor también forman parte de la experiencia. El lugar que queramos dar a la IA en este tipo de literatura plantea otras preguntas que no pretendo resolver aquí.

Este libro incluye una parte de ficción, pero su principal ambición sigue siendo comprender una transformación técnica y organizativa. La forma debe resultar agradable y clara; ante todo está al servicio de las ideas, los hechos y las preguntas que quiero compartir.

La IA ha hecho posible este proyecto a una escala que probablemente no habría podido alcanzar por mis propios medios al margen de mi actividad profesional. Me permite desarrollar proyectos personales, como este libro, sin renunciar al esfuerzo de reflexión que exigen.

Y, al fin y al cabo, tiene cierta coherencia haber escrito un libro sobre las transformaciones que provoca la IA trabajando precisamente de esta manera.

Laurent Deséchalliers

Índice

Nota sobre la traducción 3

Introducción 4

Sobre el autor 8

Del mismo autor 10

Agradecimientos 11

Nota sobre los ejemplos 12

¿Se ha escrito este libro con IA? 13

Malorean Corp. 19

1 Malorean Corp. — Un día fuera de lo común 20

2 La IA ha multiplicado la capacidad de producción de los desarrolladores 29

3 Malorean Corp. — El día después 42

4 Están surgiendo nuevos modelos organizativos 59

5 Malorean Corp. — Lo que podemos hacer y lo que debemos hacer 73

6 Los roles se difuminan, las responsabilidades permanecen 91

7 Malorean Corp. — Lo que Alice sabía sin haberlo escrito nunca 106

8 Más autonomía requiere reglas más claras, no menos 130

9 Malorean Corp. — Dos decisiones razonables 151

10 ¿Hará la IA desaparecer los niveles intermedios de las empresas? 171

11 Malorean Corp. — El tiempo recuperado 194

12 Conclusión 219

Malorean Corp.

1 — Malorean Corp. — Un día fuera de lo común

La sede de Malorean Corp. estaba en Le Havre. Pero desde la pandemia, aquella dirección decía muy poco sobre dónde se hacía realmente el trabajo. Los empleados estaban repartidos por toda Francia. Se reunían ante una cámara, hablaban mientras compartían pantalla y, casi siempre, nadie necesitaba estar en la misma sala para sacar adelante un asunto.

Malorean Corp. desarrollaba un ERP en línea: uno de esos programas que acaban presentes en prácticamente todas las operaciones de una empresa. Un cliente podía ser una microempresa, con unos pocos empleados y necesidades muy concretas. Otro podía ser un gran grupo, con procesos complejos, requisitos particulares y poca paciencia para las imprecisiones.

La mayoría de sus clientes eran franceses. También había algunos usuarios en Alemania y España, pero sobre todo en filiales de grupos que ya eran clientes en Francia.

Hubert Félix Duca, el director general, aspiraba a algo más. Le habría gustado que Malorean Corp. lograra expandirse de verdad por Europa.

El problema era que un ERP no cruza fronteras con la misma facilidad que una página web corporativa. Las normas cambian de un país a otro. También los procedimientos de nóminas, las obligaciones administrativas y la manera de trabajar de las empresas. Hubert conocía bien esas limitaciones. Venía del mundo de la gestión, la contabilidad y las finanzas, no del desarrollo de software. Para él, una función solo tenía valor si los clientes podían utilizarla sin meterse en problemas.

Dentro de la empresa, las funciones estaban bastante claras.

Dos equipos de cinco desarrolladores se ocupaban de mejorar el programa. Alice y Bob, los dos Product Managers, trabajaban con ellos para comprender las necesidades, elegir las mejoras más útiles y orientar el producto. Alice había sido desarrolladora: podía participar en los debates técnicos sin perderse. Bob había estudiado en una escuela de negocios y abordaba las cuestiones desde otra perspectiva. Tenían trayectorias distintas, pero trabajaban muy bien juntos.

Un equipo comercial buscaba nuevos clientes y cuidaba las relaciones existentes, con la ayuda de una persona de marketing. Cinco especialistas intervenían en la preventa

y después durante la implantación, de forma presencial o a distancia. No se limitaban a instalar el ERP: ayudaban a las empresas a concretar sus necesidades, adaptar sus procesos y, a menudo, seguían acompañándolas durante los seis meses posteriores a la venta.

El equipo de soporte tenía doce personas. Siete atendían las cuentas pequeñas y cinco, las grandes. Muchas veces eran los primeros en enterarse de que alguna función obligaba a un cliente a buscar soluciones alternativas en lugar de utilizar el programa con normalidad. Dos juristas velaban por los asuntos de Malorean Corp., pero también aportaban sus conocimientos al diseño del producto, sobre todo cuando una mejora afectaba a normas delicadas, como las relativas a las nóminas.

Todo esto mantenía la empresa en marcha. No de forma perfecta, por supuesto. Un programa implantado en tantos clientes acumula peticiones, excepciones, costumbres difíciles de cambiar y funciones que se mantienen porque nunca se encuentra tiempo para rehacerlas. Pero cada uno sabía más o menos lo que tenía que hacer.

Entonces ChatGPT empezó a aparecer en las conversaciones.

Al principio no había ningún gran programa de transformación. Dos de las personas más curiosas lo habían probado unos meses antes: Julie Morel, del equipo comercial, y Lucas Martin, de desarrollo. Habían empezado usándolo para sus propias tareas y después contaron lo que conseguían hacer.

En el caso de Julie, las ventajas eran evidentes. Preparaba un correo delicado, pedía una redacción más clara, organizaba las notas antes de una reunión, convertía un conjunto de ideas en un esquema para una presentación. A veces también preparaba el contenido de algunas diapositivas. ChatGPT no decidía lo que había que decirle al cliente. Ella le proporcionaba la información, revisaba las propuestas, corregía las frases y enviaba únicamente lo que estaba dispuesta a respaldar.

No pretendía que la herramienta conociera mejor a sus clientes que ella. Simplemente le alegraba no tener que pasar media hora dando forma a algo que ya sabía.

Philippe Garnier, su compañero, observaba todo aquello desde cierta distancia. Era un comercial experimentado, acostumbrado a tratar con grandes cuentas. Conocía a las personas que había detrás de los contratos, los desacuerdos que no aparecían en ninguna acta y las

pequeñas frases que podían cambiar el sentido de una reunión.

—Deberías probarlo para tus correos —le dijo Julie una mañana—. Aunque solo sea para darles forma.

—Mis correos ya sé escribirlos.

No lo dijo con mala intención. Simplemente no entendía por qué iba a confiar a una herramienta parte de su trabajo cuando, a su juicio, su valor residía precisamente en conocer a los clientes y saber hablar con ellos.

Julie no insistió. Tenía una reunión que preparar.

Lucas utilizaba ChatGPT de una manera muy distinta. Era un desarrollador experimentado que vivía en Lyon y no buscaba únicamente ahorrar unos minutos en tareas repetitivas. Dedicaba mucho tiempo a experimentar, a plantear preguntas cada vez más difíciles, a volver a empezar cuando una respuesta no le convencía. Mientras Julie quería, sobre todo, obtener un resultado útil, Lucas trataba de descubrir hasta dónde se podía llegar.

Las impresiones de aquellos primeros usuarios acabaron llegando a Hubert. Por unos veinte euros por persona al mes, las cuentas parecían sencillas: bastaba con que cada empleado ahorrase un poco de tiempo en las tareas más tediosas para que la inversión mereciera la pena. Así

que Malorean Corp. decidió poner ChatGPT a disposición de todos.

No hubo una gran revolución anunciada con presentaciones por todo lo alto. Cada uno podía usarlo cuando le resultara útil. Algunos lo abrían todos los días, otros apenas lo tocaban y Philippe no lo usaba en absoluto. Para la mayoría, la IA seguía siendo un asistente: le explicabas lo que querías, te proponía algo, lo comprobabas y después decidías.

La vida de la empresa continuaba.

El miércoles, durante una conversación entre los equipos de Producto y Desarrollo, volvió a salir un asunto antiguo: la gestión de las vacaciones y los permisos.

A nadie le gustaba demasiado aquel módulo. Era viejo, rígido y ofrecía pocas opciones. Funcionaba lo suficiente para que siguieran utilizándolo, pero no tan bien como para dejar de recibir quejas.

Un empleado que quisiera solicitar días libres tenía que hacerlo desde el ordenador de trabajo. No podía gestionarlo fácilmente desde el móvil un domingo por la noche, mientras pensaba en las vacaciones. Y el cálculo de las ausencias tenía un defecto aún más incómodo: el módulo no contabilizaba bien los festivos. Si alguien

pedía de lunes a viernes y el miércoles era festivo, podía descontarle cinco días en lugar de cuatro.

El soporte conocía el problema. Los comerciales también. Para algunos clientes, el módulo se había convertido en una fuente habitual de frustración. Se conservaba porque ya existía, no porque respondiera realmente a lo que necesitaban.

—Habría que rehacerlo bien —resumió alguien.

Nadie discrepaba. Pero «rehacerlo bien» significaba revisar la interfaz, corregir las reglas de cálculo, facilitar su uso desde el móvil y sustituir parte de lo que había quedado obsoleto.

Calculaban unos dos meses de trabajo. Había otras prioridades. Una vez más, el asunto acabó en la lista de cosas pendientes.

El viernes por la tarde, al terminar una videollamada, Lucas pidió unos minutos.

—Tengo algo que enseñaros.

No había adelantado de qué se trataba. Nadie sabía que llevaba trabajando en el módulo de vacaciones desde la conversación del miércoles.

Compartió su pantalla. Apareció una interfaz clara y moderna, con la información esencial donde uno esperaba encontrarla.

Lucas mostró cómo presentar una solicitud de vacaciones. Después redujo la ventana al ancho de un teléfono. La interfaz se adaptó. Se podía seguir adelante, elegir las fechas y confirmar la solicitud sin necesidad de volver al ordenador.

Pasó a un segundo ejemplo: una semana de lunes a viernes, con un festivo el miércoles.

El sistema descontó cuatro días.

Cambió las fechas. El resultado se recalculó. Volvió a la vista móvil, pasó por varias pantallas y siguió haciendo pruebas. No eran maquetas estáticas una detrás de otra: los formularios respondían, se hacían los cálculos y el proceso funcionaba.

Al principio, algunos compañeros hicieron preguntas normales: dónde estaba cierta opción, qué ocurría si se cambiaba una fecha. Después dejaron de preguntar.

En las pequeñas ventanas de la videollamada, las expresiones de los rostros cambiaron.

El miércoles habían hablado de dos meses de trabajo. El viernes, Lucas les estaba mostrando una primera versión

funcional. No les había pedido ayuda. No había avisado a nadie. Y, de momento, no explicaba cómo lo había conseguido.

—Espera —dijo por fin alguien—. ¿Ese es el módulo del que hablamos anteayer?

Lucas confirmó que sí.

Se hizo el silencio.

En las pantallas, todos miraban lo mismo: un trabajo que creían que tendrían que planificar durante semanas y que había aparecido en dos días.

Nadie entendía cómo había sido posible.

2 — La IA ha multiplicado la capacidad de producción de los desarrolladores

Durante décadas, buena parte de la organización del desarrollo de software se ha construido en torno a una limitación muy sencilla: **desarrollar software es caro**. Convertir una idea en un producto exigía tiempo, competencias difíciles de encontrar y mucha coordinación. Había que definir el alcance de una funcionalidad, dividir el trabajo, priorizarlo, desarrollarla, probarla, revisarla y ponerla en producción. El número de desarrolladores disponibles, su experiencia y el tiempo que podían dedicar a un asunto limitaban directamente lo que una empresa podía emprender.

Esta escasez ha moldeado las organizaciones. Hemos aprendido a proteger el tiempo de los desarrolladores, preparar su trabajo con antelación, reducir las interrupciones, organizar los backlogs y debatir las prioridades mucho antes de empezar a construir. Buena parte de la forma moderna de gestionar productos nació en ese contexto: había que evitar dedicar unos recursos de ingeniería costosos a una mala idea.

La llegada de los asistentes de programación y, más tarde, de agentes capaces de ejecutar tareas más amplias está empezando a cambiar esa ecuación.

El cambio ya no consiste solo en mejorar el autocompletado

Las primeras herramientas de IA para desarrolladores podían entenderse como una nueva generación de instrumentos de productividad: completar una función, explicar un fragmento de código, escribir una prueba o proponer una refactorización.

La siguiente generación cambia radicalmente la naturaleza del trabajo. Un agente puede recibir una tarea, explorar un repositorio, modificar varios archivos, ejecutar pruebas, corregir ciertos errores y entregar un cambio listo para revisión. Un desarrollador puede delegar así buena parte de la ejecución y, sobre todo, poner en marcha y hacer avanzar más trabajo que antes.

La consecuencia no es únicamente que «el desarrollador programe más deprisa».

La cantidad de trabajo de desarrollo que una sola persona puede iniciar e impulsar aumenta de manera extraordinaria.

Algunas empresas están empezando a publicar resultados espectaculares. AMPECO, que describe su propia transición hacia un sistema de ingeniería AI--native, afirma haber cuadruplicado su velocidad de desarrollo y reducido un 50 % los errores por historia de usuario. Estas cifras no deben tratarse como una media del mercado: son los resultados declarados por una empresa en un contexto particular. Pero permiten hacerse una idea de la magnitud del cambio que se busca: ya no se trata solo de ahorrar unos minutos al escribir una función, sino de aumentar considerablemente la capacidad de ejecución de un equipo. [1]

Este resultado espectacular, declarado por una empresa, no es el único indicio disponible. Yo mismo he vivido algo parecido con Claude Code. En cuatro días y con más de 180 commits, construí una primera versión de un producto con pantallas, API y pruebas. Con mis métodos anteriores, calculo que habría necesitado entre uno y dos meses. Conté la experiencia en **Malorean.net** en julio de 2026. Por supuesto, no permite generalizar una mejora de productividad: muestra lo que un PM con experiencia previa en desarrollo puede llegar a construir, pero

también las dificultades para controlar la calidad, el mantenimiento y el código generado. [7]

Esto cambia la escala de la pregunta.

Si la IA solo permitiera ahorrar unos pocos puntos porcentuales en la escritura de código, la organización podría seguir funcionando igual y absorber esa mejora. Cuando la capacidad de producción crece mucho más deprisa, las demás limitaciones adquieren una importancia relativa mayor.

Acelerar la programación no basta para acelerar la empresa

Un equipo de software no es una cadena cuyo único eslabón sea escribir código.

Antes de construir hay que comprender un problema, decidir si merece una inversión, elegir una dirección y aclarar ciertas limitaciones. Después, hay que revisar, probar, asegurar, desplegar, observar y mantener lo construido. Y a lo largo de todo el proceso hay que tomar decisiones.

Si solo uno de esos eslabones se acelera mucho, se limita a trasladar más trabajo a los demás.

El código puede producirse más deprisa sin que la comprensión de los clientes avance al mismo ritmo. Los

prototipos pueden multiplicarse sin que aumente la capacidad de decidir cuáles merecen seguir adelante. Pueden llegar más cambios a revisión sin que crezca en la misma proporción la atención humana disponible para examinarlos.

El problema no es, por tanto, que la IA vuelva de pronto inútiles las demás actividades.

Es que **desplaza los cuellos de botella**.

Lo que antes quedaba oculto por la lentitud de la ejecución se vuelve más visible: la calidad del problema elegido, la claridad de la intención, las decisiones, la validación, la coherencia y la capacidad de aprender después de entregar el producto.

Producir más no significa crear más valor

Esta distinción se vuelve fundamental.

Cuando producir software era difícil, la cantidad entregada podía parecer una buena aproximación a la capacidad de una organización para crear valor. Esa simplificación ya era imperfecta. Con la IA se vuelve peligrosa.

Ahora un equipo puede generar más código, más pull requests, más prototipos y más funcionalidades sin mejorar el producto. Incluso puede empeorar su

rendimiento global si ese volumen adicional aumenta los defectos, el trabajo que hay que rehacer, los costes o la complejidad del mantenimiento.

Dicho de otro modo:

saber construir no equivale a saber qué merece la pena construir.

Por eso las métricas de actividad resultan todavía menos suficientes. El número de cambios, la cantidad de código o el mero uso de una herramienta de IA no indican si un equipo ha creado más valor.

Canva ofrece un ejemplo interesante de este cambio hacia el impacto: la empresa desarrolló un sistema que permite a sus equipos de Producto, Ingeniería y Datos estimar el impacto comercial de sus experimentos y relacionar más directamente las decisiones locales con sus consecuencias para el negocio. [2]

La pregunta cambia entonces por completo.

Antes:

¿cuánto podemos construir?

Ahora:

de todo lo que ya podemos construir, ¿qué merece realmente la pena?

> ***El problema de la productividad del software tiene más de cuarenta años***
>
> *En 1986, Fred Brooks presentó* ***No Silver Bullet****, publicado en* ***IEEE Computer*** *en 1987. Distinguía las dificultades* ***accidentales*** *de producir software —los lenguajes, las herramientas, la traducción de un diseño a código— de su complejidad* ***esencial****: comprender, diseñar, articular y hacer evolucionar un sistema. Según él, ningún avance técnico o de gestión, por sí solo, permitía esperar que la productividad, la fiabilidad y la sencillez del desarrollo se multiplicasen por diez en una década. [3]*
>
> *El ensayo se incorporó en 1995 a la edición conmemorativa de* ***The Mythical Man-Month****, publicado originalmente en 1975. [4] La IA generativa podría cambiar la escala de la ejecución, pero no garantiza que el problema elegido sea el adecuado. Esta es una interpretación actual de las ideas de Brooks,* ***no una predicción suya sobre los agentes de IA****.*

El nuevo cuello de botella es el criterio

A medida que ejecutar resulta menos costoso, otros recursos adquieren más valor en comparación: conocer al

cliente, comprender el problema, disponer de contexto estratégico, saber expresar una intención clara, evaluar los riesgos, tener criterio de producto, saber decir que no y asumir la responsabilidad de las decisiones.

En un artículo publicado en **Malorean.net** en marzo de 2026 defendí esta idea: conforme las herramientas técnicas resultan más accesibles, la ventaja competitiva se desplaza hacia la comprensión del contexto: el trabajo de las empresas, los usuarios, los clientes, las limitaciones técnicas, la competencia y las regulaciones. Es un análisis estratégico, no un resultado experimental. Ayuda a precisar qué significa ejercer el criterio: entender suficientemente bien una situación para elegir lo correcto que hay que construir, y no limitarse a saber producirlo. [8]

La IA hace más accesible el «cómo».

Pero no hace evidente el «por qué» de forma automática.

Permite producir una solución antes de que la organización haya terminado de comprender el problema. Permite responder rápidamente a una petición de un cliente que la empresa debería haber rechazado por no encajar en su mercado. Permite multiplicar las variantes de una funcionalidad sin saber cuál consigue realmente un mejor resultado.

Cuanto más fácil resulta construir, más importante es la disciplina de elegir qué no se va a construir.

Ramp ilustra bien este principio: su CPO insiste en la necesidad de definir con claridad los perfiles de cliente ideal (ICP) prioritarios, los problemas que la empresa quiere resolver especialmente bien y aceptar que hay que rechazar algunas peticiones —incluso lucrativas— cuando desvían el producto hacia otra dirección. [5]

En julio de 2026, Marty Cagan llamó a este desfase la **paradoja de la productividad de la IA**. Su diagnóstico es que construir más deprisa no soluciona el problema de las ideas que nunca deberían haberse desarrollado. Distingue así lo que construimos para aprender de lo que entregamos para lograr un resultado. Es el análisis de un profesional de la gestión de producto, no una medición que demuestre que todas las empresas se ven afectadas por igual. [6]

La paradoja de la IA es, por tanto, esta:

aumenta enormemente nuestra capacidad de actuar y, por eso mismo, hace todavía más importante la elección de nuestras acciones.

Una transformación del sistema de trabajo, no solo de las herramientas

Se podría responder a esta evolución dando a cada desarrollador un asistente de programación y pidiendo a los equipos que sigan trabajando como siempre.

Eso produce mejoras locales.

Pero a medida que crecen esos beneficios, la organización se topa con sus otros límites: su capacidad para comprender, decidir, coordinar, verificar y absorber todo lo que se produce.

En ese punto, la cuestión deja de ser simplemente la adopción de una herramienta.

Se convierte en una cuestión organizativa.

Una organización diseñada cuando los recursos de ingeniería eran los más escasos no tiene por qué seguir siendo óptima cuando esa escasez se reduce.

La pregunta es inevitable:

¿qué ocurre con una organización diseñada para proteger un recurso escaso —la capacidad de ingeniería— cuando ese recurso se vuelve mucho más abundante?

Bibliografía

[1] How we built an AI-native engineering system — AMPECO

- URL: **https://www.ampeco.com/blog/how-we-built--an-ai-native-engineering-system/**
- Experiencia de una organización de ingeniería reconstruida alrededor de agentes, que comunica un notable aumento de capacidad.

[2] Measuring Commercial Impact at Scale — Canva

- URL: **https://www.canva.dev/blog/engineering/measuring-commerical-impact-at-scale/**
- Explica cómo Canva relaciona los experimentos de sus equipos de Producto e Ingeniería con una estimación accesible de su impacto comercial.

[3] No Silver Bullet: Essence and Accidents of Software Engineering — Fred Brooks

- URL: **https://www.cgl.ucsf.edu/Outreach/pc204/NoSilverBullet**
- Ensayo presentado en 1986 y publicado en 1987, que diferencia la complejidad esencial de las dificultades accidentales del desarrollo de software.

[4] The Mythical Man-Month, edición conmemorativa, capítulo 16 — Fred Brooks

- URL: **https://www.oreilly.com/library/view/mythical--man-month-the/0201835959/ch16.xhtml**
- Índice editorial de la edición de 1995 que confirma la incorporación posterior de No Silver Bullet; solo se ofrece una vista previa del capítulo.

[5] 5 insights from Ramp's CPO on building for hypergrowth — Ramp

- URL: **https://ramp.com/blog/5-insights-from--ramps-cpo-reddit-ama**
- El CPO de Ramp destaca el foco en los ICP prioritarios y la necesidad de rechazar las peticiones que dispersan los esfuerzos.

[6] The AI Productivity Paradox — Marty Cagan, Silicon Valley Product Group (2026)

- URL: **https://www.svpg.com/the-ai-productivity--paradox/**
- Análisis de julio de 2026 sobre la diferencia entre desarrollar más deprisa gracias a la IA y crear resultados reales; una opinión profesional, no una demostración causal.

[7] Quatre jours de vibe coding dans la peau d'un PM rouillé — Laurent Deséchalliers, **Malorean.net** (2026)

- URL: **https://malorean.net/articles/2026-07-28--quatre-jours-de-vibe-coding-dans-la-peau-d-un--pm-rouille.es.html**
- Experiencia personal con Claude Code: velocidad de desarrollo de un producto, dificultades técnicas, pruebas y control de calidad; ni estudio controlado ni prueba generalizable.

[8] Dans le logiciel, l'avantage ne sera plus la technologie. Ce sera la compréhension du contexte — Laurent Deséchalliers, **Malorean.net** (2026)

- URL: **https://malorean.net/articles/2026-03-23--dans-le-logiciel-lavantage-ne-sera-plus-la--technologie-ce-sera-la-comprehension-du--contexte.es.html**
- Ensayo estratégico del autor sobre el traslado de la escasez hacia la comprensión del contexto del cliente, su actividad, las regulaciones y la tecnología; no es una demostración cuantitativa.

3 — Malorean Corp. — El día después

—He utilizado la IA.

Lucas lo había dicho casi con naturalidad, como si bastara para explicar lo que acababan de ver.

Seguía compartiendo pantalla. El nuevo módulo de vacaciones ocupaba casi toda la ventana. Unos minutos antes había mostrado cómo solicitar días libres desde el móvil y cómo el programa tenía en cuenta los festivos al calcularlos.

—Espera —intervino Alice—. Cuando dices que has utilizado la IA, ¿quieres decir que te ha ayudado a escribir el código?

—Sí. Bueno, no solo eso. A diseñar el módulo, entender el código que ya existía, desarrollar las pantallas, hacer los cálculos, escribir las pruebas... Para casi todo, en realidad.

—¿Y lo has hecho desde el miércoles?

—Desde el miércoles por la tarde. He terminado los últimos detalles esta mañana.

Alice observó las caras en la pantalla. Algunos sonreían. Otros parecían intentar entender qué se les había escapado.

—Pero no tenías especificaciones —señaló.

—No. Lo hablamos el miércoles. Conocía el módulo y sus problemas. Primero le pedí a la IA que analizara lo que había. Después fuimos avanzando poco a poco.

—¿Y funciona todo?

Lucas soltó una risita.

—Todo, no lo sé. Lo que os he enseñado, sí. Seguro que quedan cosas por comprobar.

Thomas, que trabajaba en el mismo equipo, intervino por primera vez.

—¿Y cómo es el código por dentro?

—Puedes echarle un vistazo. He hecho bastantes pruebas.

—Lo miraré —respondió Thomas.

No parecía especialmente impresionado. O quizá lo estaba, pero no quería demostrarlo.

Hubert Félix Duca había permanecido callado hasta entonces.

Había asistido a suficientes demostraciones para saber que un programa que funciona durante diez minutos no está necesariamente preparado para que lo utilicen cientos de empresas. Conocía las excepciones, las

necesidades particulares de los grandes clientes y los problemas que solo aparecen cuando se implanta algo.
Pero también recordaba lo que había oído el miércoles.
Dos meses.
Y lo que tenía delante se parecía mucho a lo que habían imaginado.
—Lucas, ¿puedes repetirme cuánto tiempo has dedicado a esto? —preguntó.
—Dos días, más o menos.
Hubert se inclinó ligeramente hacia la pantalla.
—Vale.
No se le ocurrió qué más decir.
La reunión terminó unos minutos después. Cada uno volvió a sus ocupaciones, aunque nadie volvió a hablar de verdad de las tareas de aquella tarde.

El sábado por la mañana, Alice recordó la demostración mientras se preparaba un café.
Lo que la inquietaba no era solo la velocidad de Lucas.
Era que no había necesitado a nadie.
Conocía bien su trabajo de Product Manager. Antes de iniciar un desarrollo, trataba de entender el problema, contrastaba peticiones, hablaba con los clientes y

trabajaba con los desarrolladores para encontrar una solución razonable.

En un asunto como el módulo de vacaciones seguramente habría organizado varias conversaciones. Habría ayudado a aclarar las reglas, resolver los casos particulares y decidir qué debía desarrollarse primero.

Lucas no había hecho nada de eso.

Había escuchado una conversación, tomado la iniciativa y construido una solución.

En parte lo admiraba. Pero también se preguntaba qué pasaría si todos los desarrolladores empezaban a trabajar así.

Sin embargo, durante los últimos meses su propio día a día había cambiado considerablemente.

Al principio, Alice utilizaba la IA para resumir actas de reuniones o reformular documentos. Después empezó a confiarle tareas más importantes. Había automatizado parte del tratamiento de los comentarios de los clientes, la preparación de algunos informes de síntesis, la agrupación de peticiones similares y varias tareas administrativas que antes le ocupaban horas.

Ahora a veces terminaba en unos minutos un trabajo que antes le llevaba buena parte de la mañana.

No echaba de menos ninguna de esas tareas.

Al contrario.

Por fin tenía más tiempo para hablar con los clientes, comprender los problemas que se escondían detrás de sus peticiones y pensar qué merecía realmente la pena construir.

La IA no había hecho que su trabajo fuera menos interesante.

Le había quitado de encima parte de lo que le impedía hacerlo bien.

Entonces, ¿por qué los desarrolladores tendrían que seguir trabajando exactamente igual que antes?

Alice dejó la taza.

La pregunta le parecía evidente.

La respuesta, mucho menos.

Hubert también había pasado buena parte del fin de semana pensando.

Su empresa siempre había tenido más ideas que capacidad para hacerlas realidad. Había funcionalidades que llevaban meses esperando. Algunas mejoras solicitadas por los clientes se aplazaban

sistemáticamente. Y estaba aquel proyecto de expansión europea, que aún exigía mucho trabajo en el programa.

Conocía el obstáculo de sobra: no tenían tiempo suficiente para hacerlo todo.

Quizá el viernes Lucas había demostrado que ese obstáculo ya no era tan insalvable.

Si un proyecto estimado en dos meses podía producir una primera versión en dos días, ¿cuántos proyectos más se volvían de pronto posibles?

En Malorean Corp. trabajaban diez desarrolladores.

Hubert intentó imaginar qué pasaría si todos avanzaran igual de deprisa.

Después intentó imaginar a diez desarrolladores poniendo en marcha cada uno sus propias funcionalidades, sin coordinación.

La segunda imagen le gustó bastante menos.

Pensó en llamar a Alice el sábado mismo. Se contuvo.

El domingo por la noche se limitó a enviarle un mensaje.

¿Tienes un momento mañana por la mañana? Me gustaría que volviéramos a hablar de lo del viernes.

La respuesta llegó unos minutos después.

Sí. A mí también.

El lunes a las nueve de la mañana, Alice y Hubert se conectaron por videollamada.

—Bueno, ¿qué? —preguntó Hubert.

—¿Qué de qué?

—¿Has entendido lo que pasó el viernes?

Alice sonrió.

—Técnicamente, más o menos. Lucas utilizó la IA para desarrollar mucho más deprisa de lo que habíamos imaginado.

—¿Y lo demás?

—Ese es precisamente el problema.

Hubert asintió.

—Llevo todo el fin de semana dándole vueltas. Nos falta capacidad desde hace años. Rechazamos peticiones, aplazamos proyectos, nos pasamos la vida eligiendo lo que no vamos a hacer. Y ahora Lucas nos presenta en dos días algo que habíamos calculado que llevaría dos meses.

—Una primera versión —precisó Alice—. Todavía no sabemos si está lista para producción.

—Sí, ya lo sé. Pero aunque hagan falta otras dos semanas para terminarla, cambia muchas cosas.

Alice no podía discutirlo.

—Lo que me preocupa —continuó Hubert— es que haya hecho todo esto sin contar contigo.

Alice tardó un momento en responder.

—A mí también me ha molestado —admitió.

—¿Crees que deberíamos recordarle cómo trabajamos?

—Lo he pensado.

—¿Y?

—Y creo que sería un error.

Hubert frunció el ceño.

—Pero si empieza a decidir por su cuenta lo que desarrolla, podemos encontrarnos con unas cuantas sorpresas.

—Claro. Pero fíjate en lo que ha ocurrido. El miércoles dijimos que había que rehacer el módulo de vacaciones. Si hubiéramos seguido nuestro procedimiento habitual, yo habría empezado a recopilar necesidades, habríamos organizado reuniones, preparado el asunto, dividido el trabajo... Seguramente habríamos pasado varias semanas antes de enseñar nada.

—Eso no significa necesariamente perder el tiempo.

—No. Y tampoco digo que debamos dejar de entender las necesidades. Pero Lucas nos ha demostrado que podía avanzar sin esperar a todo eso.

—¿Así que propones dejar que los desarrolladores decidan por sí mismos?

—No, justo lo contrario.

Alice reflexionó unos segundos.

—Creo que estamos mezclando dos cosas: decidir qué debemos hacer y decidir cómo hacerlo.

—Las dos están relacionadas.

—Sí. Pero eso no significa que yo tenga que intervenir en cada decisión.

Se incorporó un poco en la silla.

—Mira, mi trabajo ha cambiado muchísimo estos últimos meses. Utilizo la IA todos los días. He automatizado un montón de cosas que antes hacía a mano. Dedico mucho menos tiempo a preparar documentos o a organizar la información.

—Ya lo sé, me lo has contado.

—Pero quizá no te das cuenta de cuánto ha cambiado realmente mi trabajo. Ahora tengo mucho más tiempo para lo que de verdad importa: comprender a los clientes, detectar problemas y decidir dónde invertir nuestra energía.

Hubert la escuchaba.

—Antes, una parte de mi trabajo consistía en hacer que los asuntos avanzaran dentro de la organización. Preparar los tickets, precisar las peticiones, responder preguntas, coordinar conversaciones. ¿Y si mañana los desarrolladores necesitaran mucho menos que yo hiciera todo eso?

—Tendrías más tiempo.

—Exacto. Y ellos también.

Hizo una pausa.

—Aceptamos que la IA transforme mi profesión. ¿Por qué querríamos que la de los desarrolladores se quedara igual?

Hubert se recostó en su silla.

—De acuerdo. Pero alguien tiene que seguir decidiendo qué construimos.

—Por supuesto. Y, sobre todo, por qué lo construimos.

—Entonces, ¿cómo te imaginas las cosas?

Alice vaciló.

No había preparado ninguna presentación. No tenía ningún modelo organizativo que proponer ni ningún proceso nuevo que implantar.

Solo tenía una intuición.

—Me pregunto si no deberíamos probar algo.

Hubert esperó a que continuara.

—Podríamos confiarle un asunto a Lucas. Un problema real de un cliente. Y darle mucha más libertad para resolverlo.

—¿Sin especificaciones?

—Sin darle una solución ya hecha. Le explicamos el problema, le facilitamos el contacto con quienes lo conocen y dejamos que investigue.

—¿Y tú?

—Sigo estando disponible. Le ayudo a entender las necesidades, cuestiono sus ideas y evito que construya algo solo porque puede hacerlo. Pero no le digo qué tiene que hacer a cada paso.

Hubert lo pensó.

—¿Quieres que haga parte del trabajo de Product Manager?

—Quizá. O simplemente que dejemos de repartir sistemáticamente el trabajo entre nosotros.

—¿Y si no funciona?

—Lo veremos.

Hubert sonrió.

—Como método, no parece muy sólido.

—Ya lo sé.

Alice también sonrió.

—Pero prefiero probar con un asunto limitado antes que pasar tres meses discutiendo una nueva organización.

Hubert no respondió enseguida.

Entonces pareció recordar algo.

—Precisamente tenemos un problema que algunos clientes del sector agroalimentario llevan semanas señalando.

Alice asintió. Sabía perfectamente a cuál se refería.

El ERP de Malorean Corp. gestionaba las existencias y las operaciones de fabricación, pero no permitía seguir correctamente los lotes de materias primas utilizados para fabricar los productos terminados.

En una fábrica de alimentos, cuando un proveedor notificaba que un lote de materias primas era defectuoso, había que poder identificar rápidamente los productos fabricados con él.

En ese momento, los clientes de Malorean Corp. tenían que reconstruir esa información por su cuenta.

—La semana pasada volvieron a comentármelo —explicó Hubert—. Un cliente me dijo que había tenido que pasar un día entero con su responsable de calidad para

localizar los productos afectados por una alerta de un proveedor.

—Sí. El soporte nos ha comunicado varios casos parecidos.

—¿Crees que Lucas podría trabajar en eso?

Alice reflexionó.

Era un asunto más delicado que el módulo de vacaciones. No bastaba con diseñar unas pantallas. Había que entender qué pasaba realmente en las fábricas, cómo se utilizaban las materias primas y cómo identificaban los clientes sus productos.

Pero precisamente eso le interesaba.

—Creo que podríamos confiarle el problema, sí.

—¿Y crees que conoce suficientemente el sector?

—Probablemente no.

Hubert arqueó las cejas.

—Qué tranquilizador.

—Yo tampoco conozco todos los detalles. Pero sé con quién hablar. Tenemos clientes que conocen sus restricciones al dedillo. Y personas de soporte y de implantación que ven los problemas todos los días. Lucas puede ir a hablar con ellas.

—¿Quieres que hable directamente con los clientes?

—¿Por qué no?

—Normalmente ese es tu trabajo.

—Precisamente.

Hubert no parecía del todo convencido.

—¿Y si construye algo que no responde a la necesidad?

—Mi trabajo es ayudarle a evitarlo. Pero no redactándole una especificación de cincuenta páginas.

—¿Y si quiere poner su solución en producción?

—Habrá que comprobarla y validarla, como cualquier otra mejora del software. Que trabaje con más libertad no significa que le demos permiso para asumir cualquier riesgo.

Hubert se quedó callado unos instantes.

—¿Sabes lo que me preocupa? —preguntó al fin—. No es solo que pueda equivocarse. Es que dejemos de saber quién hace qué.

—A mí también me preocupa.

—¿Y aun así quieres probar?

—Sí.

Se tomó un momento antes de añadir:

—Porque la otra opción es seguir exactamente como antes y pedirles simplemente a los desarrolladores que

programen más deprisa. Y no estoy segura de que eso sea lo que Lucas nos enseñó el viernes.

Siguieron hablando unos veinte minutos.

Hubert quería evitar que el experimento alterara los proyectos en curso. Alice no quería convertirlo en una gran iniciativa anunciada a toda la empresa antes siquiera de empezar.

Así que acordaron mantenerlo sencillo.

Lucas trabajaría en el problema de trazabilidad de los lotes. Tendría libertad para explorar soluciones y reunirse directamente con las personas que pudieran ayudarle a comprender la necesidad.

Alice sería su referente de Producto. No le prepararía un backlog detallado, pero seguiría participando en las decisiones importantes.

Comenzarían con un alcance limitado, sin prometer a ningún cliente una fecha de entrega.

Y, sobre todo, se tomarían el tiempo necesario para observar qué ocurría.

—Deberíamos decidir cómo sabremos si el experimento ha salido bien —señaló Hubert.

—Sí. Y no quiero que midamos solo cuántas funcionalidades ha desarrollado.

—Entonces, ¿qué quieres medir?

—Para empezar, si consigue entender el problema y proponer algo útil. Después, hasta dónde puede avanzar sin que tengamos que organizarle todo el trabajo.

—Y cuánto tiempo nos lleva a nosotros.

—También.

Hubert miró la hora.

—Bueno. Lo intentamos.

Alice sonrió, aunque no se sentía del todo tranquila.

—Lo intentamos.

—¿Hablas con él hoy?

—Sí.

Terminó la videollamada.

Alice permaneció unos instantes frente a la pantalla.

Abrió la mensajería y buscó a Lucas.

Empezó a escribir:

Hola, Lucas. ¿Tienes un momento hoy? Me gustaría proponerte algo un poco distinto de lo que hacemos normalmente.

Releyó el mensaje antes de enviarlo.

Por primera vez en mucho tiempo, iba a confiar un problema a un desarrollador sin haber empezado ella misma a imaginar con precisión la solución.

Todavía no sabía si era una buena idea.

Y ya no estaba tan segura de que su papel consistiera en saberlo antes que él.

4 — Están surgiendo nuevos modelos organizativos

La IA permite ejecutar más trabajo. Pero ¿cómo reorganizan las empresas sus equipos y sus decisiones? Sus respuestas son distintas.

Algunas refuerzan la gestión de producto alrededor de los desarrolladores. Otras amplían la función de estos últimos, dan más medios a los departamentos no técnicos o reorganizan el trabajo en torno a pequeños equipos asistidos por agentes.

El reparto tradicional del trabajo deja de darse por supuesto.

El modelo tradicional tenía su lógica: Product Management concentraba buena parte del contexto del cliente y del negocio, Diseño plasmaba la experiencia e Ingeniería convertía esas intenciones en software. Los tickets, las especificaciones, las entregas entre equipos y las reuniones permitían sincronizar estas funciones en torno a una capacidad de desarrollo limitada.

Esta organización no se convierte de repente en una mala idea.

Pero cuando aumenta la capacidad de ejecución, algunas empresas empiezan a redistribuir de otra manera a las personas, el contexto y el trabajo.

No se trata todavía de buscar «el modelo correcto».

Se trata de observar las principales respuestas que están apareciendo.

Reforzar Producto alrededor de los desarrolladores

La respuesta más intuitiva consiste en mantener una separación relativamente clásica entre Producto e Ingeniería, pero aumentando la capacidad de gestión de producto disponible para los desarrolladores.

Gamma proporciona un ejemplo interesante. La empresa habla de aproximadamente un Product Manager por cada cuatro ingenieros, frente a una proporción histórica que suele situarse más cerca de uno por cada diez. [1]

En este modelo, el aumento de la capacidad de Ingeniería viene acompañado de una mayor presencia de Product Management.

La idea es sencilla: si se pueden construir más cosas, se destina también más capacidad a comprender, priorizar y orientar ese trabajo.

Dar al desarrollador una mayor responsabilidad sobre el asunto

Otra estrategia consiste en ampliar el ámbito de actuación del ingeniero.

El desarrollador ya no espera necesariamente a que alguien convierta por completo un problema en una solución y después en tickets. Puede participar más en la comprensión de la necesidad, la creación de prototipos, las decisiones locales y el seguimiento del resultado.

En su oferta de empleo para Product Engineer, Linear describe las responsabilidades esperadas: llevar una funcionalidad desde la idea hasta el producto entregado y participar en la definición del problema sin depender constantemente de Product Management. [2]

Este modelo trata de reducir algunos relevos entre especialidades concentrando más contexto y capacidad de acción en la misma persona.

No significa que todos los desarrolladores se conviertan en PM.

Más bien describe una ampliación de la función de Ingeniería en determinados asuntos.

Permitir que las profesiones crucen sus antiguas fronteras

Este movimiento no afecta solo a los desarrolladores.

En su estudio sobre la evolución de los roles, Figma señala que el 64 % de los participantes se identifican con al menos dos funciones y que más de un tercio declaran responsabilidades que abarcan varias profesiones. Este resultado no puede atribuirse únicamente a la IA, pero sí describe unas fronteras profesionales menos rígidas. [3]

Atlassian también documenta flujos de trabajo en los que los diseñadores modifican directamente el código real del producto y abren ellos mismos una pull request antes de que Ingeniería lo revise. [4]

Por su parte, Cursor describe una organización en la que Producto, Diseño e Ingeniería participan más directamente en la construcción. [5]

La conclusión no es tanto que «todo el mundo hace de todo», sino que **cada vez más personas intervienen fuera de los límites tradicionales de su profesión**.

Esta mayor permeabilidad plantea otra cuestión: si varias especialidades pueden actuar sobre el mismo terreno, ¿cómo se mantienen claras las responsabilidades? Volveremos a ello.

Confiar más trabajo a pequeños equipos responsables

Otra familia de respuestas consiste en poner un ámbito más amplio de trabajo en manos de menos personas.

AMPECO describe unas reuniones breves de alineación en las que un ingeniero y un Product Owner definen la intención antes de que los agentes se hagan cargo de buena parte del análisis, el desarrollo y las pruebas. [6]

Taito explica una forma de trabajar en la que el equipo valida un plan y, después, una o dos personas dirigen la construcción con la ayuda de agentes. [7]

En estos modelos, una pequeña unidad humana puede poner en marcha mucho más trabajo que antes.

El cambio organizativo observable consiste, por tanto, en una posible reducción de los relevos entre profesiones y en la concentración de la responsabilidad sobre un asunto en una persona o un grupo pequeño.

Queda por definir qué pueden decidir esas personas por sí solas y cuándo deben recurrir al responsable de un ámbito.

Automatizar parte de las comprobaciones

Una producción más rápida también ejerce presión sobre los mecanismos de validación.

Algunas empresas responden automatizando más controles o tratando los cambios de manera diferente según su riesgo.

Meta utiliza un Diff Risk Score para estimar el riesgo de que un cambio provoque un incidente. [8]

Cloudflare establece estándares de Ingeniería que pueden comprobarse automáticamente antes de incorporar cambios. [9]

Taito también combina reglas deterministas, agentes de revisión y criterios de riesgo. [7]

Por ahora, lo importante es constatar que **la validación ya no tiene por qué organizarse igual para todos los cambios**.

Más adelante veremos cómo combinar la automatización, la responsabilidad humana y los umbrales a partir de los cuales hay que elevar una decisión.

Proporcionar contexto en lugar de prescribir cada paso

Otras empresas van más allá: cuestionan el ticket detallado como principal herramienta de coordinación.

Linear critica explícitamente el modelo en el que el PM delimita un problema, lo registra en un sistema y espera a que Ingeniería lo ejecute. La empresa insiste en que las personas y los agentes deben poder acceder

directamente a los comentarios de los clientes, las ideas, la dirección estratégica, las decisiones y el código. [10]

Cursor también subraya la necesidad de establecer objetivos suficientemente claros para permitir una mayor iniciativa local. [5]

El cambio observado no es, por tanto, la desaparición de toda coordinación.

Es el traslado de una parte de esa coordinación desde las instrucciones detalladas hacia **un contexto más completo y directamente accesible**.

Dedicaremos un capítulo a explicar qué debe contener ese contexto, cómo mantenerlo y cómo delimitar la autonomía.

Permitir construir a quien conoce la necesidad

Algunas empresas incluso permiten que los equipos que conocen la necesidad creen pequeñas herramientas por su cuenta, sin tener que pasar siempre por la cola de trabajo de los desarrolladores.

Shopify creó Quick, una plataforma interna que permite a más empleados publicar rápidamente sitios web y pequeñas herramientas. [11]

Vercel describe un modelo en el que empleados no técnicos construyen herramientas internas en un entorno

seguro sin desviar a los desarrolladores de las prioridades del producto principal. [12]

Atlassian también muestra a diseñadores que intervienen en el código antes de la revisión técnica. [4]

En estos casos, la IA no se limita a hacer más rápidos a los desarrolladores.

A veces permite trasladar la ejecución a **la persona que ya conoce el contexto de la necesidad**.

Un principio anterior a la IA

Ya en 2018, Marty Cagan defendía lo que llamaba **empowered product teams**: equipos multidisciplinares a los que la dirección confía problemas que resolver en lugar de listas de funcionalidades que entregar. Les pide que busquen la solución combinando las competencias de Producto, Diseño e Ingeniería y que respondan del resultado, no del volumen producido. [13]

Con Chris Jones desarrolló este enfoque en el libro **EMPOWERED** en 2020. Este modelo no anunciaba ni profesiones intercambiables ni el fin de los managers: por el contrario, exigía una visión compartida, competencias sólidas y responsables que acompañaran a los equipos. [14]

Por tanto, esta idea no surgió con los agentes. Lo que la IA puede cambiar es la capacidad de ejecutar de forma más rápida y directa algunas de las soluciones que el equipo autónomo ha decidido explorar. **La autonomía de los equipos de Producto es una aspiración anterior; la IA cambia sus medios y sus limitaciones.**

Opciones que todavía no convergen

Estas respuestas no son etapas de una misma transformación. Reforzar Producto y dar más responsabilidades de producto a los desarrolladores pueden parecer estrategias contradictorias. Sin embargo, ambas pueden funcionar en empresas diferentes o incluso coexistir dentro de la misma organización.

Lo que tienen en común es otra cosa: **cuando ejecutar resulta más accesible, la manera de repartir el contexto, las decisiones y las responsabilidades vuelve a ser una elección.**

Queda por ver cómo repercute ese desplazamiento en el día a día de las personas, más allá de los anuncios y los organigramas.

Pronto volveremos a Malorean Corp. para observar cómo afectan estos modelos a la vida de una pequeña empresa

francesa. Después examinaremos las responsabilidades y las reglas necesarias para esa autonomía.

Bibliografía

[1] The bottleneck keeps shifting: What AI is changing about how we build — Atlassian / Gamma

- URL: **https://www.atlassian.com/blog/how-we-build/the-bottleneck-keeps-shifting-what-ai-is-changing--about-how-we-build**
- Describe el refuerzo de Product Management en Gamma cuando aumenta la capacidad de producción de Ingeniería.

[2] Product Engineer — Linear Careers

- URL: **https://linear.app/careers/0c7c2e26-0a98-42cf-a47c-9a3999fb513b**
- Oferta de empleo que explica las responsabilidades esperadas de un Product Engineer desde la definición del problema hasta la entrega.

[3] Are Roles and Responsibilities a Thing of the Past? — Figma

- URL: **https://www.figma.com/blog/2025-shifting--roles-report/**

- Documenta la creciente permeabilidad de los roles y la ampliación de responsabilidades más allá de los límites profesionales tradicionales.

[4] Designers' workflow for shipping code — Atlassian

- URL: **https://www.atlassian.com/blog/ai-at-work/designers-workflow-for-shipping-code**
- Explica cómo los diseñadores intervienen directamente en el código del producto antes de la validación de Ingeniería.

[5] How Cursor Is Navigating the New Shape of Product Craft — Atlassian / Cursor

- URL: **https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of--product-craft**
- Muestra una organización en la que Producto, Diseño e Ingeniería participan más directamente en la construcción.

[6] How we built an AI-native engineering system — AMPECO

- URL: **https://www.ampeco.com/blog/how-we-built--an-ai-native-engineering-system/**
- Describe cómo unas pocas personas definen la intención antes de delegar gran parte de la ejecución a agentes.

[7] Inside an AI-native engineering organization — Swarmia / **Taito.ai**

- URL: **https://www.swarmia.com/blog/inside-an-ai--native-engineering-organization/**
- Describe a un equipo pequeño dirigiendo agentes y concentrando buena parte de la ejecución en una o dos personas.

[8] Diff Risk Score: AI-driven risk-aware software development — Meta

- URL: **https://engineering.fb.com/2025/08/06/developer-tools/diff-risk-score-drs-ai-risk-aware--software-development-meta/**
- Presenta una puntuación de riesgo para adaptar los procesos de validación a la criticidad probable de cada cambio.

[9] How Cloudflare enforces engineering standards using AI — Cloudflare

- URL: **https://blog.cloudflare.com/engineering--standards-enforcement/**
- Muestra cómo los estándares de Ingeniería se convierten en controles automáticos dentro del flujo de desarrollo.

[10] Issue tracking is dead — Linear

- URL: **https://linear.app/next**
- Defiende un modelo en el que comentarios, estrategia, decisiones y código comparten más contexto que un ticket aislado.

[11] Quick: An internal hosting platform for the AI era — Shopify

- URL: **https://shopify.engineering/quick**
- Presenta una plataforma interna que permite a más empleados construir y publicar directamente.

[12] Stopping the slow death of internal tools — Vercel

- URL: **https://vercel.com/blog/stopping-the-slow--death-of-internal-tools**
- Muestra cómo los usuarios de negocio pueden crear herramientas internas sin desviar Ingeniería de las prioridades del producto principal.

[13] Empowered Product Teams — Marty Cagan, Silicon Valley Product Group (2018)

- URL: **https://www.svpg.com/empowered-product--teams/**
- Texto de referencia que diferencia los equipos responsables de resolver problemas de los que se limitan a entregar funcionalidades.

[14] EMPOWERED: Ordinary People, Extraordinary Products — Marty Cagan y Chris Jones (2020)

- URL: **https://www.svpg.com/books/empowered--ordinary-people-extraordinary-products/**
- Presenta el libro sobre equipos de Producto autónomos, sus objetivos, el contexto estratégico y el papel de los responsables en su desarrollo.

5 — Malorean Corp. — Lo que podemos hacer y lo que debemos hacer

---

El martes por la mañana, Lucas empezó con una pregunta que sorprendió a Alice.

—¿Puedo hablar con el cliente?

El día anterior había recibido los pocos informes que soporte conservaba sobre el problema de trazabilidad de los lotes. Los mensajes hablaban de números de fabricación, materias primas, archivos Excel y alertas de proveedores. Había pasado la tarde organizándolos con ayuda de la IA.

—Por supuesto —respondió Alice—. ¿Por qué?

—Porque no entiendo exactamente qué es lo que les falta.

Alice sonrió. Una semana antes, probablemente habría organizado ella misma la reunión y después habría transmitido sus conclusiones al equipo de desarrollo. Pero ahora había prometido dejar que Lucas llevara el asunto.

—Te presentaré a la persona de soporte que se ocupa de esa cuenta. Podrá organizar la conversación.

El miércoles, Lucas descubrió que la expresión «gestionar los lotes» ocultaba una situación mucho menos sencilla.

El responsable de calidad del cliente les enseñó una carta recibida de un proveedor. Había que retirar una materia prima. El proveedor conocía el número de lote. El fabricante sabía qué pedidos había recibido. Pero, para localizar con precisión los productos terminados afectados, sus equipos tenían que cruzar varios documentos.

—La última vez —explicó el responsable de calidad— estuvimos cuatro personas revisando archivos durante buena parte del día. Y aun así tuvimos que volver a la fábrica para estar seguros.

Lucas pidió que le mostraran cómo se registraban los datos durante la fabricación. El cliente le enseñó una pantalla del ERP y después una hoja de cálculo.

—Pero ¿por qué apuntáis también ese número en Excel?

—Porque vuestro programa no nos permite encontrar lo que necesitamos.

La respuesta no fue agresiva. Precisamente por eso resultó aún más difícil de escuchar.

Al terminar la llamada, Lucas tenía tres páginas de notas y menos certezas que al principio.

—Bueno, ¿qué tal? —preguntó Alice cuando Lucas volvió a llamarla.

—Había empezado a imaginar una pantalla para registrar los números de lote.

—¿Y ahora?

—Ahora creo que lo primero es entender cómo registran lo que realmente han consumido. Si no, tendremos una pantalla preciosa, pero seguiremos sin poder responder a su pregunta.

Alice asintió. Podría habérselo explicado el lunes, pero no le disgustaba que lo hubiera descubierto por sí mismo.

El viernes siguiente, Lucas le enseñó una primera versión en una base de datos de demostración.

Se podía asociar un lote de materia prima a una operación de fabricación y localizar después los productos terminados resultantes. Había añadido un pequeño ejemplo para seguir un lote hasta los productos afectados por una alerta.

Alice hizo varios clics.

—¿Has hecho esto esta semana?

—Sí. Todavía hay muchas cosas incorrectas o que faltan, pero ya tenemos algo que enseñar al cliente.

—¿Y si añadimos también la posibilidad de seguir el recorrido en sentido contrario?

—Eso ya está.

Abrió otra vista.

Alice soltó una carcajada.

—Claro, cómo no.

Desde la demostración del módulo de vacaciones, había cambiado su manera de imaginar la dimensión de los proyectos. Lo que antes habría pospuesto hasta el trimestre siguiente ahora le parecía estar al alcance de la mano.

—También podríamos buscar por proveedor. Y mostrar una alerta directamente en los pedidos afectados. ¿Y por qué no preparar los documentos para retirar los productos?

Lucas no contestó de inmediato.

—Podemos apuntar esas ideas —dijo al fin—. Pero antes me gustaría volver a ver al cliente con lo que tenemos.

—¿Vamos a esperar quince días para enseñarle una versión más completa?

—No es la demostración lo que me preocupa.

—Entonces, ¿qué te preocupa?

Esta vez Lucas compartió una pantalla bastante menos agradable de mirar. Tablas antiguas, campos con nombres incomprensibles y varios vínculos entre fabricación y existencias.

—Podemos crear todas las pantallas que queramos. El problema es que nuestro ERP nunca se diseñó para conservar toda esa historia de los lotes. Hay movimientos de existencias de hace años, algunos corregidos a mano. Y cierta información simplemente no existe.

—La IA puede ayudarnos a ponerlo todo en orden, ¿no?

—Puede ayudarme a escribir una migración. No puede inventar lo que nunca registramos.

Alice guardó silencio.

—Si queremos hacerlo bien —continuó Lucas—, tendremos que modificar parte del modelo de datos y comprobar qué efectos tiene en los demás módulos. No podemos permitirnos desajustar las existencias de un cliente para ganar dos semanas.

—No te estoy pidiendo que rompas el programa.

—Ya lo sé. Pero desde el viernes, cada vez que te enseño algo, me pides otras tres cosas.

No había agresividad en el comentario. Aun así, a Alice le dolió un poco.

—Es verdad —reconoció.

—A mí también me entran ganas de ir deprisa. Ya me conoces. Pero que pueda generar código en una hora no significa que quince años de funcionamiento del ERP se vuelvan sencillos.

Después de dudar un instante, añadió:

—Thomas ha revisado parte del código del módulo de vacaciones. Ha encontrado dos casos que no habíamos tratado bien. Tenía razón. Ya los hemos corregido, pero me ha recordado algo: podemos construir mucho más deprisa y seguir necesitando a alguien que conozca las trampas.

Alice recordó a Thomas el viernes de la demostración, cuando se limitó a pedir que le enseñaran el código.

—Entonces, ¿qué propones?

—Seguimos con la demostración utilizando datos ficticios. Comprobamos que hemos entendido bien la necesidad. Pero, antes de tocar las existencias reales, quiero que dediquemos tiempo a comprender el sistema actual y hacer las comprobaciones necesarias.

Alice miró la primera versión, que seguía abierta en su otra pantalla.

La semana anterior había defendido la autonomía de Lucas ante Hubert. De repente, le parecía mucho menos agradable descubrir que esa autonomía incluía el derecho a decirle que no.

—De acuerdo —dijo finalmente—. Por ahora no tocamos los datos reales.

El lunes siguiente, Hubert abrió su reunión semanal con una pregunta que empezaba a resultar familiar.

—¿Cómo va nuestro experimento?

—Avanza —respondió Alice—. Más deprisa de lo previsto en algunas cosas. Más despacio en otras.

—¿Por ejemplo?

Alice le mostró la demostración de Lucas y resumió la conversación del viernes.

Hubert se detuvo en las reservas técnicas.

—Hace bien en tener cuidado —dijo—. No quiero que resolvamos un problema de trazabilidad provocando problemas de existencias a todos nuestros clientes.

—Sí. Y creo que le he presionado un poco.

—¿Tú?

—Desde hace dos semanas, tengo la sensación de que todo es posible. En cuanto aparece una idea, pienso que podemos probarla enseguida.

Hubert sonrió.

—Así me sentí yo después de su primera demostración.

—El problema es que resulta muy fácil confundir la velocidad con la que conseguimos ver algo en una pantalla con la velocidad a la que obtenemos algo fiable.

—Al menos —dijo Hubert— Lucas sabe cuándo hay que frenar.

—Sí. Y tengo que confiar en su criterio para eso.

8

Dos días después, Julie Morel preguntó a Alice si tenía cinco minutos.

Alice conocía bien aquella expresión. Con Julie, cinco minutos podían convertirse fácilmente en media hora.

—La semana pasada me reuní con un posible cliente bastante importante —empezó Julie—. Tienen varias fábricas y buscan un ERP que se adapte a sus formas de trabajar. Mira lo que he preparado.

Compartió una serie de maquetas muy cuidadas. Mostraban distintas etapas de aprobación según la categoría de pedido, reglas propias de cada

establecimiento y pantallas que cambiaban en función del usuario.

—¿Has hecho tú todo esto? —preguntó Alice.

—Con ChatGPT. Recuperé mis notas de la reunión, le pedí que propusiera un recorrido y luego lo fuimos corrigiendo juntas. Me llevó una tarde.

Alice comprendió de inmediato el valor de aquellas imágenes. En unas horas, Julie había dado forma concreta a una necesidad que hasta entonces solo existía en un informe comercial.

—¿Y el cliente ha visto las maquetas?

—Todavía no. Quería enseñártelas primero. Pero me dijo claramente que, si podíamos hacer ese tipo de cosas, la conversación cambiaría mucho.

Alice pasó de una pantalla a otra.

—La idea es interesante. Deberías comentársela directamente a Lucas.

Julie la miró sorprendida.

—¿Directamente? Pensaba que tú prepararías el asunto.

—¿Por qué? Tú conoces al cliente, has hablado con él e incluso has preparado las maquetas. Puedes explicarle lo que busca mejor que yo.

—¿Y Lucas puede ver si es viable?

—Sí. Pero, Julie, cuidado: estudiar si es viable no significa comprometernos a desarrollarlo.

—Por supuesto.

Alice conocía aquel «por supuesto». No era mentira. Era el entusiasmo de una comercial ante una posible venta.

Al día siguiente se reunieron con Lucas. Julie repasó sus maquetas, explicó las peticiones del posible cliente y respondió a las preguntas.

Lucas se interesó enseguida por cómo cambiaban las reglas de un establecimiento a otro.

—Ya tenemos parte de ese mecanismo en el ERP —dijo—. No para hacer exactamente esto, pero existe. Seguramente podríamos preparar algo para probarlo.

—¿En cuánto tiempo? —preguntó Julie.

—¿Para una demostración? No necesariamente mucho.

Julie miró a Alice. Su sonrisa era casi la de alguien que ya hubiera cerrado el contrato.

—Si se lo mostramos la semana que viene, puede marcar una gran diferencia.

Alice sintió que sus propias ganas de avanzar estaban a punto de responder antes de que hubiera pensado. Recordó el proyecto de trazabilidad, las peticiones

pendientes, las reglas particulares de aquel cliente y todos los demás que quizá no las necesitarían nunca.

—No —dijo.

Los otros dos se volvieron hacia ella.

—¿No a qué? —preguntó Julie.

—No a empezar a desarrollar esa funcionalidad dentro de nuestro producto para conseguir este contrato.

—Pero acabas de decir que es interesante.

—Lo es. Pero no sabemos si ese cliente necesita una función que debería estar disponible para todos o una forma de trabajar muy particular de sus fábricas.

—Podemos perder la venta.

—Es posible.

Julie claramente no esperaba aquella respuesta.

—Por fin tenemos medios para responder deprisa a estas peticiones —insistió—. Sería una pena no aprovecharlos.

—No quiero renunciar a esa posibilidad. Quiero que entendamos qué estamos ofreciendo. Si añadimos todas las peticiones particulares al núcleo del ERP, acabaremos con un programa imposible de mantener y mejorar.

—Pero Lucas dice que podemos hacerlo.

—Lucas dice que puede preparar una demostración. No es lo mismo.

Hubo un silencio. Julie miró sus maquetas una última vez.

—Entonces, ¿qué le digo?

—Que hemos entendido su problema y queremos enseñarle una posibilidad. No que hayamos decidido incorporar esa funcionalidad al producto.

Julie asintió. Parecía decepcionada, pero no desanimada.

—Vale. Puedo organizar una reunión para que nos explique mejor las diferencias entre sus fábricas.

—Perfecto. Y conserva las maquetas. Nos ayudarán a hacer las preguntas adecuadas.

Cuando Julie salió de la llamada, Alice se quedó unos segundos con Lucas.

—¿De verdad podrías haber construido eso deprisa?

—Una primera versión, sí.

—¿Y te habría apetecido hacerlo?

—¿Sinceramente? Sí. Parecía divertido.

Alice se echó a reír.

—Yo también tenía ganas de decir que sí.

—¿Qué te ha hecho cambiar de idea?

—Que los dos estábamos hablando de lo que podíamos hacer y casi habíamos dejado de hablar de lo que debíamos hacer.

La semana siguiente, Lucas envió a Alice un enlace acompañado de un mensaje.

Estuve haciendo unas pruebas ayer. ¿Puedes echarles un vistazo?

Alice abrió la página. Reconoció la interfaz de su ERP, pero había un pequeño panel lateral que permitía modificar ciertas reglas. Podía elegir un establecimiento, cambiar un umbral de aprobación, mostrar u ocultar una etapa y volver a lanzar la demostración.

Llamó a Lucas.

—¿Qué es esto?

—Una pequeña herramienta para preparar demostraciones con reglas distintas. He reutilizado nuestro antiguo mecanismo de configuración. Solo funciona en una instancia de pruebas.

—¿Cuánto tiempo te ha llevado?

—Dos horas, quizá. Se me ocurrió al recordar las maquetas de Julie.

—¡Pero si ni siquiera es el asunto en el que estás trabajando!

—No. Quería comprobar si podíamos evitar modificar el código cada vez que quisiéramos probar una variante.

Alice empezó a cambiar parámetros. La pantalla reaccionaba. Probó otra configuración y después una tercera.

—Espera. ¿Quieres decir que puedo preparar yo misma varias versiones de una demostración?

—Con las reglas que hemos puesto a disposición de la herramienta, sí. No cualquier cosa. Y, desde luego, nunca en los entornos de los clientes.

—Pero puedo enseñar dos recorridos distintos a Julie y preguntarle cuál se corresponde realmente con lo que busca su posible cliente.

—Exactamente.

Alice se quedó un momento mirando la página. El primer módulo de vacaciones había permitido a Lucas trabajar más deprisa. Esta herramienta le daba a ella una posibilidad que no tenía el día anterior.

Podía probar una idea sin pedirle a un desarrollador que la implementara específicamente para ella.

—¿Te imaginas lo que podríamos hacer con esto?

Lucas sonrió.

—Creo que sí. Pero antes de que me pidas convertirlo en un producto completo, te recuerdo que es algo que he improvisado en dos horas.

—Prometido.

Colgó y volvió a jugar con los parámetros.

El lunes siguiente, Hubert observó que Alice no había abierto su documento habitual de seguimiento.

—¿No has preparado nada para hoy?

—Sí. Pero quiero enseñarte una cosa.

Le mostró la herramienta de Lucas y creó ante sus ojos dos variantes de la misma demostración.

Hubert lo entendió bastante rápido.

—¿No has pedido a los desarrolladores que hagan estos cambios?

—No. Lucas ha creado una pequeña herramienta. Puedo probar ciertas reglas por mi cuenta.

—¿Podríamos dejar que la utilizara el equipo comercial?

—Quizá más adelante. De momento no es lo bastante sólida. Pero Julie ya puede trabajar conmigo en los casos de su posible cliente.

Hubert contempló la pantalla con interés.

—¿Y el proyecto de los lotes?

—Lucas sigue avanzando. Ha vuelto a visitar al cliente con su primera versión. Entendemos mejor qué hay que

hacer. Pero se niega a tocar los datos históricos antes de asegurar cómo vamos a tratarlos.

—Me parece razonable.

—Sí. El viernes me molestó. Hoy me alegro de que insistiera.

Alice hizo girar el bolígrafo entre los dedos.

—Está ocurriendo algo extraño, Hubert. Lucas habla con clientes y toma decisiones sobre las soluciones. Julie llega con maquetas casi utilizables. Y yo ahora puedo probar algunas variantes de nuestro ERP sin pasar por desarrollo.

—¿Te parece preocupante?

—A ratos. Me pregunto qué será de nuestras profesiones si todo el mundo empieza a hacer parte del trabajo de los demás.

—¿Y has encontrado una respuesta?

—No del todo.

Se quedó pensando.

—Solo sé que no quiero ser la persona por la que deba pasar cada idea. Julie no tenía ninguna razón para esperar a que yo convirtiera sus maquetas en tickets. Lucas no necesita que le explique cómo construir cada pantalla.

—Pero te has negado a que desarrollemos la petición del posible cliente.

—Sí. Porque era una decisión sobre nuestro producto. Podía equivocarme y todavía puedo cambiar de opinión, pero era responsabilidad mía asumirla.

—Y Lucas se negó a tomar un atajo con las existencias.

—Porque sabe lo que nos arriesgamos a romper. En eso tiene un criterio que debo escuchar. Aunque yo tenga prisa.

Hubert observó un instante la pequeña ventana en la que aparecía Alice.

—Así que podemos dejar que las personas hagan más cosas, pero no podemos permitir que las decisiones queden flotando entre ellas sin responsable.

—Eso es lo que estoy empezando a pensar.

Dudó y añadió:

—Antes, a veces tenía la sensación de que mi trabajo era importante porque muchos asuntos pasaban por mí. Ahora me pregunto si lo más importante no es lo que conseguimos al final. ¿Resolvemos el problema adecuado? ¿Construimos algo fiable? ¿Merece la pena hacerlo?

Hubert asintió.

—Y si nadie sabe quién debe responder a esas preguntas, tendremos un problema.

—Sí.

Permanecieron unos segundos en silencio. Ninguno de los dos quería convertir aquella conversación en el anuncio de una nueva organización.

El experimento apenas acababa de empezar. Les había abierto nuevas posibilidades, pero también les había dado más decisiones que tomar.

Alice recibió una notificación. Era un mensaje de Julie: el posible cliente había aceptado ver las variantes de demostración. También quería saber cuándo podrían estar disponibles para sus equipos.

Alice no respondió inmediatamente.

El software permitía ahora fabricar respuestas mucho más deprisa.

Pero alguien todavía tenía que decidir cuáles estaba Malorean Corp. dispuesta a prometer y a asumir.

6 — Los roles se difuminan, las responsabilidades permanecen

Durante mucho tiempo, las fronteras entre Producto, Diseño e Ingeniería sirvieron como puntos de referencia sencillos.

Producto comprendía el problema y decidía qué construir. Diseño concebía la experiencia. Ingeniería convertía esa intención en software. En realidad, esta división nunca fue tan nítida, pero aun así estructuraba los equipos, las contrataciones, los procesos y las responsabilidades.

Este reparto ya se cuestionaba antes de la IA: los equipos de Producto autónomos descritos en el capítulo 1.4 hacían colaborar a Producto, Diseño e Ingeniería en torno a un problema compartido. Ahora la IA acelera el cruce de otra frontera: la de la ejecución, cuando cada persona puede realizar más tareas que antes estaban reservadas a una profesión.

El panorama organizativo anterior mostró esta evolución, en particular a través del estudio de Figma. [1] En Malorean también describí el movimiento en ambas

direcciones: los desarrolladores se acercan a la comprensión del problema y los PM, a su ejecución. [2]

Sería tentador deducir que los roles van a desaparecer. Creo que este cambio nos invita más bien a reformular la responsabilidad de cada uno: **cuanto más permeables son los roles, más explícitas deben ser las responsabilidades.**

La capacidad de actuar ya no pertenece a una sola profesión

Lo primero que redistribuye la IA no es necesariamente la autoridad para decidir. Es la capacidad de actuar.

Pensemos en los diseñadores que modifican directamente el código, una práctica documentada por Atlassian: un equipo técnico sigue revisando la calidad y las condiciones para llevar los cambios a producción. [3] Este paso no supone volver a la antigua división del trabajo; demuestra que poder intervenir en el código no es lo mismo que responder de su fiabilidad.

La oferta de empleo de Linear para Product Engineer ilustra la misma distinción: la empresa espera que esta persona pueda llevar un asunto desde el problema inicial hasta el resultado entregado. [4] Eso no le concede

automáticamente autoridad sobre todas las decisiones de estrategia de producto.

Por tanto, se pueden reducir los relevos entre profesiones sin eliminar los conocimientos especializados que justificaban algunos de ellos. La pregunta ya no es cuántas especialidades intervienen, sino **quién responde de qué cuando varias trabajan sobre el mismo terreno**.

Poder contribuir no significa ser responsable del ámbito

Hay que distinguir dos cosas que las organizaciones suelen confundir: el permiso para actuar y la responsabilidad.

Un comercial puede construir una herramienta interna. Eso no lo convierte en el responsable de la seguridad de la plataforma.

Un PM puede modificar una interfaz. Eso no significa que pase a ser responsable de la coherencia del sistema de diseño.

Un desarrollador puede tomar una decisión local sobre el comportamiento del producto. Eso no le atribuye automáticamente la responsabilidad sobre la estrategia, el posicionamiento o la hoja de ruta global.

Esta distinción ya existe en muchos equipos, pero la IA obliga a hacerla explícita.

En un modelo muy especializado, la profesión y la responsabilidad solían coincidir. Quien podía actuar también pertenecía al ámbito del que respondía. En cuanto la capacidad de actuar se extiende, esa coincidencia desaparece.

El riesgo consiste entonces en confundir autonomía con responsabilidad diluida.

Ya he defendido un principio parecido al hablar de calidad: los desarrolladores no pueden entregar un cambio y considerar después que sus defectos son cosa de un equipo de QA, de un turno para corregir errores o del siguiente desarrollador que esté libre. La calidad técnica de lo que se entrega sigue siendo una responsabilidad identificable. [5]

La mayor permeabilidad de los roles no cambia este principio. Lo hace más importante.

El concepto de Garante

Para responder a este problema utilizo el concepto de **Garante**.

En cada ámbito importante, una persona o un equipo sigue siendo garante de su coherencia y calidad.

El Garante no tiene por qué realizar todo el trabajo de ese ámbito.

Tampoco es quien debe aprobar cada pequeña decisión.

Es la persona o el equipo al que la organización acude cuando una decisión importante afecta a ese ámbito, cuando se supera un límite o cuando algo sale mal.

En una organización sencilla, podemos imaginar al menos tres responsabilidades de garantía explícitas:

- **Garante de Producto**: coherencia del producto, decisiones estructurales, cliente objetivo, propuesta de valor, decisiones importantes y relación con la estrategia.
- **Garante de Diseño**: sistema de diseño, coherencia de la experiencia, accesibilidad, nuevos patrones y nivel de exigencia de la experiencia de usuario.
- **Garante de Desarrollo**: arquitectura, seguridad, rendimiento, facilidad de mantenimiento, calidad técnica y condiciones de puesta en producción.

El principio es sencillo:

todos pueden contribuir a varios ámbitos, pero cada uno conserva a alguien que responde por él.

Este modelo permite cruzar fronteras profesionales sin que las responsabilidades desaparezcan.

Un comercial puede construir sin convertirse en garante técnico

Pongamos un ejemplo deliberadamente sencillo.

Un comercial detecta una necesidad, utiliza una herramienta asistida por IA y crea una pequeña aplicación o una modificación útil. Siempre que la empresa disponga de un entorno suficientemente seguro, no hay razón para considerar que solo Ingeniería tiene derecho a construir.

Pero esa libertad no significa que la calidad técnica pase a ser responsabilidad del comercial.

Si el cambio afecta a la seguridad, los permisos, los datos sensibles o la arquitectura, el Garante de Desarrollo debe poder exigir una revisión, imponer una corrección o bloquear el cambio.

El comercial sigue siendo responsable de la necesidad que quiere resolver.

Ingeniería sigue siendo garante de lo que la empresa acepta mantener desde el punto de vista técnico.

Lo mismo se aplica al Diseño.

Un PM puede montar una pantalla con componentes conocidos. Si se mantiene dentro del sistema de diseño y utiliza patrones dominados, exigir sistemáticamente una

aprobación de Diseño reduciría innecesariamente su autonomía.

En cambio, si ese PM se aparta del sistema de diseño, inventa un patrón nuevo o concibe una interacción importante y compleja, el Garante de Diseño debe poder intervenir.

La frontera ya no es «¿quién tiene derecho a tocar qué?».

Pasa a ser: **«¿a partir de qué nivel de riesgo o novedad debe intervenir el garante de un ámbito?»**

El responsable de un asunto y el Garante de un ámbito no son la misma persona

Esta es probablemente una de las distinciones más importantes en las nuevas organizaciones.

El **owner** es quien lleva un asunto.

Debe comprender el problema, impulsar el trabajo, tomar las decisiones locales, coordinar lo necesario, comprobar el resultado y responder de que el asunto llegue a buen puerto.

El **Garante** asume una responsabilidad más transversal.

No tiene por qué ser dueño del asunto. Responde de un determinado nivel de coherencia, calidad o riesgo dentro de un ámbito.

Un Product Engineer puede ser owner de una funcionalidad sin convertirse en garante de toda la arquitectura de la empresa.

Un PM puede ser owner de un lanzamiento sin convertirse en garante de todo el Diseño.

Un comercial puede ser owner de una herramienta de negocio sin convertirse en garante de la seguridad de las aplicaciones internas.

Esta distinción aclara quién responde del resultado sin exigir que una sola persona sea experta en todos los ámbitos.

AMPECO ilustra una variante de este movimiento: un Product Owner y un ingeniero alinean el «qué» y el «por qué», mientras los agentes se encargan de buena parte del análisis, el desarrollo y las pruebas. La ejecución se distribuye, pero una persona sigue siendo responsable de la validación final. [6]

Por tanto, la IA puede reducir el número de personas necesarias para ejecutar un asunto y, al mismo tiempo, aumentar la necesidad de saber con toda claridad **quién responde de él**.

Puede que el puesto ya no sea la mejor forma de entender la organización

Esta evolución plantea una pregunta más amplia: ¿tiene sentido seguir organizando la empresa principalmente a partir de los títulos de los puestos?

El debate sobre el Product Owner es revelador. En Malorean defendí la idea de que el PO es, ante todo, una función —un mandato de decisión y responsabilidad— más que una profesión que encuentre su razón de ser en mantener un backlog. [7]

Esta distinción resulta especialmente útil con la IA.

Si definimos un puesto por la lista de tareas que realiza, la IA vuelve inestable esa definición. Muchas tareas pasan a estar al alcance de perfiles más diversos.

Si definimos una responsabilidad por aquello de lo que una persona o un equipo deben responder, resiste mucho mejor.

Un PM puede escribir menos especificaciones sin dejar de responder por la calidad de las decisiones de producto.

Un desarrollador puede escribir menos código manualmente sin dejar de responder por la calidad técnica.

Un diseñador puede dejar de ser la única persona que mueve elementos en una pantalla sin dejar de ser garante de la coherencia de la experiencia.

En Cursor, donde las fronteras entre Producto, Diseño y Desarrollo también son menos rígidas, su responsable de Producto recuerda lo esencial: elegir qué merece ser construido y ofrecer una orientación suficiente para evitar desviaciones. [8]

Así, las profesiones cambian no tanto porque pierdan todas sus responsabilidades como porque dejan de tener el monopolio sobre algunas formas de ejecución.

El Garante no debe convertirse en otra burocracia

Hay, sin embargo, una trampa evidente.

Si cada intervención de un PM en el código tiene que recibir la aprobación de Ingeniería, cada decisión local de un desarrollador debe pasar por el PM y cada pantalla creada por alguien que no sea diseñador espera a un comité de Diseño, entonces no habremos resuelto nada.

Simplemente habremos recreado los antiguos pasos obligatorios con otro nombre.

El Garante no debe convertirse en una aprobación sistemática.

8

Su existencia sirve precisamente para permitir más autonomía: cada persona puede actuar dentro de unas reglas conocidas, sin pedir permiso a cada momento, porque la organización sabe dónde residen las responsabilidades y en qué casos deben volver a intervenir sus garantes.

Por otra parte, en la vida real este modelo nunca funciona a la perfección.

Llegan a producción cosas que el Garante no habría aprobado exactamente de esa manera. Algunas decisiones locales pasan inadvertidas. Los equipos a veces se toman más libertades de las previstas. Otras personas, en cambio, elevan una cuestión demasiado pronto por prudencia.

Eso no significa necesariamente que el modelo haya fracasado.

Una organización verdaderamente autónoma acepta que parte de las decisiones se tome sin inspección previa. De lo contrario, no hay autonomía.

El problema no consiste, por tanto, en garantizar que nunca haya desviaciones.

Consiste en definir **qué desviaciones son aceptables, cuáles debe detectar el sistema y cuáles requieren la**

intervención de un Garante antes de que resulten costosas.

Aquí cambia la pregunta.

Hemos distinguido dos responsabilidades: la de quien lleva un asunto y la del Garante de un ámbito.

Queda por resolver otra cuestión:

¿cuándo debemos dejar que el owner decida por su cuenta y cuándo tiene que intervenir el Garante?

En Malorean Corp., esta frontera entre iniciativa y responsabilidad empieza a generar fricciones.

Volveremos a ellas en el siguiente episodio, antes de examinar los mecanismos de control y las reglas para elevar decisiones.

Bibliografía

[1] Are Roles and Responsibilities a Thing of the Past? — Figma

- URL: **https://www.figma.com/blog/2025-shifting--roles-report/**
- Estudia la ampliación de las responsabilidades y la creciente permeabilidad de las profesiones en los equipos actuales de desarrollo de producto.

[2] PM, développeurs et IA : les rôles se brouillent, les responsabilités restent — Malorean

- URL: **https://malorean.net/articles/2026-08-01-pm--developpeurs-et-ia-les-roles-se-brouillent-les-responsabilites-restent.es.html**
- Presenta la interpretación del autor: la IA amplía quién puede actuar sin eliminar las responsabilidades de Producto, Diseño e Ingeniería.

[3] Designers' workflow for shipping code — Atlassian

- URL: **https://www.atlassian.com/blog/ai-at-work/designers-workflow-for-shipping-code**
- Explica cómo los diseñadores modifican directamente el código del producto y se mantiene la revisión de Ingeniería para asegurar la calidad y el despliegue.

[4] Product Engineer — Linear Careers

- URL: **https://linear.app/careers/0c7c2e26-0a98-42cf-a47c-9a3999fb513b**
- Oferta que describe lo que Linear espera de un Product Engineer, desde la definición del problema hasta la entrega de funcionalidades.

[5] La qualité appartient à ceux qui livrent — Malorean

- URL: **https://malorean.net/articles/2026-06-03-la--qualite-appartient-a-ceux-qui-livrent.es.html**
- Defiende una responsabilidad técnica explícita: la calidad no puede delegarse después de entregar un cambio a quienes reparan los errores.

[6] How we built an AI-native engineering system — AMPECO

- URL: **https://www.ampeco.com/blog/how-we-built--an-ai-native-engineering-system/**
- Muestra una ejecución ampliamente delegada a agentes sin retirar a las personas la responsabilidad sobre la intención, la arquitectura y la validación.

[7] Le PO n’est pas un métier, c’est une fonction — et l’IA n’y change rien, elle accélère — Malorean

- URL: **https://malorean.net/articles/2026-08-03-le--po-nest-pas-un-metier-cest-une-fonction.es.html**
- Distingue el mandato de decisión y ownership de una profesión definida principalmente por gestionar artefactos o un backlog.

[8] How Cursor Is Navigating the New Shape of Product Craft — Atlassian / Cursor

- URL: **https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of--product-craft**
- Muestra fronteras profesionales más permeables, manteniendo el criterio de producto, los objetivos y los mecanismos de control como responsabilidades esenciales.

7 — Malorean Corp. — Lo que Alice sabía sin haberlo escrito nunca

---

El jueves por la mañana, Lucas llamó a Alice por tercera vez aquella semana.

—Solo serán dos minutos —prometió.

Alice había aprendido a desconfiar de aquella frase. Lucas nunca le pedía reuniones largas. Llegaba con una pregunta concreta, la comentaban y él volvía a trabajar. Precisamente por eso resultaba difícil negarse a aquellas interrupciones.

—Dime.

—Sobre la trazabilidad de los lotes, el cliente quiere poder buscar por proveedor. Eso puedo añadirlo a la demostración. Pero si el proveedor ha entregado varios lotes con el mismo número interno, ¿los mostramos juntos o le pedimos que elija?

Alice se lo pensó unos segundos.

—No debemos dar a entender que hemos identificado un lote si hay varios que coinciden. Muestra los resultados por separado. Podemos comprobar con el cliente cómo los distingue.

—Eso pensaba, pero no estaba seguro.

—Entonces, ¿por qué me llamas?

Lucas soltó una risa incómoda.

—Porque después de lo de las maquetas de Julie, prefiero consultarte antes de decidir algo que parezca una regla de producto.

La semana anterior, Alice le había pedido que no convirtiera la petición de un posible cliente en una funcionalidad estándar. Ahora le daba pena que Lucas se hubiera quedado con la idea de que era mejor consultarla ante la menor duda.

—De acuerdo. Pero para esta no me necesitabas.

—Me tranquiliza oírtelo decir.

Colgaron. La llamada había durado siete minutos. Alice volvió al documento que estaba escribiendo, releyó la última frase y ya no recordó qué quería añadir.

Diez minutos después, Lucas le envió un mensaje.

Otra pregunta: en las fabricaciones antiguas para las que no tenemos información sobre los lotes, ¿ponemos «desconocido» u ocultamos la columna?

Empezó a escribir una respuesta. Después se detuvo.

La pregunta era distinta, pero el razonamiento era el mismo: nunca presentar como cierta una información que

el ERP no poseía. Podría haberle dado ese principio en lugar de resolver los dos casos por separado.

Aun así, contestó.

Mostramos claramente que falta la información. No ocultemos el problema.

Lucas respondió con un pulgar hacia arriba.

Alice miró la agenda. Una reunión con soporte se había aplazado. Por la tarde tenía una cita con un cliente. Entre ambas debía revisar las propuestas de Julie para su posible cliente con varias sedes.

Le gustaba que Lucas hablara directamente con ella. Era justamente lo que había querido cuando le confió el asunto. Pero él avanzaba tan deprisa que con cada nueva posibilidad aparecía una nueva pregunta. Y cuanto más la consultaba, menos tiempo tenía ella para hacer aquello para lo que estaba mejor preparada: entender a los clientes y pensar en la dirección del producto.

Lucas también perdía tiempo. A veces esperaba su respuesta para continuar trabajando.

Se le pasó por la cabeza una idea desagradable: si aquello seguía así, quizá acabarían metiendo de nuevo todas las decisiones en tickets. Sería más fácil organizarse. Ella escribiría las especificaciones, Lucas

ejecutaría y nadie tendría que preguntarse hasta dónde llegaba su autonomía.
Justo lo que habían intentado superar.

Al mediodía, Alice se conectó con Bob por videollamada. Habían adquirido la costumbre de hablar sin orden del día, simplemente para contrastar lo que les iba ocurriendo.
—¿Qué tal con Lucas? —preguntó Bob.
—Demasiado bien, en cierto modo.
—No parece un cumplido.
—Avanza muy deprisa, se da cuenta de muchas cosas y habla con los clientes. Pero me llama continuamente para consultarme decisiones que podría tomar él solo.
—¿Y tú le contestas?
—Por supuesto.
—Entonces hace bien en llamarte.
Alice puso los ojos en blanco.
—Gracias, Bob.
—Solo lo digo medio en broma. Si puedo conseguir tu respuesta en cinco minutos en lugar de arriesgarme a equivocarme, ¿por qué no iba a hacerlo?
—Porque nunca conseguiremos trabajar así.

Bob le pidió un ejemplo. Alice le contó las dos preguntas de aquella mañana, después la conversación con Julie y, finalmente, una duda reciente de Lucas: ¿debían añadir una opción solicitada por un cliente si probablemente solo le serviría a él?

—¿Te das cuenta? —dijo Alice—. Siempre son variantes de la misma pregunta.

—¿Y cuál es tu respuesta?

—Depende.

Bob sonrió.

—Ahí lo tienes. Esa es precisamente la respuesta que obliga a la gente a volver a llamarte.

Alice estaba a punto de protestar, pero tenía razón.

—No es un «depende» arbitrario —continuó—. Miro si respondemos a un problema recurrente, si es útil para nuestros clientes prioritarios, si vamos a complicar el producto para todo el mundo, si podremos mantener la solución…

Se interrumpió.

—¿Lo ves? —dijo Bob—. Acabas de darme parte de la regla.

—No es una regla. Es más bien una manera de pensar.

—Mejor todavía. Quizá sea eso lo que deberías transmitirle a Lucas.

Alice guardó silencio.

Llevaba años escribiendo tickets. En ellos describía los comportamientos esperados, los casos particulares y los criterios de aceptación. Pero casi nunca escribía aquello que le permitía escoger entre varios comportamientos posibles. Ese razonamiento se quedaba en su cabeza o se perdía en alguna reunión cuando se entregaba el trabajo.

—Probablemente tengo cientos de cosas así —dijo.

—Seguramente. Yo también.

—Y nunca hemos tenido motivos para ponerlas por escrito.

—Motivos sí teníamos. Lo que no teníamos era tiempo. Y nadie abría un ticket llamado «explicar por qué a veces rechazamos una buena idea».

Ambos se rieron.

Después de la llamada, Alice abrió un documento nuevo. Escribió: **Principios para tomar decisiones de producto**.

Añadió una primera frase, la borró y probó con otra. Al cabo de quince minutos había escrito un párrafo que

parecía una política interna. Era correcto. Pero no se parecía en absoluto a lo que le había explicado a Bob.

Cerró el documento.

Después abrió ChatGPT.

—Ayúdame a sacar de mi cabeza todo lo que sé de nuestro producto —dictó—. No quiero un procedimiento teórico. Hazme preguntas, ayúdame a recuperar ejemplos y a distinguir qué es una regla de verdad y qué es simplemente una costumbre mía.

La primera respuesta le propuso empezar por las decisiones recientes.

Alice empezó a hablar.

Habló del posible cliente de Julie, de las atractivas maquetas y del riesgo de convertir una necesidad particular en una promesa para todos los clientes. Explicó que una cuenta importante podía representar mucho dinero sin convertirse automáticamente en la dirección adecuada para el producto.

—Espera, tengo que explicarte por qué —añadió en voz alta—. Nuestro ERP sirve a empresas muy distintas. Si añadimos todas las reglas particulares al núcleo del programa, acabamos complicándole la vida a todo el

mundo. Pero a veces una petición particular revela un problema mucho más general. Tampoco podemos rechazarlo todo.

ChatGPT le preguntó cómo establecía la diferencia.

Alice describió las conversaciones con soporte, los casos observados en varios clientes, las peticiones de los comerciales y las limitaciones de las implantaciones. Recordó debates antiguos de los que no había conservado ninguna acta. Explicó por qué habían descartado algunas opciones, por qué habían preferido una configuración a una nueva funcionalidad y por qué una excepción relacionada con nóminas no se trataba igual que una preferencia de interfaz.

A veces se sorprendía respondiendo a preguntas que nunca se había planteado formular explícitamente.

La IA reformulaba, agrupaba ideas y proponía un principio. Alice lo corregía.

—No, no es que «nunca desarrollemos nada específico». Podemos hacerlo en determinadas condiciones, pero tenemos que decidir conscientemente qué nos cuesta. Y, sobre todo, no debemos presentar una hipótesis comercial como si fuera una decisión de producto.

Un poco después, ChatGPT escribió que una funcionalidad de trazabilidad podía garantizar la reconstrucción de todos los lotes históricos.

—¡De eso nada! —dijo Alice ante la pantalla—. Si los datos nunca se registraron, no podemos reconstruirlos por arte de magia. Eso es una limitación que hay que mostrar, no una capacidad que se pueda prometer.

Hizo corregir el pasaje. Pidió que el documento distinguiera lo que estaba decidido, lo que ella consideraba probable y lo que quedaba por comprobar con soporte, Lucas o los clientes. Evitó introducir información confidencial de las cuentas reales; reformuló los ejemplos.

Pasó una hora. Después, otra.

El documento se llenaba de cosas que Alice sabía desde hacía mucho: los clientes que Malorean quería atender prioritariamente, los problemas recurrentes, los compromisos aceptados, las promesas que no debían hacerse, los límites de los datos antiguos y los motivos por los que un prototipo no era una funcionalidad lista para vender.

Añadió enlaces a la información que ya tenían: comentarios del equipo de soporte, decisiones que aún

podían recuperarse, notas de reuniones y documentación del producto. Cuando faltaban pruebas, el texto tenía que indicarlo.

Hacia el final de la tercera hora, ChatGPT le propuso una síntesis.

Alice la releyó. Algunas páginas eran demasiado largas. Otras mezclaban un principio general y un ejemplo. Corrigió por última vez las formulaciones demasiado categóricas y añadió una fecha y el alcance del documento.

Después lo exportó como PDF.

Fue pasando las páginas, sorprendida por la cantidad que había.

Tres horas antes, solo pretendía explicarle a Lucas cuándo podía tomar decisiones sin consultarla. Ahora tenía en las manos un retrato imperfecto, pero sorprendentemente rico, de lo que el equipo sabía sobre el producto sin haberlo compartido nunca por escrito.

No era una verdad definitiva. Algunos pasajes requerían confirmación o actualización. El PDF no sustituiría a los clientes, los datos ni el criterio de los responsables.

Pero al menos la siguiente persona no tendría que empezar desde cero.

Lucas abrió el archivo a la mañana siguiente.

—¿Has escrito todo esto para mí?

—No yo sola —respondió Alice—. He pasado tres horas hablando con ChatGPT. Me ha ayudado a plantear las preguntas, ordenar mis respuestas y detectar lo que faltaba. Después lo he releído y corregido. Pero la mayoría de estas cosas ya las sabía.

—Entonces, ¿por qué no las había visto nunca?

Alice sonrió, algo avergonzada.

—Porque no cabían en ningún ticket.

Lucas recorrió el índice. Encontró principios, ejemplos de decisiones, limitaciones del negocio, información sobre los clientes, enlaces a las fuentes y preguntas aún abiertas.

—¿Puedo darle esto a mi IA cuando trabaje en la trazabilidad?

—Sí, dentro de nuestro entorno autorizado. Pero dile que lo utilice como contexto, no como prueba de que todo es cierto. Y si una regla es importante, volvemos a la fuente o a la persona que responde de ella.

—Me ahorraría muchísimo tiempo. A veces le pido a la IA que me ayude a escoger entre dos soluciones, pero solo sabe lo que le he contado. Puede proponerme una

respuesta muy convincente que no tenga en cuenta una decisión de hace tres años.

Alice conocía esa sensación. Una respuesta razonable por sí sola podía ser absurda para su ERP.

Lucas se detuvo en un pasaje.

—Aquí dices que nunca debemos ocultar información de negocio que falta si eso puede hacer pensar que el resultado está completo.

—Es lo que te dije ayer.

—Sí. Y si hubiera tenido esa frase ayer, no te habría llamado.

Siguió leyendo. Después levantó la vista.

—Vale, pero aún tengo un problema.

Alice sintió un momento de desaliento.

—¿Cuál?

—Ahora entiendo mejor cómo razonas. Pero sigo sin saber a partir de qué momento tengo derecho a decidir por mi cuenta.

Alice dejó la taza.

Acababa de encontrar el agujero más importante del documento.

Retomaron juntos el asunto. Alice compartió una página en blanco.

—Vamos a probar algo sencillo. Antes de llamarme, te haces tres preguntas: ¿entiendo el resultado que buscamos? ¿Mi decisión se mantiene dentro de las reglas conocidas? ¿Qué podría salir mal si me equivoco?

—Habrá que definir qué significa «dentro de las reglas».

—Precisamente.

Escribieron tres niveles.

Verde: Lucas decidía por su cuenta cuando la decisión era local, reversible, conforme a las reglas conocidas y no suponía una promesa nueva al cliente. Tenía que poder explicar su razonamiento y comprobar el resultado.

Naranja: consultaba a Alice cuando encontraba una situación nueva, una contradicción entre objetivos o una duda importante sobre el comportamiento del producto. Si había un riesgo técnico, también debía recurrir a las personas competentes, como Thomas cuando se trataba del historial del ERP.

Rojo: nadie decidía por su cuenta cuando se trataba de modificar una regla estructural, comprometer comercialmente a Malorean, tocar datos sensibles o asumir un riesgo importante y difícilmente reversible. Era

necesario tomar una decisión explícita con los responsables correspondientes y, a veces, con Hubert.

Lucas repasó la tabla.

—Entonces, si cambio cómo se presentan los resultados en la demostración para evitar confusiones, eso es verde.

—Sí.

—Si quiero añadir un comportamiento que podría afectar a todos nuestros clientes, vengo a verte.

—Lo hablamos, sí. Sobre todo si no hemos entendido la necesidad.

—¿Y si quiero modificar los datos históricos para registrar los lotes?

—No puedo darte yo sola el visto bueno. Hace falta un análisis técnico y los controles adecuados. Me lo has recordado unas cuantas veces.

Lucas sonrió.

—Quería asegurarme de que te acordabas.

Añadieron una regla: que Alice no respondiera no convertiría nunca una decisión naranja o roja en verde. Pero tampoco había que pedirle permiso para cada decisión verde.

—¿Y si me equivoco de color? —preguntó Lucas.

—Me lo dices. Ajustamos las reglas. Prefiero que mejoremos el marco a pasarnos la vida reprochándote que no adivinaste lo que estaba pensando.

Lucas asintió.

—Incluso podría pedirle a mi IA que contraste mis decisiones con esta tabla, explicando las razones y la información que falta.

—Sí. Siempre que no decida por nosotros si un riesgo es aceptable.

Alice incorporó la tabla al documento de referencia. Empezaba a entender la diferencia entre transmitir información y dar realmente los medios para decidir.

8

Podrían haberse quedado ahí. Pero Lucas volvió a la demostración de trazabilidad.

—Llevamos una hora hablando de mi autonomía. Voy a hacerte una pregunta sobre el asunto en sí: ¿cómo sabremos si lo hemos resuelto bien?

Alice respondió casi automáticamente:

—Si el cliente puede localizar los productos fabricados con un lote determinado.

—Sí, pero ¿cómo lo comprobamos? ¿Y qué nivel de fiabilidad necesitamos?

Esta vez no buscó una respuesta inmediata.

Habían pasado días explorando pantallas, reglas y estructuras de datos. Tenían una demostración prometedora. Sin embargo, nadie había escrito todavía con precisión cómo evaluar el resultado.

Alice recordó al responsable de calidad que había dedicado buena parte de un día a cruzar archivos tras una alerta de un proveedor.

—Deberíamos partir de ahí —dijo—. Si recibe la alerta de un número de lote, ¿puede encontrar las fabricaciones afectadas sin volver a hacer todas las búsquedas a mano? ¿Cuánto tarda? ¿Está completo el resultado? Y, cuando no disponemos de la información, ¿lo indicamos claramente?

—Podemos probarlo con él. No limitarnos a preguntarle si le gusta la demostración.

—Exactamente.

Alice abrió una página nueva.

—La verdad es que deberíamos haber empezado por aquí antes de lanzarnos.

Lucas se echó a reír.

—Si me hubieras pedido esto hace dos semanas, te habría dicho que estabas intentando volver a convertírmelo en un ticket.

—Puede ser. Pero mira lo que estamos escribiendo. Anotaron el problema observado, los usuarios afectados, el resultado esperado, cómo medirlo, las limitaciones de los datos, lo que quedaba fuera del alcance y las decisiones que necesitaban aprobación. Una página, nada más.

Lucas propuso llamarlo **Mini PRD**.

—No me estás diciendo qué pantalla tengo que construir —observó—. Nos estamos poniendo de acuerdo sobre lo que hará útil el trabajo.

—Y sobre los límites que no debemos cruzar.

Alice pensó en voz alta:

—Para los próximos asuntos, esto debería ser nuestro punto de partida. Antes de preguntarnos qué vamos a desarrollar, preguntarnos qué resultado queremos conseguir y cómo lo comprobaremos. Solo después definimos el alcance y decidimos qué merece la pena construir.

—Y si sé completar este documento con la información adecuada, puedo empezar sin esperar a que me prepares un backlog.

—Puedes proponer el planteamiento. Eso no significa que validemos cualquier compromiso sin revisarlo. Pero sí, ese es el objetivo.

Completaron el Mini PRD de la trazabilidad dejando abiertas las preguntas para las que todavía no tenían respuesta. Aún había que comprobar algunos casos con el cliente y confirmar ciertas limitaciones técnicas.

El documento no creaba una falsa sensación de certeza. Decía qué querían aprender.

Alice lo consideró más útil que una lista de funcionalidades por marcar.

El lunes siguiente, la reunión semanal de Alice con Hubert fue diferente.

Alice no compartió ni el cuadro de seguimiento habitual ni la lista de funcionalidades desarrolladas por Lucas.

Le mostró los niveles de decisión y después el Mini PRD.

Hubert leyó las primeras líneas.

—¿Habéis dedicado tanto tiempo a hacer documentos ahora que Lucas va mucho más deprisa?

Alice conocía la objeción. Ella misma la había planteado unos días antes.

—Precisamente creo que nos faltaban documentos de este tipo. Escribíamos qué había que construir, pero casi nunca la información necesaria para elegir. Como resultado, Lucas me pedía continuamente que decidiera. Con esto puede tomar parte de las decisiones sin mí.

—¿Y cómo sabes que se mantiene dentro de los límites?

—Hemos definido cuándo puede decidir por su cuenta y cuándo tiene que consultarnos. Por lo demás, no vamos a inspeccionarlo todo. Comprobaremos los resultados y corregiremos las reglas si es necesario.

Hubert se detuvo en la línea sobre los datos antiguos.

—No podréis garantizar la trazabilidad de lo que nunca se registró.

—No —respondió Alice—. Y el documento lo deja claro. Tenemos que mostrárselo al cliente.

Hubert asintió.

—Eso prefiero saberlo antes de que un comercial prometa lo contrario.

—Justo —dijo Alice—. El contexto no será útil solo para Lucas. Julie también podría consultar lo que sabemos y lo que todavía no hemos decidido. Pero habrá que

mantener la información actualizada y dejar claro qué partes tienen autoridad.

Acordaron conservar una versión de trabajo editable, de la que el PDF sería una fotografía con fecha. Las decisiones importantes se registrarían junto con sus motivos; los principios recurrentes podrían convertirse, cuando fuera pertinente, en instrucciones para los agentes o en controles de las herramientas. Nada de eso sustituiría las pruebas técnicas ni las validaciones necesarias antes de poner algo en producción.

—No quiero descubrir dentro de seis meses que la IA está aplicando una vieja regla que ya nadie defiende — dijo Hubert.

—Yo tampoco —respondió Alice.

Por primera vez, Hubert pareció especialmente interesado en el Mini PRD.

—¿Y vais a comprobar si el cliente puede localizar realmente sus lotes?

—Sí —respondió Alice—. Vamos a pedirle que pruebe casos con nosotros. Si el resultado no es fiable, dará igual cuántas pantallas hayamos terminado.

Hubert cerró el documento.

—Muy bien. Quiero saber qué aprendéis, no solo qué entregáis.

El miércoles, Alice observó algo extraño en su mensajería.

No tenía ningún mensaje de Lucas desde por la mañana.

Revisó el canal del proyecto. Sin embargo, había actividad: Lucas había probado varias formas de presentar los resultados en la instancia de demostración. En una de ellas había explicado que había optado por separar los lotes con referencias idénticas para no sugerir una relación inexistente. Citaba el principio incluido en el contexto de Producto, indicaba qué datos había utilizado y proponía comprobar con el cliente si la pantalla se entendía bien.

No había pedido permiso.

Alice leyó las explicaciones y volvió al trabajo. No tenía nada que añadir.

Al final de la tarde, Lucas la llamó.

—Tengo un naranja que quizá se esté volviendo rojo —dijo.

—Te escucho.

—El cliente también quiere buscar en sus fabricaciones antiguas. Para parte del historial no tenemos una relación fiable entre las materias primas y los productos terminados. Puedo mostrar lo que sabemos, pero si presentamos una búsqueda completa, podrían creer que no obtener resultados significa que ningún producto está afectado.

Alice se incorporó.

—Tienes razón. No podemos permitir esa ambigüedad.

—Podría añadir una advertencia y diferenciar los periodos cuyos datos son utilizables. Pero me gustaría que decidiéramos juntos qué podemos prometerle al cliente.

—En este caso, sí. Y también tendremos que consultar con soporte cómo entendería el cliente esa limitación.

Lucas compartió una pantalla. Empezaron a trabajar sobre el problema. No era la aprobación de un botón ni una respuesta que Alice ya conociera. Tenían delante una decisión que podía cambiar la confianza que un cliente depositaría en el software.

Durante la conversación, Alice se dio cuenta de que ni una sola vez había pensado en cuánto tiempo le estaba quitando.

Antes de colgar, Lucas vaciló.

—Sabes, tu PDF no ha respondido a todas mis preguntas.

—Menos mal —dijo Alice.

—¿Menos mal?

—Si hubiera logrado meter en él todas las respuestas, querría decir que ya no te quedaría nada por descubrir.

Lucas sonrió.

—De acuerdo. Pero creo que ahora entiendo mejor cuándo venir a verte.

Después de la llamada, Alice abrió de nuevo su agenda. Tenía una hora libre antes de la siguiente reunión con un cliente.

Una hora entera.

Todavía no sabía si su experimento funcionaría a mayor escala ni cuántos errores cometerían por el camino. Solo sabía que Lucas había avanzado sin ella en varias decisiones y había vuelto justo cuando su criterio realmente importaba.

Añadió al documento compartido una nueva pregunta pendiente sobre cómo presentar los datos históricos. No era un ticket que le dijera a Lucas qué tenía que programar. Era una decisión que debían estudiar con las personas implicadas.

Su forma de trabajar no se había vuelto menos exigente. Simplemente empezaba a situar la exigencia donde hacía falta.

8 — Más autonomía requiere reglas más claras, no menos

Una organización puede decidir que más personas deben poder actuar sin tener que esperar a un especialista.

La permeabilidad de los roles y las responsabilidades que hemos analizado crean un nuevo problema de coordinación: varias personas pueden intervenir sobre el mismo asunto sin disponer necesariamente de los mismos criterios para decidir.

Pero si cada nueva capacidad de actuar exige una aprobación humana sistemática, la autonomía sigue siendo teórica.

Simplemente habremos trasladado los pasos obligatorios de una profesión a otra.

El verdadero reto consiste en crear un sistema en el que se puedan tomar más decisiones de manera local sin perder coherencia, seguridad, calidad ni orientación de producto.

Aquí aparece una paradoja importante:

cuanto más queremos distribuir la autonomía, más explícitas deben ser las reglas.

La autonomía no nace de la ausencia de normas.

Nace de poder actuar sin pedir permiso continuamente porque los objetivos, los límites, los riesgos y las responsabilidades están suficientemente claros.

Si todo necesita aprobación, no hay autonomía

Imaginemos un Product Engineer al que confiamos la responsabilidad de un asunto.

Habla con los usuarios, explora una solución, crea prototipos, desarrolla y prepara el lanzamiento.

Sobre el papel, es autónomo.

Pero si debe preguntar al PM por cada pequeña decisión de producto, al diseñador por cada detalle de la interfaz y al Tech Lead por cada decisión técnica, la organización no habrá redistribuido realmente la responsabilidad. Solo habrá obligado a una persona a pasar por más aprobaciones.

El mismo obstáculo afecta a las demás profesiones: permitir que alguien construya no sirve de mucho si cada paso exige los procesos y las validaciones de siempre.

Por tanto, la cuestión no es únicamente **quién puede actuar**.

También hay que saber **dentro de qué límites puede esa persona decidir por sí misma**.

En Cursor, esta lógica se expresa mediante objetivos y mecanismos de control suficientemente claros para que quienes construyen puedan actuar sin desviarse de la visión ni de la hoja de ruta. [1]

Así, la organización sustituye parte de las instrucciones detalladas por un marco para tomar decisiones.

Sustituir parte de los tickets por criterios de decisión

Ya vimos, con Linear, por qué un ticket que lo prescribe todo puede limitar la iniciativa. [2] Ahora la pregunta es práctica: ¿de qué información necesita disponer un equipo para decidir por sí mismo sin esperar otra especificación?

No se trata de eliminar los tickets. En Malorean he defendido el backlog como herramienta para actuar y coordinarse, no como memoria completa del producto. [3]

Un marco de autonomía podría, por tanto, hacer explícita la información que orienta las decisiones locales sin fijar de antemano cada solución.

Podría incluir:

- el cliente o perfil de cliente ideal (ICP) prioritario;
- el problema que hay que resolver;
- el resultado esperado;

- los límites del alcance;
- las restricciones importantes;
- el nivel de calidad exigido;
- aquello que la organización acepta expresamente no hacer;
- los umbrales a partir de los cuales debe elevarse una decisión.

Ramp ofrece un ejemplo concreto de esta lógica: aclarar los ICP fundamentales, los problemas prioritarios y aquello que la empresa quiere resolver especialmente bien también permite aceptar que se rechacen oportunidades lucrativas cuando desvían el producto hacia otra dirección. [4]

Figma recuerda, por su parte, que pasar rápidamente de cero a uno no basta. Los casos reales, las excepciones y el nivel de calidad siguen siendo determinantes para transformar un prototipo rápido en un buen producto. [5]

Esta lista ilustra una elección organizativa: no limitarse a precisar **lo que hay que hacer**, sino hacer visibles los límites que deben respetarse. No la presento como un método universal.

El contexto se convierte en infraestructura

Para tomar buenas decisiones de forma local, necesitamos acceder a la información que permite comprender el problema.

Probablemente sea uno de los cambios organizativos más importantes de la era de los agentes.

Una organización no puede pedir que más personas tomen decisiones y, al mismo tiempo, mantener casi todo el contexto en la cabeza de unos pocos especialistas.

Linear destaca que los comentarios de clientes, la orientación estratégica, las decisiones y el código deben estar al alcance tanto de las personas como de los agentes. [2]

En Malorean he descrito la función del PM como la de un **arquitecto del Contexto**: transformar conocimiento disperso —entrevistas, objeciones comerciales, soporte, decisiones, datos, fuentes y señales débiles— en un recurso organizativo reutilizable. [6]

Esta idea cambia profundamente la función de la documentación.

El objetivo deja de ser producir más páginas.

Consiste en hacer accesibles los elementos que alguien necesita para decidir sin tener que reconstruir todo el razonamiento desde cero.

Un buen contexto permite comprender:

- qué han dicho realmente los clientes;
- qué hipótesis se han probado ya;
- qué restricciones importan;
- qué decisiones estructurales se han tomado;
- qué compromisos se han aceptado;
- qué métricas permiten evaluar el resultado.

Cuanto mayor sea la autonomía, más fiable y compartido deberá ser ese contexto.

De lo contrario, la empresa distribuye capacidad para actuar sin distribuir capacidad para comprender.

Conservar la memoria del porqué

Esta redistribución crea otro problema.

Se toman más decisiones de manera local, a veces por personas distintas y a veces con agentes que no estuvieron presentes cuando se resolvieron cuestiones anteriores.

Pero una decisión sin memoria pronto vuelve a ser objeto de debate.

Se recuerda la conclusión, pero no el contexto.

Se sabe que existe una regla, pero ya no se recuerda por qué se eligió.

Llega un nuevo cliente, entra un nuevo responsable, cambia un equipo y una discusión que ya se había resuelto vuelve a empezar con menos información que la primera vez.

Esa es precisamente la función que atribuyo a los **Product Decision Records**. [7]

Un PDR no documenta cada pequeña decisión.

Registra aquellas decisiones de producto suficientemente importantes como para orientar a varios equipos, varios productos o distintas situaciones futuras.

Conserva, entre otras cosas:

- el contexto;
- las alternativas consideradas;
- la decisión adoptada;
- sus motivos;
- las consecuencias positivas y negativas;
- los riesgos aceptados;
- las condiciones que justificarían reabrir la decisión;
- las personas que decidieron o fueron consultadas.

El objetivo no es añadir burocracia.

Al contrario, un buen PDR debe evitar que se repita cincuenta veces la misma discusión.

En este punto resulta útil distinguir:

el código conserva el registro de lo que se ha construido; la memoria de las decisiones debe conservar por qué se eligieron determinadas reglas estructurales.

Para mantener la autonomía en el tiempo, necesitamos ambas cosas.

Incorporar los controles al propio sistema

Incluso con un buen contexto, sería ineficaz exigir a todos los empleados que memorizaran todas las reglas de la organización.

Por eso algunas empresas trasladan las reglas repetitivas desde la aprobación humana hacia las herramientas. Es un mecanismo observable, no una garantía de que la automatización baste para asegurar la calidad.

Meta utiliza un Diff Risk Score para estimar la probabilidad de que un cambio provoque un incidente y alimentar procesos adaptados al riesgo. [8]

Cloudflare transforma estándares explícitos de Ingeniería en comprobaciones automáticas. Algunas reglas obligatorias pueden impedir que se apruebe un cambio hasta que se haya corregido o autorizado expresamente una excepción. [9]

Este movimiento es importante porque cambia el papel de los expertos.

Un Garante técnico no necesita recordar manualmente la misma regla en cincuenta pull requests si esa regla puede comprobarse automáticamente.

Un Garante de Diseño no necesita aprobar cada botón si el sistema de diseño hace fácil la elección correcta y visible la incorrecta.

Un Garante de Producto no necesita volver a explicar cada semana las mismas restricciones si las decisiones estructurales, los clientes objetivo y los objetivos están accesibles.

Las reglas se convierten en infraestructura.

Pruebas, CI, sistemas de diseño, estándares, reglas de seguridad, plantillas, instrucciones para agentes, métricas y contexto compartido asumen parte del control que antes realizaban las personas.

Así, los expertos pueden concentrarse en las excepciones.

No todas las decisiones necesitan el mismo control

Aquí es donde el riesgo se vuelve fundamental.

Una traducción de texto y un cambio en el sistema de permisos no deberían seguir el mismo proceso de validación.

Un cambio local y reversible no exige el mismo nivel de revisión que una migración de datos o una decisión que modifica el posicionamiento de la empresa.

Una manera de comprender estas organizaciones es sustituir la pregunta binaria —«¿hace falta aprobarlo?»— por una pregunta gradual: **¿qué nivel de control justifica este riesgo?**

Los tres niveles siguientes son **una propuesta de análisis de este libro**, no una práctica documentada en todas las empresas citadas.

Autonomía local

La decisión se mantiene dentro de los objetivos y estándares conocidos.

Es reversible, sus posibles consecuencias son limitadas y no altera una regla estructural.

El owner decide.

Validación específica

La decisión introduce una novedad, una excepción o un riesgo importante dentro de un ámbito.

Interviene el Garante correspondiente: Desarrollo, Diseño, Producto u otro.

No tiene por qué hacerse cargo del asunto. Aporta el criterio especializado que falta.

Escalado
La decisión modifica los propios límites del sistema.

En el ámbito técnico: arquitectura estructural, seguridad, datos sensibles, migraciones críticas o riesgo normativo.

En Diseño: una interacción nueva e importante, una ruptura con el sistema de diseño o consecuencias relevantes para otros recorridos.

En Producto: cambios del cliente objetivo, la propuesta de valor, los precios, inversiones importantes, la hoja de ruta o decisiones estratégicas.

En ese caso, el owner ya no está decidiendo simplemente **dentro** de las reglas.

Está proponiendo cambiarlas.

En el modelo que propongo, esta es la clase de situación que requiere la intervención de un Garante o un nivel superior de decisión.

Sin límites explícitos aparecen dos riesgos opuestos
La ausencia de límites claros al decidir expone a la organización a dos riesgos opuestos.

El primero es la burocracia.

Todo se eleva. Los expertos se convierten en cuellos de botella. Los responsables tienen que aprobar decisiones triviales y la autonomía desaparece.

El segundo es la dilución de las responsabilidades. Cada persona decide localmente sin saber cuándo una decisión deja de ser local. Se acumulan decisiones contradictorias hasta provocar un problema visible para el cliente o la empresa.

En Malorean he descrito varias herramientas de coherencia organizativa que sirven precisamente para hacer más claras estas fronteras: lenguaje compartido, documentos de una página, jerarquía de métricas, registro de decisiones y reglas de escalado. [10]

Su valor no consiste en generar más documentación. Consiste en obligar a responder explícitamente ciertas preguntas:

- ¿quién decide?
- ¿sobre qué asuntos?
- ¿en qué momento?
- ¿qué puede mantenerse en el ámbito local?
- ¿qué debe elevarse?
- ¿en qué condiciones puede reabrirse una decisión?

Estos dos riesgos llevan a una hipótesis organizativa: **situar mejor las decisiones** puede importar más que multiplicar o eliminar aprobaciones.

Decidir también significa aceptar una renuncia

Los mecanismos de control no pueden ser exclusivamente técnicos.

Una parte esencial de las reglas es la capacidad de tomar decisiones que impliquen renuncias reales.

Las organizaciones suelen adoptar decisiones tan vagas que mantienen una apariencia de consenso. Cada persona puede interpretarlas según sus prioridades y, más adelante, la ejecución revela que las restricciones eran incompatibles.

Llamo a esto **decisiones ambiguas**. [11]

Son especialmente peligrosas en un sistema muy autónomo.

Si el objetivo es ambiguo, cada equipo puede tomar una decisión razonable localmente, pero incoherente en conjunto.

Por tanto, conceder más autonomía también exige hacer visibles las renuncias aceptadas.

¿Qué cliente tiene prioridad?

¿A qué segmento no vamos a atender por ahora?

¿Qué nivel de calidad es innegociable?

¿Qué restricción podemos sacrificar temporalmente?

¿Qué objetivo prevalece cuando entra en conflicto con otro?

La autonomía no necesita unas reglas que lo digan absolutamente todo.

Necesita un marco que deje los verdaderos compromisos suficientemente claros para que las decisiones locales converjan.

Medir el resultado se convierte en el último mecanismo de control

La autonomía no termina cuando se entrega algo.

Si alguien es realmente responsable de un asunto, también tiene que observar qué ocurre después.

El primer capítulo ya distinguía la cantidad producida del valor creado. Ahora la pregunta es distinta: **¿cómo sabe después quien ha tomado una decisión si era acertada?**

Canva ofrece un mecanismo concreto: sus equipos pueden relacionar sus experimentos con una estimación de su impacto comercial. [12]

Para cada asunto, esto nos lleva a precisar:

- qué transformación se esperaba;

- qué medida permite observarla;
- qué nivel de calidad debía mantenerse;
- qué hicieron realmente los usuarios;
- si conviene continuar, corregir, abandonar o investigar más.

La medición cierra el ciclo de responsabilidad.
Sin ella, la autonomía puede convertirse en un simple permiso para producir.
Con ella, se transforma en responsabilidad sobre el resultado.

La autonomía no supone que el management desaparezca

El modelo de equipos autónomos de Marty Cagan, presentado en el capítulo 1.4, ayuda a aclarar aquí la función del management. Confiar un problema a un equipo sin entregarle una solución predeterminada exige explicar los objetivos y el contexto estratégico. [13]
Cagan insiste además en el desarrollo de competencias, el acompañamiento y el liderazgo: dejar que el equipo decida no significa abandonarlo. [14]
La comparación tiene, sin embargo, un límite importante. Cagan describe la responsabilidad de equipos de Producto con competencias complementarias. El

concepto de Garante, que propongo en este libro, responde a otro problema: ¿cómo conservar la responsabilidad de un ámbito cuando la ejecución pasa de una profesión a otra?

El Garante cuida las reglas, no aprueba cada gesto

El capítulo 1.6 presentaba al Garante como responsable de la coherencia de un ámbito incluso cuando no ejecuta personalmente cada tarea. Ahora su función se concreta: establecer reglas comprensibles, automatizar las comprobaciones repetitivas e intervenir cuando el riesgo o la novedad supera lo que un equipo puede decidir por su cuenta.

Si tiene que inspeccionarlo todo, vuelve a convertirse en un cuello de botella. Si nadie responde ya de nada, la responsabilidad se diluye.

Por tanto, la autonomía no consiste en aprobar cada acción ni en eliminar todos los controles: consiste en hacer posibles más decisiones locales manteniendo claras las responsabilidades.

Pero esta transformación plantea una pregunta que va más allá de Producto e Ingeniería.

Si el contexto circula directamente, si las decisiones pueden tomarse en ámbitos más locales y si los agentes

asumen parte de la coordinación, **¿qué sucede con los niveles de management que antes desempeñaban esas funciones?**

Malorean Corp. nos mostrará primero cómo afecta esta cuestión a personas concretas. Después examinaremos sus implicaciones para toda la empresa.

Bibliografía

[1] How Cursor Is Navigating the New Shape of Product Craft — Atlassian / Cursor

- URL: **https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of--product-craft**
- Muestra cómo los objetivos, los mecanismos de control y el criterio de producto permiten actuar a roles más permeables sin desviarse de la dirección elegida.

[2] Issue tracking is dead — Linear

- URL: **https://linear.app/next**
- Defiende un sistema en el que los comentarios de clientes, la estrategia, las decisiones y el código comparten más contexto que los procesos centrados en tickets.

[3] Le backlog n’est pas un dépotoir : c’est un outil d’action — Malorean

- URL: **https://malorean.net/articles/2026-06-03-le-backlog-nest-pas-un-depotoir.es.html**
- Distingue el backlog, herramienta temporal de coordinación, de la memoria, el contexto y el razonamiento de producto que deben mantenerse.

[4] 5 insights from Ramp’s CPO on building for hypergrowth — Ramp

- URL: **https://ramp.com/blog/5-insights-from-ramps-cpo-reddit-ama**
- Destaca los ICP y problemas prioritarios y la necesidad de rechazar oportunidades que dispersan los esfuerzos.

[5] What It Takes to Build Great Products Now — Figma

- URL: **https://www.figma.com/blog/what-it-takes-to-build-great-products-now/**
- Recuerda que crear prototipos con rapidez no basta: los casos reales, las excepciones y la calidad siguen siendo decisivos para conseguir un producto duradero.

[6] Le PM comme architecte du Contexte — Malorean

- URL: **https://malorean.net/articles/2026-06-08-le--pm-comme-architecte-du-contexte.es.html**
- Propone transformar señales, fuentes, decisiones y criterios dispersos en un contexto reutilizable para mejorar las decisiones futuras.

[7] Product Decision Record : tracer les choix produit qui structurent l'entreprise — Malorean

- URL: **https://malorean.net/articles/2026-06-03--product-decision-record-tracer-les-choix-produit.es.html**
- Formaliza una memoria breve de las decisiones estructurales: contexto, alternativas, decisión, consecuencias y condiciones explícitas para revisarla.

[8] Diff Risk Score: AI-driven risk-aware software development — Meta

- URL: **https://engineering.fb.com/2025/08/06/developer-tools/diff-risk-score-drs-ai-risk-aware--software-development-meta/**
- Presenta una puntuación de riesgo que permite adaptar los procesos de control a la probabilidad de incidente de un cambio.

[9] How Cloudflare enforces engineering standards using AI — Cloudflare

- URL: **https://blog.cloudflare.com/engineering--standards-enforcement/**
- Muestra cómo los estándares explícitos pasan a ser controles automáticos capaces de bloquear ciertas infracciones antes de producción.

[10] Les outils de cohérence organisationnelle — Malorean

- URL: **https://malorean.net/articles/2026-06-02-les--outils-de-coherence-organisationnelle.es.html**
- Relaciona lenguaje común, objetivos, métricas, memoria de decisiones y reglas de escalado para aclarar los criterios organizativos.

[11] Pourquoi les organisations préfèrent les décisions molles — Malorean

- URL: **https://malorean.net/articles/2026-06-02--pourquoi-les-organisations-preferent-les--decisions-molles.es.html**
- Analiza las decisiones aparentemente consensuadas que ocultan renuncias, diluyen responsabilidades y desplazan las contradicciones hacia la ejecución.

[12] Measuring Commercial Impact at Scale — Canva

- URL: **https://www.canva.dev/blog/engineering/measuring-commerical-impact-at-scale/**
- Describe una medición de impacto comercial de autoservicio que conecta experimentos locales, decisiones de producto y creación de valor empresarial.

[13] Team Objectives – Empowerment — Marty Cagan, Silicon Valley Product Group (2020)

- URL: **https://www.svpg.com/team-objectives--empowerment/**
- Defiende que los equipos reciban problemas por resolver acompañados del contexto estratégico necesario para tomar decisiones locales.

[14] Product Leadership Is Hard — Marty Cagan, Silicon Valley Product Group (2020)

- URL: **https://www.svpg.com/product-leadership-is--hard/**
- Explica por qué los equipos de Producto autónomos requieren un liderazgo más exigente y un acompañamiento activo, no la retirada del management.

9 — Malorean Corp. — Dos decisiones razonables

El lunes por la mañana, Vivien, responsable de coordinar los dos equipos de desarrollo, comenzó la reunión con la misma pregunta de siempre.

—¿Quién puede contarme cómo van los asuntos de esta semana?

Ya tenía abierto su cuadro de seguimiento. Los tickets, los responsables, los plazos y los bloqueos estaban ordenados en unas columnas que había ido perfeccionando pacientemente. Conocía a los dos equipos de cinco desarrolladores, sus fortalezas, sus costumbres, los asuntos en los que debía insistir y los que era mejor dejar madurar.

Nadie respondió de inmediato.

Entonces habló Lucas.

—El viernes actualicé el Mini PRD de trazabilidad. Probamos con el cliente la búsqueda por lote. Lo siguiente es validar los casos en los que faltan datos. He anotado las preguntas abiertas en el contexto.

—¿Y la fecha de entrega? —preguntó Vivien.

—Todavía depende de las comprobaciones sobre las existencias históricas. Thomas tiene que revisar un punto conmigo.

Thomas asintió. Él estaba trabajando en una mejora del proceso de preparación de envíos. También había actualizado el avance en la herramienta que utilizaban.

Vivien fue preguntando a los demás desarrolladores. Las respuestas eran parecidas. Cada uno conocía mejor su problema, había tomado algunas decisiones y podía explicar el siguiente paso. Algunos todavía pedían ayuda para dividir el trabajo. Los demás no habían esperado a aquella reunión para avanzar.

Al terminar, Vivien miró su cuadro.

Todo lo que acababa de escuchar ya estaba allí, a veces con más detalles.

—Bueno —dijo—. Ya hemos repasado todo.

La reunión había durado doce minutos.

Antes habría necesitado cuarenta y cinco.

Vivien no sabía muy bien si alegrarse.

Llevaba en Malorean el tiempo suficiente para haber conocido a todos los desarrolladores trabajando juntos en la oficina de Le Havre. Había organizado entregas,

gestionado retrasos, negociado prioridades con Producto y explicado al director por qué una petición prometida para el viernes no estaría disponible hasta el martes siguiente.

Cuando la empresa pasó al teletrabajo después de la pandemia, adaptó su método. Ya no bastaba con hablar por los pasillos. Pidió que las decisiones quedaran por escrito, que los bloqueos fueran visibles y que todos registraran los acuerdos alcanzados con otros departamentos.

Aquello se convirtió en su trabajo cotidiano: hacer circular información que nadie tenía completamente a la vista.

Le gustaba. Disfrutaba especialmente cuando un proyecto que iba mal recuperaba el rumbo porque alguien había conseguido poner en contacto a las personas adecuadas.

Pero desde que los equipos empezaron a utilizar herramientas de IA, algo había cambiado. Los desarrolladores producían más deprisa y, sobre todo, algunos empezaban a encargarse de los asuntos de otra manera. Lucas hablaba directamente con los clientes. Alice compartía un contexto de Producto que explicaba las razones de las decisiones, no solo la lista de

funcionalidades pendientes. Bob empezaba a adoptar la misma práctica en sus propios proyectos.

Y ahora un agente preparaba para Vivien parte de los informes que antes elaboraba a mano.

El trabajo no había desaparecido. Se le escapaba en pequeñas partes o se hacía en otra parte.

Todavía no se lo había contado a nadie.

El miércoles siguiente, Lucas le enseñó a Alice una nueva versión de la demostración de trazabilidad.

—Hemos probado qué ocurre cuando un proveedor señala un lote potencialmente defectuoso. Podemos identificar las fabricaciones de las que tenemos información. Y he añadido un estado «en revisión» para que no parezca que está todo en orden con ese lote.

Alice miró la pantalla.

Un aviso indicaba que había que examinar la situación. Para las fabricaciones antiguas, otro mensaje aclaraba que no se disponía de cierta información.

—Es más honesto que la primera versión —dijo.

—Sí. El responsable de calidad del cliente nos pidió expresamente que no mezcláramos «no se ha

identificado ningún problema» con «no tenemos información suficiente».

Alice asintió. Era una distinción que ya habían recogido en el documento de contexto.

—¿Lo has anotado en el Mini PRD?

—Sí. Y he añadido casos de prueba.

—Entonces, sigue adelante.

No necesitó decidir dónde colocar el aviso ni cuál sería el texto exacto del mensaje. Lucas sabía qué resultado querían conseguir. Había elegido una presentación y pensaba contrastarla con el cliente.

Un poco después, Thomas le mostró a Vivien una mejora del módulo de expediciones.

El problema era distinto. En algunos clientes, un pedido podía quedar detenido debido a estados intermedios o mal registrados, aunque no existiera una prohibición explícita. Los equipos de logística perdían tiempo haciendo corregir expedientes en los que no se había identificado ningún riesgo.

Thomas había aclarado las reglas de preparación: un lote marcado como «bloqueado» impedía el envío; las demás situaciones seguían siendo visibles y continuaban por el circuito habitual, con los controles existentes.

—¿Estás seguro de que no cambia el comportamiento de otros módulos? —preguntó Vivien.

—En los casos que ya existían, sí. He añadido pruebas de regresión. No he tocado las reglas de bloqueo.

Vivien no tenía motivos para dudar de la prudencia de Thomas. A veces conocía excepciones que ni siquiera aparecían ya en la documentación histórica.

—Muy bien. Que alguien revise el cambio antes de seguir.

Thomas asintió.

Los dos cambios todavía no se habían probado juntos en un mismo escenario.

8

El problema apareció el viernes, en una instancia de pruebas integrada.

Una persona de soporte había preparado un recorrido completo: recepción de una materia prima, fabricación, alerta de un proveedor, identificación de los productos afectados y preparación de un envío.

Lucas seguía la demostración por videollamada. Thomas también estaba presente. Alice había pedido que la invitaran para observar cómo entendería el funcionamiento futuro alguien que no lo hubiera diseñado.

Durante los primeros minutos todo fue bien.

Apareció el número de lote. Se mostraron las fabricaciones. Se vio el aviso «en revisión».

Después, la persona de soporte pasó al módulo de expediciones.

El albarán seguía en preparación. Ningún mensaje impedía continuar.

—Un momento —dijo Alice—. ¿Estamos hablando del mismo lote?

Lucas se acercó a su pantalla.

—Sí. ¿Por qué?

—Aquí arriba está en revisión, pero en este otro módulo podríamos seguir como si no pasara nada.

Thomas abrió los detalles del pedido.

—No está bloqueado. Está en revisión. No es el mismo estado.

—Para mí —respondió Lucas—, «en revisión» significaba precisamente que no debíamos considerarlo validado hasta que alguien decidiera qué hacer.

Thomas se volvió hacia la cámara.

—Entonces, ¿por qué no lo marcaste como «bloqueado»?

—Porque no quería inventarme una prohibición general. Hay clientes que necesitan poder investigar sin detener

toda su actividad. Es su responsable de calidad quien debe decidir qué está realmente afectado.

—Y en mi módulo —replicó Thomas— tenemos lotes antiguos con datos incompletos. Si empiezo a bloquear todo lo que no esté expresamente validado, habrá pedidos que no avancen nunca.

Alice miró las dos pantallas.

Comprendía las razones de Lucas. También comprendía las de Thomas.

El problema era que un usuario podía leer «en revisión» en una parte del ERP y no encontrarse con ninguna advertencia que impidiera continuar en otra.

—Paramos aquí la demostración —dijo.

Nadie protestó. No se habían tocado datos reales de clientes y los cambios no habían llegado a producción. Precisamente para eso servía aquella prueba.

Pero el silencio posterior fue menos cómodo que el que sigue a una demostración exitosa.

Vivien se incorporó a la llamada cuando le explicaron lo ocurrido.

Dejó que Lucas describiera su decisión y que Thomas contara la suya. Hizo algunas preguntas sobre las pruebas y el alcance de cada modificación.

—¿Tenéis los dos vuestro Mini PRD? —preguntó.

—Sí —respondió Lucas.

—¿Y vuestras reglas de decisión?

—Sí —respondió Thomas—. No he cambiado el bloqueo que ya existía. Era una mejora local del proceso de preparación.

—Yo tampoco —añadió Lucas—. He añadido información para no ocultar las dudas. Habíamos escrito precisamente que eso era importante.

Vivien abrió ambos documentos, uno al lado del otro.

No estaban equivocados. Simplemente no hablaban de lo mismo.

El primero describía cómo señalar una incertidumbre en la trazabilidad. El segundo, cuándo debía detenerse un envío. Ninguno se había redactado para responder a la cuestión que conectaba ambos asuntos: ¿qué debía hacer el software cuando un lote era sospechoso, pero todavía no se había bloqueado expresamente?

—Tenemos un vacío entre los dos proyectos —dijo Vivien.

Thomas asintió.

—Técnicamente podemos elegir varios comportamientos. Pero tenemos que saber cuál queremos.

—Y hay que preguntarle al cliente qué hace realmente su responsable de calidad en ese caso —añadió Lucas—. Tengo hipótesis, no una regla válida para todos.

Alice intervino.

—Voy a aclarar qué comportamiento esperamos del producto. Pero no lo haré sola. Necesitamos a soporte, a Lucas, a Thomas y seguramente a un cliente para comprobar las consecuencias.

Vivien levantó la vista.

—Entonces os organizo una reunión mañana.

Alice vaciló.

—Puedes ayudarnos a reunir a las personas adecuadas, sí. Pero no te corresponde tomar la decisión por nosotros.

No se lo decía para dejarlo fuera. Sin embargo, Vivien sintió que la frase desplazaba algo.

Durante años había sido quien devolvía los problemas al circuito correcto. Esta vez, los implicados ya sabían que tenían que hablar entre ellos. No necesitaban que transformara el desacuerdo en una lista de tickets.

Necesitaban otra cosa: una decisión que ninguno de los dos desarrolladores podía tomar por su cuenta.

Al día siguiente, Alice empezó preguntando a cada uno qué intentaba evitar.

Lucas no quería que un responsable de calidad creyera contar con una trazabilidad completa cuando faltaban datos.

Thomas no quería que una regla imprecisa detuviera innecesariamente las operaciones de clientes con historiales incompletos.

Soporte recordó que una alerta de un proveedor no siempre significaba una retirada inmediata de productos. Algunos clientes tenían que identificar primero el alcance del problema, consultar a sus equipos de calidad y decidir qué medidas adoptar.

—Muy bien —dijo Alice—. Entonces puede que nuestro error haya sido intentar dar dos significados distintos al mismo estado.

Compartió una propuesta.

Un lote podía marcarse como **pendiente de revisión** sin que el ERP pretendiera adoptar, en lugar de la empresa cliente, una decisión normativa o de calidad. En cambio,

los procesos que pudieran utilizar o enviar ese lote debían mostrar la alerta y exigir el proceso de confirmación previsto, en vez de permitir que desapareciera.

Los datos históricos inciertos debían seguir apareciendo como tales. Y los casos en los que el programa no tenía la información necesaria no podían presentarse como si no existiera ningún riesgo.

—Eso —dijo Thomas— podemos comprobarlo en el módulo de expediciones. Pero quiero que probemos también los pedidos antiguos.

—Y quiero que revisemos los términos con el cliente —añadió Lucas—. «Pendiente de revisión» quizá no signifique lo mismo para su responsable de calidad.

Alice asintió.

No habían terminado el diseño. Por fin habían identificado quién debía decidir qué.

Anotó la decisión provisional y las preguntas abiertas en el contexto compartido, junto con sus razones y los casos que debían comprobarse. Añadió una condición a su tabla: cuando una decisión local modificaba el significado de un dato utilizado por varios módulos, ya no

se consideraba automáticamente verde, aunque el cambio de código fuera reversible.

—Deberíamos haberlo visto antes —dijo Lucas.

—Sí —respondió Alice—. Pero precisamente por eso hacemos pruebas. Ahora sabemos qué mirar la próxima vez.

Thomas estudió ambos documentos.

—También hará falta una prueba que recorra varios módulos. Tenemos muchas que comprueban cada comportamiento por separado. Muchas menos que comprueban qué pasa cuando se encuentran.

Vivien, que llevaba unos minutos escuchando, tomó nota.

Por primera vez desde el comienzo de la llamada, volvía a encontrarse con un problema que conocía bien: nadie tenía una visión completa de todas las dependencias.

Pero le llamó la atención cómo se estaba construyendo la solución. Alice no había dictado las pantallas. Lucas y Thomas no esperaban que alguien les dividiera las tareas. Soporte aportaba la experiencia del cliente. Su propio papel había consistido en señalar el vacío y facilitar la conversación.

Había sido útil.

Pero no se parecía casi nada a la reunión de entregas que organizaba cada lunes.

Por la tarde, Thomas recibió un mensaje de un desarrollador con menos experiencia del otro equipo.

No entendía por qué un antiguo proceso de existencias realizaba dos comprobaciones aparentemente idénticas.

El agente de programación le proponía eliminar una.

Thomas abrió el archivo.

—No toques eso antes de que lo miremos juntos — respondió.

Pasaron media hora compartiendo pantalla. La segunda comprobación cubría un caso heredado de una antigua evolución del producto, mal explicado en el código.

Thomas ya no recordaba exactamente a qué entrega se remontaba, pero sabía dónde buscar.

—Tu herramienta no se ha inventado una respuesta equivocada —explicó—. Ha visto dos controles parecidos y ha propuesto simplificar. Sin conocer la historia del programa, entiendo por qué.

—¿De verdad nunca utilizas la IA para programar? — preguntó el desarrollador.

—No para que me proponga cambios a ciegas en asuntos como este.

—Pero acabas de pasar treinta minutos con algo que Lucas quizá habría comprobado en cinco.

Thomas reflexionó.

—Puede ser. Y quizá Lucas habría encontrado lo mismo. Lo importante es que comprendas por qué existe el control. La próxima vez sabrás hacer una pregunta mejor, tanto a tu herramienta como a mí.

El desarrollador preguntó si podía documentar la explicación.

—Por supuesto. Estoy harto de ser el único que lo recuerda.

Thomas colgó, sorprendido por sus propias palabras. Durante años, su conocimiento de las excepciones lo había convertido en alguien imprescindible. A veces temía que la IA hiciera menos valiosa aquella experiencia. La conversación le había mostrado otra cosa: podía ayudar a los demás a desarrollar el criterio que les faltaba, en lugar de guardarse todas las respuestas.

Abrió el documento compartido y escribió unas líneas.

En la reunión del lunes, Hubert preguntó a Alice cómo iba el experimento.

Ella le contó las dos decisiones incompatibles, la prueba que había revelado el problema y la corrección de las reglas.

—O sea, que habéis dejado decidir a dos desarrolladores y han estado a punto de generar una incoherencia —resumió Hubert.

—Sí.

—No parece muy tranquilizador.

—Lo detectamos antes de que nada llegara a producción —respondió Alice—. El problema no es que dos personas hayan tomado decisiones. Es que ninguna de nuestras reglas decía cómo debían encajar sus decisiones entre sí.

Hubert examinó las notas del incidente.

—Antes seguramente habríamos organizado otra reunión de validación.

—Puede ser. Pero otra reunión no habría sacado necesariamente el problema a la luz. Necesitábamos a alguien que probara el recorrido completo y responsabilidades claras cuando una decisión afectara a varios ámbitos.

—¿Y ahora quién decide?

—Lucas se encarga de su asunto. Thomas lleva el suyo y nos ayuda con los riesgos técnicos. Yo soy la garante de la coherencia del producto. En los casos delicados consultamos a los especialistas correspondientes. No podemos delegar todo eso a una herramienta.

Hubert asintió lentamente.

—¿Y Vivien?

Alice tardó unos segundos en responder.

—Nos ayudó a poner en contacto a los dos equipos y entender dónde estaba el bloqueo. Pero no tenía por qué repartir las tareas ni decidir el comportamiento del producto.

Hubert pareció pensar en otra cosa.

—Lo vi el viernes —dijo—. Me enseñó el nuevo seguimiento de los proyectos. Su informe es muy bueno. Pero creo que la máquina podría preparar el ochenta por ciento de lo que me contó.

—Él también lo sabe —respondió Alice.

—Lo que no sé es qué debemos concluir de eso.

Alice no tenía una respuesta preparada.

—No que haya dejado de ser útil —dijo al fin—. Acabamos de ver que sabe hacer que las personas trabajen juntas.

Solo que ya no es necesariamente un puesto por el que tenga que pasar todo.

Hubert se quedó mirando la pantalla.

—Tendremos que hablar con él.

Vivien recibió la invitación para el día siguiente.

Lo primero que pensó fue en una reorganización. Desde hacía semanas tenía cada vez más huecos libres en la agenda. Los desarrolladores llevaban sus propios asuntos. Las herramientas recopilaban automáticamente parte de la información que él antes pasaba las tardes reuniendo. Seguía interviniendo en dificultades humanas, ayudando a incorporarse a los nuevos y desbloqueando situaciones complejas, pero esas intervenciones ya no llenaban sus días como antes.

Le había gustado sentirse indispensable para las entregas de Malorean.

No estaba seguro de que le gustara descubrir que ahora lo era menos.

Sin embargo, la reunión empezó con una pregunta que lo sorprendió.

—¿Qué te gustaría seguir haciendo, Vivien? —preguntó Hubert.

—Mi trabajo —respondió.

Hubert sonrió.

—Precisamente eso es lo que estamos intentando entender.

Vivien miró a Alice, invitada para hablar de cómo funcionaban los equipos, y después volvió hacia Hubert.

—No quiero que finjamos. Los equipos ya no necesitan que les pregunte todas las semanas cómo van. Y tampoco quiero pasarme el tiempo releyendo resúmenes que un agente ya ha preparado.

—¿Qué siguen necesitando? —preguntó Alice.

—Los conflictos que no quedan escritos en ninguna parte. Las personas que no se atreven a decir que están perdidas. Los nuevos que no saben a quién acudir. Las decisiones que afectan a varios equipos. Y, a veces, alguien que simplemente se tome el tiempo de escuchar.

Hizo una pausa.

—Pero eso no significa que todas esas cosas tengan que seguir pasando por mí.

Hubert se enderezó.

—No. Y habrá que dejar muy claro quién responderá de ellas si cambia tu función. No quiero que eliminemos un

título y descubramos después que nadie se ocupa de las personas.

Vivien asintió.

Hablaron del acompañamiento de los desarrolladores, de las decisiones transversales y de lo que debía seguir supervisando la dirección. Alice explicó cómo Thomas ayudaba a sus compañeros a comprender mejor el código histórico.

No salió ninguna solución definitiva de aquella llamada.

Cuando terminó, Vivien permaneció unos momentos mirando el calendario.

Se sentía menos amenazado que el día anterior, pero tampoco estaba tranquilo. Su profesión no había desaparecido. Se estaba descomponiendo en varias responsabilidades, algunas de las cuales podían asumirse de otra manera.

Esa tarde cerró el ordenador antes de lo habitual.

Por una vez, no tenía ningún retraso de entrega que recuperar.

Y todavía no sabía qué iba a hacer con aquel tiempo recuperado.

8

10 — ¿Hará la IA desaparecer los niveles intermedios de las empresas?

Hoy un equipo puede consultar directamente sus decisiones, actuar dentro de unas reglas conocidas y confiar parte del seguimiento a agentes. Necesita con menos frecuencia a un responsable que le transmita la información u organice cada etapa del trabajo.

Entonces, **¿qué será de los niveles jerárquicos que desempeñaban esa función?**

La respuesta parece evidente: si la IA asume la coordinación, los mandos intermedios dejan de ser útiles. Sin embargo, un responsable no se limita a transmitir instrucciones. Resuelve desacuerdos, acompaña a las personas, conecta equipos y se encarga de los problemas que no encajan en ningún procedimiento.

Por tanto, la verdadera cuestión no es tanto cuántos puestos van a desaparecer, sino comprender **adónde irán sus funciones y su poder de decisión**.

Una jerarquía construida para coordinar las decisiones

Un cliente importante comunica un problema. Soporte se lo transmite a su responsable, que contacta con

Producto. Producto pide una estimación a Ingeniería. Varios responsables reúnen la información, comparan las urgencias y, si es necesario, la decisión sube otro nivel.

Este recorrido no tiene por qué ser absurdo. Cada responsable dispone de una parte del contexto y puede comprometer a una parte de la empresa. Se vuelve pesado cuando esa cadena es la única manera de comprender un problema y decidir.

Históricamente, la jerarquía permitió crecer a las empresas a pesar de que la información estuviera fragmentada. Los niveles intermedios traducen los objetivos de la dirección en prioridades, coordinan equipos, vigilan las dependencias y resuelven conflictos sobre recursos. También participan en la evaluación de los empleados, su desarrollo profesional y la resolución de problemas humanos.

Pero no todas estas actividades tienen la misma naturaleza. Preparar un informe de incidente es diferente de decidir renunciar a una oportunidad comercial. Hacer seguimiento de un plazo no exige el mismo criterio que resolver un conflicto o ayudar a alguien a progresar.

Ya había defendido en Malorean una distinción parecida al hablar del Product Owner: un puesto puede perder su razón de ser como simple intermediario sin que

desaparezca la necesidad de la función que debía desempeñar. Hay que cuestionarse el trabajo realizado, no únicamente el título de quien lo realiza. [1]

Si se reducen los niveles intermedios, ¿quién asumirá las funciones que desempeñaban?

El trabajo a distancia ya había cambiado las reglas de la coordinación

La IA no ha inventado esta pregunta. Las empresas que trabajan a distancia, a veces en varios países y husos horarios, han tenido que aprender a decidir sin depender de las conversaciones de pasillo o las reuniones improvisadas.

En GitLab, el principio **handbook-first** consiste en hacer accesibles por escrito las reglas y el contexto. Cualquiera puede proponer una modificación; los responsables de cada ámbito deciden si se incorpora. Así, un empleado puede encontrar información sin esperar a que un manager se la transmita. [2]

Alan describe una práctica distinta, pero complementaria. Las decisiones importantes se debaten en hilos de GitHub: una persona toma la decisión y la pone en práctica, otra sigue respondiendo del éxito del asunto y

se consulta o informa a los interesados. El debate puede estar abierto sin necesidad de unanimidad. [3]

En 37signals, creadora de Basecamp, buena parte de la coordinación se realiza mediante intercambios escritos, balances e información conservada con los proyectos, en lugar de dispersarse en conversaciones efímeras. [4]

No estamos hablando de tres empresas sin management. Son tres maneras de **reducir la dependencia de una persona que concentra la información o tiene que recordar continuamente lo que se ha decidido**.

Este precedente importa para los agentes de IA. Un sistema puede consultar una regla, recuperar una decisión e identificar a un responsable siempre que la organización haya hecho explícita esa información. Una decisión que solo existe en la cabeza de un manager sigue siendo inaccesible tanto para sus compañeros como para los agentes.

Pero conviene no idealizar el trabajo a distancia. Un estudio realizado en Microsoft con más de 61 000 empleados observó, entre otras cosas, que las redes de colaboración se volvieron más aisladas entre sí tras el paso al teletrabajo. Escribir más no garantiza que se comparta el contexto ni que los equipos colaboren. [5]

La verdadera herencia del trabajo a distancia no es la desaparición de la oficina: es la posibilidad de organizar parte del trabajo alrededor de información y decisiones accesibles sin intermediarios permanentes. La IA puede apoyarse en este avance. No lo crea por sí sola.

La IA empieza a asumir parte de ese trabajo

Un documento conserva una regla; un cuadro de mando muestra un retraso. Un agente puede ir más allá: relacionar varias informaciones, preparar un estado de avance, detectar una contradicción, proponer cómo abordarla o ejecutar una acción autorizada.

Son precisamente actividades que ocupaban parte del tiempo de los managers: consolidación, informes, seguimiento y preparación de decisiones. Su coste puede disminuir. Pero no está demostrado que desaparezcan.

Un análisis exploratorio de HEC Paris, basado entre otras cosas en 45 entrevistas, sugiere que las tareas de gestión estructuradas y basadas en datos son más susceptibles de automatización que aquellas que dependen de la confianza, las relaciones humanas y el juicio. No mide cuántas horas de management se han

eliminado realmente ni cuántos puestos han dejado de ser necesarios. [6]

Un experimento realizado en Procter & Gamble también muestra que, en tareas de innovación de producto, una persona que trabaja con IA puede obtener resultados comparables a los de un equipo de dos personas sin IA. La herramienta ayuda, en particular, a combinar perspectivas de profesiones distintas. Pero facilitar una colaboración puntual no significa sustituir de forma duradera a un responsable. [7]

En otro experimento, con 7 137 empleados de 66 empresas, quienes utilizaban regularmente un asistente de IA ahorraron unas dos horas semanales de correo electrónico durante la segunda mitad del estudio. Sin embargo, los investigadores no observaron una transformación significativa de la composición de las tareas en seis meses. El estudio trataba sobre un asistente de oficina, no sobre agentes que ejecutaran por su cuenta procesos completos. [8]

Esta diferencia es decisiva: **una mejora individual de productividad todavía no equivale a una transformación organizativa**.

Para que un nivel jerárquico resulte menos necesario, la información tiene que llegar de verdad a los equipos,

deben cambiar los derechos de decisión y las situaciones poco habituales necesitan un lugar donde resolverse. Ya no es solo una cuestión de herramientas, sino de organización.

Tres empresas, tres trayectorias muy diferentes

No todas las organizaciones responden igual. Tres casos permiten distinguir una reorganización ya puesta en marcha, una ambición diseñada alrededor de la IA y las dificultades de una transformación acelerada.

Bayer: una jerarquía realmente reducida

Con su modelo **Dynamic Shared Ownership** (DSO), Bayer ha trasladado parte de las decisiones a equipos multidisciplinares. En la carta a los accionistas correspondiente a 2025, su máximo responsable afirma haber eliminado hasta seis niveles organizativos y reducido aproximadamente dos tercios de los puestos de management. También declara que las decisiones son más rápidas y que los plazos de desarrollo se han acortado. [9]

La magnitud merece atención, aunque estos resultados los comunica Bayer. La transformación implica cambiar las responsabilidades cotidianas, no solo los nombres del organigrama: los equipos toman más decisiones locales,

los responsables dedican más tiempo a acompañarlos y se reorganizan algunas funciones de coordinación.

Hay, sin embargo, una precisión que cambia cómo debemos interpretar este caso: **DSO comenzó antes del auge de los agentes de IA generativa**. Bayer demuestra que un gran grupo puede reducir considerablemente su jerarquía; no demuestra que la IA sea la causa.

Block: hacer circular el contexto sin pasar por toda la jerarquía

Block va más lejos en el lugar que quiere dar a la IA. En un texto firmado conjuntamente por Jack Dorsey y el inversor Roelof Botha, sus dirigentes describen una organización apoyada en un sistema de información compartido.

Aspiran a que un empleado comprenda una situación y pueda actuar sin esperar a que un responsable le transmita el contexto. Los autores distinguen a los colaboradores que ejecutan, a los responsables de resultados transversales (DRI) y a los responsables que siguen contribuyendo al trabajo mientras acompañan a los demás, llamados **player-coaches**. [10]

Así, perdería importancia el manager que actuaba fundamentalmente como intermediario. En cambio, seguirían siendo necesarias la formación de las

personas, la responsabilidad por los resultados y las decisiones de dirección.

Se trata de una ambición descrita por la empresa y su inversor, no de una evaluación independiente que demuestre el éxito duradero del modelo.

Meta: cuando la producción supera la capacidad de control

El contrapunto lo ofrece una investigación de Reuters sobre los planes de reorganización de Meta en 2026: equipos más pequeños, un papel importante para la IA y la reducción prevista o iniciada de algunos niveles. Finalmente se abandonó una segunda oleada de reestructuración.

Según documentos internos citados por Reuters, los cambios en el software y la infraestructura internos aumentaron mucho más deprisa que las nuevas funcionalidades entregadas a los usuarios. Al mismo tiempo crecieron los incidentes técnicos y el tiempo dedicado a resolverlos. Meta cuestionó o matizó algunos aspectos del reportaje. [11]

Estas cifras no permiten atribuir los incidentes a la IA ni a la reducción de managers: estaban ocurriendo varias transformaciones a la vez. Pero plantean una pregunta decisiva: ¿de qué sirve producir más si la empresa ya no consigue verificar, estabilizar y asimilar lo que produce?

Bayer demuestra que es posible aplanar la jerarquía, Block plantea una versión diseñada alrededor de la IA y Meta recuerda los riesgos de una reorganización en la que la producción crece más deprisa que la coordinación. Ninguna de estas trayectorias anuncia por sí sola el futuro del management.

Las funciones de management no desaparecen con los puestos

Pongamos un ejemplo: un cliente comunica un fallo de software. Mientras el problema sea conocido y las reglas para resolverlo estén claras, el equipo puede actuar sin esperar una sucesión de autorizaciones. Con datos fiables, puede investigar, aplicar una corrección e informar a los demás.

Pero entonces dos diagnósticos se contradicen. El agente recomienda una acción y un desarrollador propone otra. La corrección podría afectar a un servicio vecino.

En ese momento alguien debe hacer explícito el desacuerdo, reunir las competencias necesarias y decidir qué riesgo asumir. La autonomía no ha eliminado la coordinación; ha cambiado la forma de ejercerla.

Un experimento de Martin Krzywdzinski y sus coautores ofrece una pista interesante. En una cadena de

producción simulada, 87 parejas afrontaron fallos con ayuda de un sistema de IA que a veces se equivocaba. Los equipos autoorganizados se comunicaban más y resolvían mejor los problemas que aquellos con un jefe designado. El análisis destaca el papel decisivo de la comunicación. [12]

Dos personas en un laboratorio no representan a una gran empresa. Pero el estudio recuerda una distinción valiosa: **un equipo puede trabajar sin un jefe formal; no puede prescindir de las tareas de coordinación.** Estas tareas pueden asumirlas los compañeros, un especialista, una persona con mandato para decidir o un responsable jerárquico. La elección depende de las competencias, los riesgos y los desacuerdos.

También queda todo aquello que ningún informe automático puede hacer por sí solo: recibir a alguien, ayudarle a progresar, resolver una tensión, acompañar una decisión difícil o reconocer su trabajo. No todos los managers desempeñan bien estas funciones y se pueden repartir de otra manera. Pero alguien necesita seguir teniendo la responsabilidad y el tiempo para realizarlas.

Gallup observa que utilizar IA no elimina la inquietud por el empleo y que percibir apoyo en el trabajo se asocia con una preocupación menor. Esto no demuestra que sea

necesario mantener al manager tradicional, pero nos recuerda que una organización también es una comunidad humana. [13]

Y, por último, ¿quién formará a las personas que mañana tendrán que tomar decisiones difíciles?

En Malorean he señalado un riesgo para los desarrolladores jóvenes: cuando una IA escribe inmediatamente el código y corrige los errores, puede eliminar ocasiones para aprender a depurar, comprender un fallo y desarrollar el propio criterio. Es una preocupación que conviene estudiar, no una prueba de una pérdida generalizada de competencias. [14]

El mismo problema afecta a los futuros responsables. También se aprende presenciando cómo se resuelve un desacuerdo, recibiendo comentarios y observando cómo un equipo sale de un conflicto. Una empresa que elimina niveles debe prever otras maneras de transmitir esa experiencia.

Al final, hay varios destinos posibles para cada función: automatizarla, asumirla dentro del equipo, transferirla a un especialista o elevarla a la dirección. **Una tarea puede desaparecer, cambiar de manos o añadirse al trabajo de otra persona.**

Por tanto, para evaluar la eficacia de una organización más plana no basta con contar los puestos eliminados. Hay que comparar el seguimiento que se evita con las nuevas cargas de supervisión de agentes, resolución de incidentes, acompañamiento y formación. Todavía no disponemos de un balance neto sólido de esta redistribución.

¿Es realmente más autónoma una organización más plana?

Queda una última pregunta, menos visible en los organigramas: **¿quién decide cuando desaparecen los intermediarios?**

Podemos imaginar equipos mejor informados, libres de resolver más problemas sin elevar cada elección. Pero también podemos imaginar lo contrario: menos jefes sobre el terreno y más decisiones controladas directamente por la dirección a través de los sistemas.

Esta tensión ya existía antes de la IA generativa. En una investigación sobre empresas industriales, Nicholas Bloom y sus coautores muestran que las herramientas que proporcionan más información a los empleados se asocian con una mayor autonomía local. En cambio, las que facilitan la comunicación con los superiores pueden

favorecer la centralización: resulta más sencillo pedir instrucciones a la cúpula. [15]

Este precedente no demuestra nada sobre los agentes actuales. Sin embargo, ayuda a comprender un mecanismo: **difundir los datos no es lo mismo que distribuir el poder.**

Los sistemas de gestión algorítmica hacen más concreto este riesgo. Pueden asignar tareas, proponer prioridades, medir el rendimiento y activar alertas. La investigación muestra que estas herramientas pueden desempeñar ciertas funciones de dirección o evaluación, incluso cuando siguen existiendo managers humanos para aplicarlas o gestionar excepciones. [16]

Un sistema no aparece como superior jerárquico en el organigrama. Sin embargo, si decide qué tareas recibe cada persona, con qué criterios se evalúan los resultados y en qué condiciones se autoriza una acción, estructura parte de la autoridad.

Por tanto, debemos mirar las decisiones que hay detrás de las pantallas: ¿quién define los objetivos de los agentes? ¿Quién elige las reglas y puede cambiarlas? ¿Quién puede cuestionarlas? Y, cuando un algoritmo se equivoca, ¿quién tiene realmente el poder de deshacer lo que ha provocado?

Un equipo puede tener una capacidad de ejecución enorme y ser incapaz de cambiar sus prioridades. Otro puede conservar un manager y, aun así, decidir de verdad sus medios y sus asuntos locales.

Ya he descrito en Malorean esas «decisiones ambiguas» que producen una apariencia de acuerdo sin hacer explícitos la elección, la renuncia aceptada o quién decide. Eliminar un nivel no cambia nada si sigue siendo igual de difícil identificar dónde reside el poder. [17]

Así, podemos imaginar una paradoja: **menos jerarquía visible, pero más control centralizado en las herramientas**.

No es inevitable. Los mismos sistemas pueden ayudar a los equipos a decidir más localmente y hacer más transparentas las reglas que distribuyen el poder. Todo depende de lo que la empresa elija delegar de verdad, no del número de casillas eliminadas de su organigrama.

Lo que realmente podría desaparecer

¿Hará la IA desaparecer los niveles intermedios de las empresas?

Algunos niveles podrían dejar de ser tan necesarios. Pero su desaparición no sería automática ni equivaldría al fin del management.

Las funciones que consisten principalmente en transmitir información, consolidar informes de avance u organizar aprobaciones repetitivas pueden perder importancia. Otras, basadas en resolver desacuerdos, gestionar conflictos, transmitir experiencia y asumir responsabilidades, tendrán que realizarse de otra manera.

Bayer muestra que es posible reducir masivamente los niveles sin que la IA sea una causa identificada. Block propone confiar más coordinación a un sistema compartido. Meta nos recuerda que acelerar la ejecución sin medir la fiabilidad y la carga humana puede debilitar la organización. Todavía no hay pruebas de una desaparición generalizada del management debida a los agentes.

Sin embargo, el desplazamiento podría ser profundo. Un responsable tendría menos veces que actuar como paso obligatorio para acceder a información o conseguir una aprobación. Podría dedicar más atención a la coherencia, las situaciones excepcionales y el desarrollo de las personas. En otros casos, algunas funciones pasarían directamente a los equipos; otras quedarían integradas en el software o concentradas en manos de unos pocos responsables de las decisiones.

No tiene por qué ser el management lo que se vuelve innecesario. Lo que empieza a cuestionarse es la jerarquía como mecanismo obligatorio de coordinación.

Para un desarrollador, la consecuencia es concreta. Cuando el trabajo pasa por menos intermediarios, no solo recibe más medios para producir. Entra en contacto más directo con las necesidades, los compromisos y los resultados de los que habrá que responder.

Es el movimiento que hemos descrito a lo largo de este primer libro: aumenta la capacidad de actuar, pero también la necesidad de comprender, elegir y asumir responsabilidades.

Este desplazamiento no garantiza mejores decisiones ni organizaciones más humanas. Pero obliga a concebirlas de otra manera: compartir el contexto, hacer identificable el poder, preservar la transmisión de la experiencia y aceptar la responsabilidad sobre los resultados.

La cuestión no es solo qué nos permite producir ahora la IA, sino qué clase de empresa queremos construir alrededor de esta nueva capacidad.

En el último episodio volveremos a Malorean Corp. para ver cómo sus dirigentes y equipos viven, más allá del organigrama, las decisiones que han tomado.

Bibliografía

[1] Le PO n’est pas un métier, c’est une fonction — et l’IA n’y change rien, elle accélère — Malorean

- URL: **https://malorean.net/articles/2026-08-03-le-po-nest-pas-un-metier-cest-une-fonction.es.html**
- Ensayo del autor que distingue el trabajo útil de una función de la existencia de un puesto que sirve fundamentalmente como intermediario entre la decisión y la ejecución.

[2] Handbook First — GitLab

- URL: **https://handbook.gitlab.com/handbook/company/culture/all-remote/handbook-first/**
- Presenta la documentación como referencia compartida y las modificaciones debatidas públicamente, manteniendo responsables humanos de las decisiones.

[3] Comment prendre des décisions dans un environnement de travail asynchrone — Alan

- URL: **https://alan.com/fr-fr/blog/healthy-business/a/decisions-asynchrones-methode-alan**
- Explica cómo se debaten las decisiones por escrito, quién las toma, quién sigue siendo responsable y a quién se consulta o informa.

[4] The 37signals Guide to Internal Communication — 37signals

- URL: **https://37signals.com/how-we-communicate**
- Describe una comunicación interna predominantemente asíncrona, con intercambios escritos y un contexto conservado en el tiempo.

[5] The effects of remote work on collaboration among information workers — Microsoft Research

- URL: **https://www.microsoft.com/en-us/research/publication/the-effects-of-remote-work-on--collaboration-among-information-workers/**
- Analiza las redes de colaboración tras el paso al teletrabajo y señala un mayor riesgo de aislamiento entre grupos.

[6] AI Automates Routine Management and Raises the Value of Leadership — HEC Paris

- URL: **https://www.hec.edu/en/dare/tech-ai/ai--automates-routine-management-and-raises-value--leadership**
- Análisis exploratorio de tareas de gestión basado en entrevistas y evaluaciones de modelos, sin medir causalmente la eliminación de puestos.

[7] The Cybernetic Teammate: A Field Experiment on Generative AI Reshaping Teamwork and Expertise — NBER

- URL: **https://www.nber.org/papers/w33641**
- Experimento en Procter & Gamble sobre la contribución de la IA a tareas de innovación y a la combinación de competencias.

[8] Shifting Work Patterns with Generative AI — NBER

- URL: **https://www.nber.org/papers/w33795**
- Experimento aleatorizado en 66 empresas: ahorro individual de tiempo sin cambios significativos en el conjunto de tareas observados durante seis meses.

[9] Chairman's Letter — Bayer Annual Report 2025

- URL: **https://reports.bayer.com/annual-report-2025/en/to-our-stockholders/chairmans-letter.html**

- Bayer comunica una importante reducción de niveles y puestos de management en su transformación DSO, anterior a los agentes.

[10] From Hierarchy to Intelligence — Sequoia Capital / Block

- URL: **https://sequoiacap.com/article/from--hierarchy-to-intelligence**
- Jack Dorsey y Roelof Botha presentan una visión de coordinación mediante un sistema compartido, con DRI y player-coaches.

[11] Mark Zuckerberg had a bold plan to replace Meta staff with AI. Here's how it imploded — Reuters

- URL: **https://www.timeslive.co.za/news/sci-tech/2026-08-26-special-report-mark-zuckerberg-had--a-bold-plan-to-replace-meta-staff-with-ai-heres--how-it-imploded/**
- Investigación de Reuters reproducida por TimesLIVE sobre las tensiones organizativas y técnicas de un proyecto de transformación en Meta.

[12] How team organization influences the ability to solve automation failures — AI & Society

- URL: **https://link.springer.com/article/10.1007/s00146-025-02761-5**

- Experimento con parejas autoorganizadas y jerárquicas que afrontaron fallos de automatización, centrado en el papel de la comunicación.

[13] Using AI More Does Not Reassure Workers, Managers Do — Gallup

- URL: **https://www.gallup.com/workplace/713231/ai--not-reassure-workers-managers-do.aspx**
- Encuesta sobre la inquietud ante la IA, la percepción de respeto y el apoyo que sienten los trabajadores.

[14] L'IA ne devrait pas seulement servir à produire dix fois plus. Elle devrait aussi nous obliger à comprendre dix fois mieux ce que nous produisons — Malorean

- URL: **https://malorean.net/articles/2026-07-23-lia--produire-dix-fois-plus-comprendre-dix-fois-mieux.es.html**
- Reflexión del autor sobre las oportunidades de aprendizaje que se pierden cuando la IA realiza las tareas y correcciones que formaban a los desarrolladores jóvenes.

[15] The distinct effects of Information Technology and Communication Technology on firm organization — NBER

- URL: **https://www.nber.org/papers/w14975**

- Distingue los efectos opuestos del acceso a la información y las tecnologías de comunicación sobre la autonomía local al decidir.

[16] Algorithmic Management: From Technology to Politics and Theory — Weizenbaum Journal of the Digital Society

- URL: **https://ojs.weizenbaum-institut.de/index.php/wjds/article/view/4_3_9/163**
- Síntesis de las funciones de dirección y evaluación que desempeñan los sistemas algorítmicos y de las posibilidades de intervención humana.

[17] Pourquoi les organisations préfèrent les décisions molles — Malorean

- URL: **https://malorean.net/articles/2026-06-02--pourquoi-les-organisations-preferent-les--decisions-molles.es.html**
- Ensayo sobre los consensos aparentes que retrasan las decisiones, ocultan las renuncias asumidas y dificultan identificar a los responsables.

11 — Malorean Corp. — El tiempo recuperado

Dos meses después de la demostración del módulo de vacaciones, Hubert Félix Duca abrió la reunión del lunes sin preguntar cómo avanzaban los desarrollos.

—Bueno, ¿qué hemos aprendido la semana pasada?

Alice compartió su pantalla.

Tenía delante el resumen de los últimos comentarios de clientes, los resultados de los experimentos en marcha y algunas decisiones todavía abiertas. Un agente había preparado durante la noche una primera agrupación de la información. Ella la había revisado por la mañana, descartado varias relaciones engañosas y añadido dos preguntas que el agente no habría podido formular por su cuenta.

—Sobre la trazabilidad, nuestro cliente piloto puede localizar las fabricaciones afectadas mucho más deprisa cuando los datos están completos —empezó—. Pero quedan casos históricos en los que tenemos que mostrar claramente nuestras limitaciones. Vamos a probar otra presentación con su responsable de calidad.

Hubert asintió.

—¿Y con los demás clientes?

—Todavía es pronto para sacar conclusiones —respondió Alice—. Precisamente por eso no hemos generalizado el despliegue.

Bob tomó la palabra. Trabajaba en otro conjunto de asuntos de Producto y acababa de mantener varias conversaciones con los equipos de preventa y soporte.

—También tenemos tres peticiones que parecen similares —dijo—. El agente las ha reunido bajo una misma idea de funcionalidad. Pero detrás hay al menos dos problemas distintos. Me gustaría que aclarásemos el cliente objetivo antes de decidir si invertimos.

—¿Cuántos desarrolladores necesitas? —preguntó Hubert.

Bob sonrió.

—De momento, ninguno.

Hubert le devolvió la sonrisa.

Bob miró una notificación y añadió:

—Por cierto, soporte está probando desde ayer una nueva clasificación de peticiones recurrentes. Una persona del equipo la ha preparado con un agente, utilizando copias anonimizadas, para comprobar si la información llega antes a los interlocutores adecuados.

—¿Quién lo ha puesto en marcha? —preguntó Hubert.

—Soporte. Alice confirmó que el alcance seguía siendo interno y reversible, con una comprobación humana de los resultados.

—¿Tú lo sabías?

—Conocía la idea, sí. No los detalles de la prueba.

Hubert sintió el impulso de preguntar por qué nadie le había avisado antes de empezar. Se contuvo. Precisamente habían acordado que las iniciativas limitadas y reversibles ya no necesitarían su aprobación.

—Me diréis si funciona —dijo finalmente—. Y si alguien tiene que mantenerlo, quiero que sepamos quién.

No había tenido que autorizar la prueba. Seguía siendo responsable de los recursos que dedicarían a una posible generalización.

No hacía tanto, aquella respuesta habría parecido extraña en una reunión de Producto. Antes se trataba sobre todo de decidir qué debían desarrollar los programadores y en qué orden. Ahora Alice y Bob dedicaban más tiempo a elegir los problemas que merecían resolverse, comprobar la coherencia de las iniciativas e intervenir cuando una decisión superaba las reglas establecidas.

Los nombres de sus puestos no habían cambiado. En los documentos administrativos seguían siendo Product Managers.

Pero su trabajo diario ya no era exactamente el mismo que dos meses antes.

Quizá el cambio más visible fuera precisamente aquello que había dejado de llamar la atención.

Un desarrollador hablaba con un cliente. Otro proponía un Mini PRD para una mejora de su módulo. Una comercial preparaba una variante de demostración en el entorno previsto para ello. Alguien de soporte señalaba directamente una incoherencia en una regla de producto.

Ya no provocaba grandes debates sobre las fronteras entre profesiones.

Malorean no había anunciado que todo el mundo pudiera hacerlo todo. La empresa simplemente había dejado de dar por supuesto que una acción debía pasar obligatoriamente por quien ocupaba tradicionalmente cierto puesto.

Los dos equipos de cinco desarrolladores seguían existiendo. La mayoría de sus integrantes se encargaban ahora de algunos asuntos de principio a fin: comprender

el problema, proponer el planteamiento, hablar con las personas afectadas, desarrollar, probar y después hacer seguimiento de los resultados.

Algunos habían adoptado esta manera de trabajar más rápidamente que otros. Dos desarrolladores preferían seguir recibiendo acompañamiento cercano para definir el alcance. Otro era excelente resolviendo problemas técnicos complejos, pero no tenía ninguna gana de multiplicar las reuniones con clientes. Nadie intentaba convertirlos a la fuerza en copias de Lucas.

También seguía habiendo tickets.

Se utilizaban para coordinar una intervención, recordar una tarea concreta o hacer seguimiento de una corrección. Pero ya no se intentaba meter en ellos toda la razón de ser de un asunto, todas las decisiones anteriores y toda la información útil para un agente.

En su lugar, cada responsable de un asunto mantenía un espacio de contexto dentro de la herramienta documental común. Allí estaban los objetivos, las decisiones, las fuentes, los riesgos, las preguntas abiertas y los enlaces a lo que ya existía: tickets, presupuestos, comentarios de soporte, actas de reuniones, pruebas y, a veces, conversaciones antiguas que habían conseguido recuperar.

No hacía falta copiarlo todo. Lo importante era que cualquiera pudiera comprender por qué se había tomado una decisión, dónde comprobar un dato y quién tenía autoridad para cambiarlo.

El primer documento de Alice había servido de ejemplo. Bob adoptó el método, después los desarrolladores y luego varias personas de soporte, ventas e implantación. Muchos descubrieron, como Alice, que podían pasar una hora explicando oralmente a ChatGPT decisiones que habrían tardado días en documentar por su cuenta.

No todo lo que estaba en la cabeza de las personas se había vuelto accesible por arte de magia. Pero una parte de las razones, las excepciones y los conocimientos tácitos empezaba a estarlo.

El teletrabajo había hecho más natural ese movimiento. Desde la pandemia, Malorean ya había aprendido a escribir más para que compañeros repartidos por toda Francia pudieran trabajar sin esperar a conversar en un pasillo. La IA había acelerado la extracción y organización del conocimiento. No había inventado la necesidad de compartirlo.

Alice abrió una página titulada **Decisiones de Producto — reglas comunes**.

—La semana pasada volvimos a tener una confusión — explicó a Bob—. En la documentación de un equipo, «validado» significaba «revisado técnicamente». En la de soporte significaba «comportamiento confirmado con el cliente».

Bob suspiró.

—Esa palabra nunca ha significado lo mismo para todos.

—Precisamente. He añadido una referencia a la decisión que tomamos después de lo ocurrido con los lotes.

Unas semanas antes, Lucas y Thomas habían estado a punto de dejar coexistir dos comportamientos incompatibles entre trazabilidad y expediciones. Sus cambios se habían detenido durante las pruebas. Desde entonces, los equipos habían aprendido a desconfiar de los términos que parecían evidentes dentro de un módulo pero cambiaban de significado en otro.

El contexto había ayudado. Pero también había creado nuevos problemas.

Algunas páginas habían quedado desactualizadas. Se habían copiado decisiones provisionales como si fueran reglas definitivas. En cierta ocasión, un agente había

recuperado una antigua hipótesis comercial como si fuera un compromiso de la empresa.

Habían tenido que establecer una disciplina: cada contexto indicaba su responsable, sus fuentes, sus decisiones confirmadas y los puntos aún inciertos. Los cambios estructurales debían quedar registrados; la información sensible no estaba a disposición de todas las herramientas indistintamente.

—Podríamos haber evitado multiplicar las páginas —comentó Bob.

—Quizá. Pero antes también existían las contradicciones. Solo que estaban en nuestras cabezas y nadie podía verlas.

Alice releyó la última modificación de su compañero.

—Apruebo el principio. Pero no la redacción sobre los clientes alemanes: todavía no sabemos lo suficiente.

Bob corrigió la línea.

A menudo, el trabajo de Garante se parecía a eso. No consistía en aprobar cada pantalla, sino en impedir que una hipótesis se convirtiera inadvertidamente en una regla.

En el otro equipo, Thomas acababa de terminar una conversación con un desarrollador que se había incorporado hacía poco.

Había dedicado una hora a explicarle el código histórico del ERP. El nuevo desarrollador había utilizado un asistente para comprender un antiguo mecanismo de nóminas y después había preguntado por qué una excepción parecía contradecir la regla general.

Thomas había encontrado el motivo de aquella excepción en una antigua entrega.

—Si te quedas con el resumen, podrías pensar que hay que limpiar el código —dijo—. Pero antes de limpiar hay que entender qué estamos a punto de quitar.

—¿Podrías poner esto en el contexto del equipo?

—Eso estoy haciendo.

Thomas compartió la nueva nota. Había añadido la fuente, la explicación y dos casos de prueba.

Seguía sin confiar a los agentes la creación de sus cambios técnicos como hacía Lucas. Su escepticismo no había desaparecido. Sin embargo, utilizaba de buen grado los nuevos espacios de documentación y colaboraba más con los demás para transmitir su experiencia.

Vivien, que había acompañado la incorporación del nuevo desarrollador, llamó a Thomas un poco después.

—¿Qué tal va?

—Bien. Es curioso. Hace preguntas antes de borrar cosas.

—Eso tranquiliza.

—Y por fin he dedicado tiempo a explicar por escrito por qué existen ciertos controles.

Vivien sonrió.

—¿Entonces has cambiado de opinión sobre la IA?

—No. He cambiado de opinión sobre guardarme esas explicaciones para mí.

Hizo una pausa.

—Eso sí, hace falta tiempo. Si me piden que acompañe a los demás y aun así me llenan todos los días de tareas de desarrollo, no funcionará.

—Lo he hablado con Hubert. Hemos reservado tiempo de acompañamiento en la planificación. No es algo que vayamos a hacer solo cuando todo lo demás esté terminado.

Vivien sabía de qué hablaba. Durante años había visto a equipos anunciar que formarían a los nuevos «cuando tuvieran un poco de tiempo». Ese momento nunca llegaba.

Siguieron hablando de las necesidades del desarrollador y Vivien terminó la videollamada.

No había abierto ni una sola vez el cuadro de entregas.

A la hora de comer, Hubert pidió a Vivien que se conectara a una llamada.

—Tengo una pregunta —empezó—. ¿Cuánto tiempo dedicas hoy a hacer lo mismo que hacías el año pasado?

Vivien reflexionó.

—No el suficiente para justificar mi antiguo puesto a jornada completa, si es eso lo que quieres saber.

Hubert no respondió inmediatamente.

Durante mucho tiempo, Vivien había sido el punto de paso de dos equipos técnicos. Gestionaba las entregas, las dependencias, parte de las urgencias y las conversaciones entre desarrolladores y responsables de otras áreas. Cuando la organización era menos autónoma, su función ocupaba sobradamente sus jornadas.

Ya no ocurría así.

Cada asunto tenía una persona claramente responsable. Los desarrolladores tomaban buena parte de las decisiones locales. La información circulaba sin que él

tuviera que retransmitirla. Las herramientas preparaban los seguimientos y señalaban dependencias, aunque todavía había que comprobar lo que afirmaban.

Los problemas humanos, los conflictos y el acompañamiento no habían desaparecido. Vivien seguía ayudando a resolverlos. Pero ya no hacía falta concentrarlos en un puesto dedicado a supervisar diariamente las entregas.

—No me molesta tener menos reuniones inútiles —continuó—. Al contrario. Pero no he pasado todos estos años aprendiendo a dirigir equipos para acabar siendo alguien que comprueba si un agente ha rellenado bien una tabla.

—No te voy a pedir eso —respondió Hubert.

Habían preparado una transición. Las decisiones técnicas importantes correspondían a los Garantes competentes. Los referentes encargados del acompañamiento disponían de tiempo asignado. Hubert conservaba la responsabilidad sobre las decisiones de carrera profesional y las dificultades humanas que no podían dejarse en manos de un simple documento. Vivien se encargaría del traspaso de los últimos asuntos de coordinación.

Nada de aquello era perfecto. Habían tenido que intervenir en un caso en el que dos personas creían que la otra estaba haciendo seguimiento de un compañero con dificultades. Un equipo también había esperado demasiado una decisión porque nadie había designado a su responsable.

Cada vez, habían corregido las responsabilidades y la forma de hacerlas visibles.

—Hay algo que no quiero —dijo Hubert—. Que hagamos desaparecer tu puesto y finjamos que el trabajo que hacías no tenía ningún valor.

—Yo tampoco.

—Entonces tenemos que encontrar dónde puede servirnos ahora tu experiencia.

Vivien miró su taza.

—Creo que tengo una idea.

Unos días antes, su mujer había vuelto a hablarle de Alemania.

Ella era alemana. Llevaban algún tiempo hablando de la posibilidad de instalarse allí. Sin urgencia: tenían su vida en Francia, sus costumbres y sus amigos. Pero aquella

posibilidad volvía a surgir con frecuencia, cada vez con más detalles.

Vivien siempre había ido posponiendo el asunto.

Le gustaba demasiado su trabajo en Malorean. Conocía a los equipos, su historia, el programa y los clientes. Marcharse le habría dado la sensación de abandonar algo que había contribuido a construir.

Sin embargo, también le gustaban los retos. En aquel momento de su carrera se sorprendía a sí mismo con ganas de afrontar uno nuevo.

El sábado anterior habían estado mirando anuncios de viviendas en Alemania sin reconocerlo del todo. Después su mujer cerró el ordenador.

—Sabes —dijo—, tampoco hace falta que sigamos hablando de esto otros diez años.

Vivien se rio.

Pero el lunes todavía le daba vueltas.

—¿Alemania? —repitió Hubert.

Vivien asintió.

—Llevamos años hablando de ello en las reuniones de estrategia. Tenemos algunos clientes allí, pero son sobre todo filiales de grupos franceses. Aún no hemos

aprendido realmente a vender nuestro ERP a empresas alemanas.

—Es verdad.

—Y todos decimos que iremos cuando el producto esté preparado. Pero nunca sabremos qué le falta hasta que empecemos a hablar en serio con empresas de allí.

Hubert lo escuchaba con atención.

—¿Te gustaría encargarte de desarrollar ese mercado?

—Sí. Pero sin fingir que me he convertido en experto del mercado alemán de la noche a la mañana. Conozco nuestro producto, nuestras limitaciones, nuestros equipos y cómo lo implantan los clientes. Sé reunir a las personas alrededor de un problema complicado. Tendré que aprender lo demás.

—¿Has trabajado alguna vez en ventas?

—No como Julie o Philippe. Y no pienso explicarles cómo hacer su trabajo. Quiero colaborar con ellos, con implantación y con nuestros especialistas del negocio para entender las necesidades alemanas y ver si somos capaces de responder a ellas.

Vivien dudó.

—Mi mujer es alemana. Llevamos un tiempo hablando de irnos a vivir allí. Sería una oportunidad de hacer algo que realmente me apetece sin dejar Malorean.

Hubert guardó silencio un momento.

—Sabes que esto no es un simple proyecto de traducción, ¿verdad?

—Por supuesto. El ERP afecta a normas y procesos que cambian de un país a otro. Traducir las pantallas no basta. Tenemos que saber qué podemos prometer de verdad y qué habrá que desarrollar, adaptar o rechazar.

—Y podemos equivocarnos.

—Probablemente nos equivocaremos en algunas cosas. Prefiero descubrirlo con clientes que después de invertir un año en funcionalidades que nadie quiere.

Hubert miró su lista de proyectos.

Durante mucho tiempo habían hablado de Alemania como una aspiración para más adelante. Primero había que terminar los asuntos atrasados, contratar a más desarrolladores, liberar tiempo para adaptar el ERP, preparar demostraciones y formar a los equipos.

Algunas de esas dificultades no habían desaparecido. Pero otras habían cambiado de magnitud.

Preparar una demostración requería menos esfuerzo. Se podían probar variantes del producto sin movilizar sistemáticamente a los desarrolladores. Soporte y preventa producían con más facilidad sus documentos y compartían sus conocimientos. Los desarrolladores ya no carecían tanto de capacidad para explorar una posibilidad acotada.

Eso no significaba que se hubieran quedado sin trabajo. Significaba que podían optar por dedicar parte de su energía a algo distinto de absorber la cola de peticiones francesas.

—Prepárame una propuesta para una primera etapa —dijo Hubert—. No una promesa de apertura del mercado. Reuniones, preguntas a las que queremos responder, competencias que necesitarás y una manera de decidir si seguimos.

Vivien sonrió.

—¿Un Mini PRD para Alemania?

—No te burles. Si conseguimos definir así un asunto técnico, al menos podemos intentar formular las hipótesis de un proyecto de mercado.

—De acuerdo. Pero quiero conservar el derecho a descubrir que estábamos equivocados.

—Cuento con ello.

La semana siguiente, Hubert reunió a Alice, Bob, Vivien, Julie y Philippe.

Les presentó la propuesta sin empezar preguntando cuántas funcionalidades tendrían que desarrollar.

—Ahora tenemos una capacidad de la que no disponíamos hace dos meses. Me gustaría dedicar parte de ella a entender qué supondría tener una presencia real en el mercado alemán.

Julie fue la primera en reaccionar.

—Ya puedo preparar demostraciones en alemán. También podríamos contactar bastante pronto con posibles clientes. Con la IA podemos adaptar los materiales en unas horas.

—Los materiales, quizá —dijo Philippe—. Pero un directivo alemán que compra un ERP no lo elige simplemente porque nuestra presentación esté en su idioma.

Julie cruzó los brazos.

—No he dicho eso.

—Ya lo sé. Solo digo que tendremos que entender cómo eligen a sus proveedores, a quién consultan y qué

confianza necesitan. Todavía no conocemos suficientemente ese mercado.

Vivien intervino antes de que la conversación se convirtiera en un debate sobre herramientas.

—Precisamente eso es lo que quiero aprender. Julie, necesitaremos tu capacidad para preparar rápidamente propuestas y demostraciones. Philippe, necesitaremos tu experiencia con grandes cuentas y con esas conversaciones que no se resuelven con una pantalla bonita. Y el equipo de implantación conoce las preguntas que a veces los clientes descubren demasiado tarde.

Alice intervino.

—Por nuestra parte, tendremos que ser muy claros con los límites del producto. Podemos explorar deprisa, pero no prometer funciones que dependan de normas que aún no hemos validado.

Bob asintió.

—También tendremos que elegir qué empresas queremos comprender primero. Alemania no es un tipo de cliente. Si intentamos satisfacer a todo el mundo, repetiremos los errores que hemos aprendido a evitar aquí.

Hubert observó los rostros en su pantalla.

Le había pedido a Vivien que propusiera un enfoque. Nadie le había pedido que formara inmediatamente un equipo permanente, reservara toda la capacidad de desarrollo o prometiera una fecha de apertura comercial.

Sin embargo, varias personas ya habían empezado a identificar lo que podían aportar.

Se había convertido en una costumbre en Malorean.

—Muy bien —dijo Hubert—. Esta es mi decisión: vamos a financiar una primera fase de exploración. Vivien será su responsable. Tendrá que decirnos qué problemas de las empresas alemanas sabemos resolver ya, cuáles requieren una adaptación y cuáles no queremos asumir. Después decidiremos si invertimos más.

Y añadió:

—Y nadie quedará asignado a este proyecto solo porque una herramienta le haya ahorrado tiempo. Vamos a ver qué capacidad real puede liberar cada uno sin abandonar a los clientes que atendemos hoy.

Vivien anotó el último punto. Llevaba suficientes años gestionando planificaciones para saber que una capacidad liberada sobre el papel no siempre estaba disponible en la práctica.

Durante los días siguientes, habló con el equipo de preventa e implantación.

Parte de su trabajo había cambiado desde que podían prepararse demostraciones a partir de configuraciones reutilizables y desde que resultaba más fácil elaborar los resúmenes de las conversaciones con clientes. Dos personas le contaron que ahora dedicaban menos tiempo a rehacer los mismos materiales y que les gustaría participar en las primeras entrevistas en Alemania.

—Puedo ayudarte a preparar las preguntas sobre la implantación —propuso una de ellas—. Y acompañarte a algunas reuniones a distancia.

—¿Y tus clientes franceses? —preguntó Vivien.

—No puedo dejarlo todo. Pero puedo reservar tiempo si lo incluimos en nuestra planificación.

Otra persona, ocupada en varias puestas en marcha complicadas, le dijo en cambio que no tenía ninguna disponibilidad.

Vivien no insistió.

Prepararon una distribución provisional: algunos espacios reservados para comprender las expectativas de las empresas alemanas, reuniones con filiales que ya eran clientes cuando resultara útil y ninguna promesa de

dedicación permanente hasta haber medido el esfuerzo real.

El tiempo ahorrado no se había transformado por sí solo en crecimiento. Todavía había que decidir qué hacer con él y aceptar no llenar inmediatamente cada hora recuperada con las peticiones de ayer.

Por la noche, Alice recibió un mensaje de Lucas.

He visto algo sobre el proyecto de Alemania. Ya tenemos un mecanismo de configuración que podría ayudarnos a probar ciertas diferencias entre procesos. Puedo preparar un entorno exploratorio sin tocar el producto estándar. ¿Os interesa?

Releyó el mensaje.

Dos meses antes, una propuesta parecida probablemente habría provocado una discusión sobre la obligación de un desarrollador de esperar a que le confiaran un asunto. Ahora, la primera pregunta era más sencilla: ¿resultaba útil para la estrategia y dentro de qué límites podía avanzar?

Alice abrió el contexto de la primera fase alemana. Vivien ya lo había completado con las hipótesis, los responsables, los límites y las preguntas abiertas.

Contestó:

Sí, si lo defines con Vivien y no convertimos una hipótesis de negocio en un comportamiento estándar sin validación. Empieza por lo que queremos aprender, no por lo que sabemos construir.

Lucas respondió casi al instante.

Entendido. Preparo un Mini PRD.

Alice sonrió.

Recordó el viernes en que Lucas les había mostrado un módulo de vacaciones desarrollado en dos días. Entonces todos habían intentado entender cómo había podido producir tanto y tan deprisa.

Aquella noche, a nadie le sorprendía ya que pudiera construir rápidamente una demostración.

La cuestión era saber para qué la necesitarían.

Vivien terminó la jornada compartiendo con su mujer las fechas de las primeras reuniones previstas con empresas de Alemania.

Todavía no había firmado ningún contrato. Malorean no había abierto una oficina ni adaptado su ERP a todos los requisitos del mercado. Tendrían que descubrir los usos

locales, comprobar las limitaciones legales y normativas y aprender a convencer a empresas que no los conocían. Había muchas más incógnitas que certezas.

—¿Te da miedo? —preguntó su mujer.

—Un poco.

—Parecías tan seguro de ti mismo cuando coordinabas tus equipos.

Vivien sonrió.

—Había pasado años aprendiendo aquel trabajo.

—¿Y ahora?

Miró el documento que había empezado a preparar para su primera visita.

Preguntas. Hipótesis. Nombres de personas con las que quería reunirse. Una primera lista de cosas que no sabía.

—Ahora —respondió— vuelvo a tener algo que aprender.

Ella se acercó a mirar la pantalla.

—¿Y nosotros, qué hacemos?

—Vamos a poder hablar en serio de nuestra mudanza.

Ella le puso una mano en el hombro.

Vivien recordó su antiguo cuadro de entregas, las reuniones de los lunes y todo el trabajo que le había permitido mantener unidos los proyectos. Le había gustado hacerlo. No lamentaba aquellos años.

Tampoco lamentaba haber aceptado que los equipos ya no lo necesitaran de la misma manera.

En la pantalla aparecía confirmada su primera reunión en Alemania.

Para Malorean todavía no era una conquista ni un éxito.

Era un comienzo.

12 — Conclusión

Profesiones con menos fronteras, responsabilidades más visibles

Las profesiones ya no delimitan con tanta precisión lo que cada persona puede hacer. Un desarrollador puede explorar una necesidad, un PM puede construir una solución y un comercial puede crear un prototipo. La IA abre caminos que antes estaban reservados a otras especialidades. Eso no convierte sus conocimientos en intercambiables.

Ya no somos responsables únicamente de lo que ejecutamos por nuestra cuenta. Podemos asumir un asunto que atraviesa varias funciones mientras los Garantes velan por la coherencia de sus respectivos ámbitos. La capacidad de actuar se extiende; la responsabilidad no debe diluirse.

El riesgo tiene dos caras: seguir diciendo «eso no es cosa de mi profesión» o pensar «si puedo hacerlo, también puedo decidir por mi cuenta». Ni defender los territorios profesionales ni eliminarlos basta para organizar el trabajo.

La autonomía necesita un contexto compartido

Tener más libertad solo tiene sentido si cada persona dispone de la información necesaria para juzgar. Sin un contexto compartido, una decisión puede parecer acertada en su ámbito inmediato y, sin embargo, contradecir una necesidad, una decisión o una limitación de mayor alcance.

Por eso documentar no consiste en acumular actas. Consiste en hacer accesibles el problema que hay que resolver, el objetivo, las hipótesis, las razones de las decisiones, los límites y las reglas para pedir ayuda o elevar una cuestión. Poner todo esto por escrito permite actuar sin tener que buscar a cada paso a la persona que conserva la información. El contexto se convierte en una herramienta para decidir, tanto para las personas como para los agentes.

Aquí cambia también el papel de los Garantes. No inspeccionan todas las iniciativas: hacen comprensibles las reglas, velan por su coherencia e intervienen cuando una decisión supera los límites establecidos. Sin mecanismos de control, la libertad puede desembocar en el caos; si todo tiene que subir para ser aprobado, esa libertad no pasa de ser una promesa.

El management también evoluciona. Parte del seguimiento y de la coordinación puede resolverse de otra manera. Pero sigue habiendo personas a las que acompañar, experiencia que transmitir, conflictos que resolver y decisiones difíciles que asumir. Esas responsabilidades no desaparecen porque se transformen las tareas que antes les daban forma.

Hacer más de lo que realmente importa

Será tentador medir esta transformación por el número de funcionalidades desarrolladas o las horas ahorradas. Es cierto que la IA puede ayudarnos a producir más. Pero llenar de inmediato el tiempo liberado con nuevas tareas equivaldría a acelerar la forma de trabajar de siempre sin cuestionar para qué sirve.

También podemos dedicar esa capacidad a entender mejor las necesidades, comprobar nuestras hipótesis, mejorar lo que ya existe y renunciar a lo que no aporta valor. Nada de esto sucede automáticamente. Hay que elegir dónde invertir la atención y los recursos que hemos liberado.

Vivien, en Malorean Corp., no descubre únicamente que su antiguo trabajo ocupa menos tiempo. Descubre que puede poner su experiencia al servicio de otro problema,

sin garantías de éxito. El progreso no depende solo de lo que se ha automatizado, sino de las posibilidades que abre esa transformación.

Al principio de la historia, todos se asombraban de que Lucas construyera tan deprisa. Al final, la pregunta es otra: ¿para qué construir, para quién y con qué consecuencias?

El objetivo de la IA no es que hagamos cada vez más cosas. Es darnos los medios para hacer más de lo que realmente importa.