Scrum Manager®
Uncovering better ways





DESCUBRIENDO *AGILE CONTRACTS*

UNA APROXIMACIÓN A LOS CONTRATOS ÁGILES

Nuevos modelos legales para nuevos paradigmas

Versión 2 (2026)

Descubriendo Agile Contracts

Una aproximación a los contratos ágiles

Versión 2 (2026)

Scrum Manager ®

Fecha de la versión: agosto 2026.
Autor de la versión: equipo de Scrum Manager®.
Uncovering Better Ways SLU es la editora y propietaria de los derechos de distribución, que libera en los términos de la licencia Creative Commons-by-nd-nc 4.0.
Derechos registrados en Safe Creative. Nº de registro:

Índice de contenidos

INTRODUCCIÓN **3**
CAPÍTULO I: GESTIÓN DE PROYECTOS **5**
Gestión tradicional 6
Gestión ágil 7
Resultados de la gestión ágil 9
CAPÍTULO II: CONTRATOS ÁGILES **11**
Principios aplicados 14
Ventajas y desventajas 18
CAPÍTULO III: ELABORAR UN CONTRATO ÁGIL **20**
Definir el marco contractual 21
Elegir un modelo de contrato 21
Elementos comunes 22
CAPÍTULO IV: MODELOS Y CLÁUSULAS **27**
Modelos de contrato 28
Cláusulas 35
CAPÍTULO V: CONSEJOS **38**
Elaboración de contratos 39
Herramientas y estrategias de negociación 43
Contratos ágiles en la era de la IA 48
CONCLUSIÓN **52**
APÉNDICES **54**
Información, recursos y comunidad profesional 55
Redes sociales y profesionales 55
Mejora continua y control de calidad 55
Referencias bibliográficas 56

INTRODUCCIÓN

Propósito y contenidos del estudio

La gestión ágil de proyectos se caracteriza por su flexibilidad para adaptarse a las necesidades de mercados en constante cambio. Este aspecto hace que la elaboración de contratos, en los que cliente y proveedor buscan protegerse para obtener el máximo beneficio con el mínimo riesgo, tenga dificultades añadidas.

Este trabajo de investigación busca aclarar dudas sobre los contratos ágiles y su elaboración: en qué consisten, qué objetivos persiguen y en qué principios se basan. Parte de los contratos tradicionales, para analizar por qué pueden resultar inadecuados en ciertos proyectos.

El capítulo 1, «Gestión de proyectos», elabora algunos conceptos básicos sobre la diferencia entre gestión tradicional y ágil, necesarios para abordar los siguientes temas.

A continuación, en el capítulo 2, se define qué son los contratos ágiles, poniendo énfasis en comprender a qué necesidades dan respuesta. Se recomienda su lectura antes de pasar al capítulo 3, donde se habla de cómo elaborar contratos ágiles.

El capítulo 4 presenta una recopilación de todos los modelos y cláusulas considerados ágiles a día de hoy. Para su fácil consulta se han presentado en orden alfabético, con una breve descripción de cada uno y ventajas e inconvenientes que pueden suponer según las circunstancias.

Por último, en el capítulo 5, se entra más en detalle en estrategias y herramientas para la negociación y aplicación de contratos ágiles, en base a las fuentes bibliográficas consultadas.

CAPÍTULO I

GESTIÓN DE PROYECTOS

Resumen de los dos modelos de gestión de proyectos más populares en la actualidad. El modelo tradicional comparte espacio con la llamada «gestión ágil».

Gestión tradicional

Hasta hace poco en el ámbito empresarial lo más habitual era el *waterfall model* o modelo en cascada: una forma predictiva de gestionar proyectos basada en la planificación exhaustiva y orientada hacia los procesos. Su complemento legal son los contratos tradicionales como, por ejemplo, los de precio fijo o los contratos por tiempo y materiales.

Este tipo de contratos presentan una serie de limitaciones, reflejo del modelo waterfall. En palabras de Jeff Sutherland:

- **Los resultados tienen incentivos negativos:** el esfuerzo se recompensa por encima del valor que se aporta al proyecto y, dada la limitación ante el cambio, se disuade la mejora continua y se añade complejidad innecesaria. Esto daña la confianza entre las partes y genera expectativas poco razonables.
- **Se erosiona el valor del producto:** el cliente no consigue lo que realmente quiere porque se especifica el proceso en vez de dar una visión clara. La rigidez limita la posibilidad de innovar y, en caso de que existan cambios, éstos ocasionan enfrentamientos entre las partes y desvían la atención lejos del alcance. El tiempo y el esfuerzo se malgastan tratando de cumplir los procesos establecidos desde el principio en el contrato.
- **Se desmotiva el cambio:** los cambios en proyectos predictivos son más costosos, por lo que el proveedor suele no molestarse en buscar alternativas. Esto dificulta el aprendizaje y responder a tiempo a las condiciones del mercado.

Gestión ágil

El origen de la gestión ágil de proyectos podría situarse en las prácticas adoptadas en los 80 por empresas como Honda, Canon, Fuji, NEC, HP o Epson, descritas por Takeuchi y Nonaka (1986) en su influyente artículo *The New New Product Development Game*. Aunque la industria del software fue la primera en adoptarla, y la primera en popularizar diversas formas de implementar los principios ágiles. En las últimas décadas han ido apareciendo distintos marcos de agilidad, como XP (*eXtreme Programming*), Scrum, DSDM, Crystal, FDD (*Feature-Driven Development*), ASD (*Adaptive Software Development*) y Lean Software Development. Con el tiempo el panorama se ha decantado: hoy Scrum y Kanban son los marcos más extendidos, y para coordinar varios equipos han surgido marcos de escalado como SAFe o LeSS.

Cuando hablamos de agilidad, es inevitable mencionar el *Manifiesto Ágil*. En marzo de 2001, diecisiete críticos de los modelos de producción basados en procesos fueron convocados por Kent Beck para una reunión, en la que se acuñó el término «métodos ágiles». Éstos son la alternativa que propusieron a las metodologías tradicionales y a las carencias que habían observado en ellas. Resumieron sus ideas en cuatro postulados:

- **«Valorar a individuos e interacciones sobre procesos y herramientas».** Este es el valor más importante del manifiesto, según el cual los procesos deberían ser sólo una guía de operación. Las herramientas mejoran la eficiencia, pero los proyectos ágiles requieren talento y personas motivadas que lo aporten.
- **«Valorar el software funcionando sobre la documentación extensiva».** No se da por inútil la documentación, sólo la innecesaria. En muchas ocasiones los documentos legales son obligatorios, pero su relevancia debe ser mucho menor que el producto final. Comunicarse a través de documentos no genera tanto valor como la comunicación directa entre personas y la interacción con prototipos del producto.
- **«Valorar la colaboración con el cliente sobre la negociación contractual».** Las prácticas ágiles están indicadas para productos poco detallados al principio para evitar que el resultado final tenga menos valor. Recordemos que el objetivo de un proyecto ágil es proporcionar el mayor valor posible al producto. Por ello es mucho más adecuado colaborar con el cliente y buscar su compromiso, de forma que se facilite la comunicación y este pueda, por ejemplo, sugerir cambios desde el primer momento.
- **«Valorar la respuesta ante el cambio sobre el seguimiento de un plan».** Para desarrollar productos de requisitos inestables, que cambian y evolucionan de forma rápida y continua, resulta mucho más valiosa la capacidad de respuesta. Es muy importante saber anticiparse a los cambios o adaptarse a ellos.

De los postulados surgen los doce principios ágiles:

Seguimos estos principios:

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

El software funcionando es la medida principal de progreso.

Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Manifiesto ágil

Resultados de la gestión ágil

El éxito de los proyectos ágiles ha hecho que cada vez más empresas los adopten, hasta convertir la agilidad en la forma dominante de gestionar el desarrollo de software. Que el enfoque importa está fuera de discusión: en el estudio *Agile Pulse* 2022 de BearingPoint, el 96 % de las personas participantes valora como alta su relevancia futura. Ahora bien, quererlo no es lograrlo: ese mismo estudio observa que, de media, las organizaciones no alcanzan ocho de cada nueve objetivos que perseguían al adoptar la agilidad (BearingPoint, 2022). La conversación, en otras palabras, ha dejado de girar en torno a si conviene ser ágil para centrarse en cómo obtener de ello valor real.

Esa misma tensión aflora en el *18th State of Agile Report* (Digital.ai, 2025): la adopción de IA en entornos ágiles saltó del 68 % al 84 % en un solo año, pero solo el 49 % de las organizaciones cuenta con mecanismos de gobernanza para ella, y el 76 % declara una presión creciente por demostrar el retorno de su inversión en agilidad. Conviene leer estas cifras como señal de tendencia —la muestra de 2025 fue reducida, unas 349 respuestas— más que como medida exacta.

La evidencia sobre el rendimiento comparado de ambos enfoques es más resbaladiza de lo que suele presentarse. Es habitual citar el informe CHAOS de The Standish Group (2015), según el cual los proyectos ágiles tendrían tasas de éxito notablemente superiores a los de cascada (Fig. 1). Pero esas cifras deben tomarse con cautela: su metodología ha sido cuestionada por la dificultad de reproducir sus resultados y por una definición estrecha de «éxito» —cumplir plazo, coste y alcance— que penaliza precisamente lo que la agilidad relativiza (Eveleens y Verhoef, 2010).

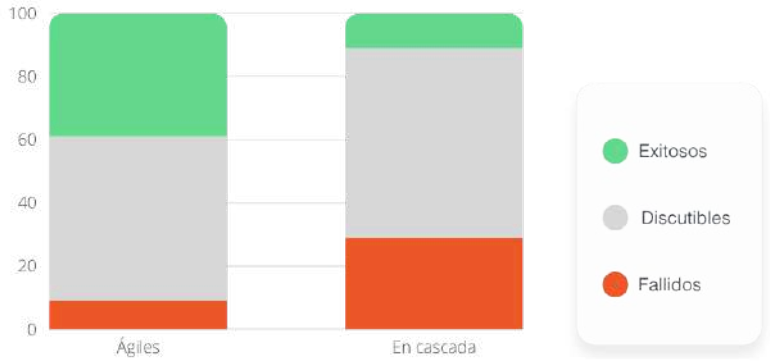


Fig. 1. *Gráfico comparativo según los datos del informe CHAOS de The Standish Group (2015), que compara proyectos ágiles y proyectos en cascada*[1].

A raíz de este nuevo tipo de gestión, surge la necesidad de nuevos modelos de contrato. Los contratos tradicionales son reflejo de los modelos de gestión junto a los que han evolucionado: son predictivos, orientados a procesos y con restricciones para los cambios. Se ajustan bien a proyectos que ponen énfasis en la planificación y la

[1] Desde 1994, The Standish Group publica el informe CHAOS, con estudios sobre el estado de más de 50.000 proyectos.

documentación. Difícilmente van a satisfacer las necesidades de un proyecto adaptativo, orientado a las personas y al cambio.

Por tanto, si este modelo ha venido para quedarse, proveedores y clientes necesitan acuerdos legales coherentes con esta filosofía.

CAPÍTULO II

CONTRATOS ÁGILES

Contratos adaptados para seguir los principios del *Manifiesto Ágil*.

Los contratos tradicionales presentan serias dificultades o limitaciones para las empresas que utilizan metodologías ágiles, que pueden resumirse como una falta de flexibilidad. Para algunos autores, como Opelt y Gloger (2013), los contratos ágiles son una evolución natural de los tradicionales, en una búsqueda de soluciones que ofrezcan garantías y contemplen posibles cambios a lo largo del proyecto. Todo ello de tal forma que siga siendo posible encajar estos contratos dentro del marco legal tradicional. Lo que surge es una síntesis que tiene en cuenta, por un lado, los principios del *Manifiesto Ágil* y, por otro, el iron triangle o «triángulo de hierro» de la gestión de proyectos clásica. Este triángulo es una representación típica de la interrelación entre el alcance, el tiempo y el coste de un proyecto[2].

Según Pries-Heje y Pries-Heje (2014):

- El alcance se refiere a los objetivos del proyecto descritos como funcionalidad, normalmente escritos como requisitos específicos.
- El tiempo es la planificación del proyecto.
- El coste es la cantidad de recursos invertidos.

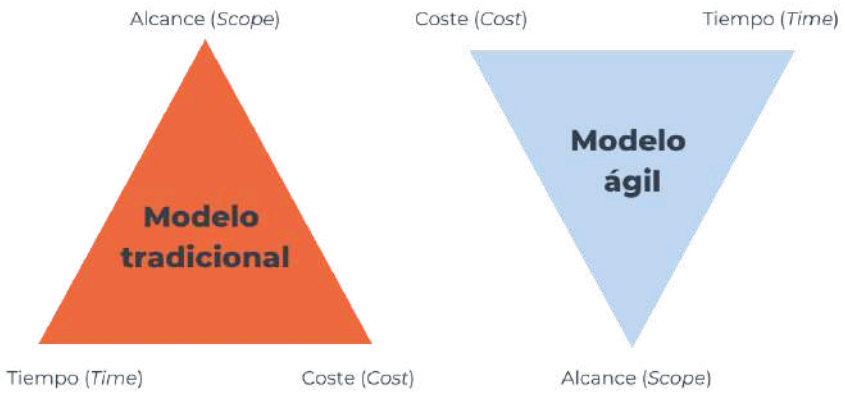


Fig. 2. *A la izquierda, el iron triangle. A la derecha, la alternativa ágil.*

El iron triangle es estático: el tiempo y los recursos disponibles delimitan el alcance. En un contrato ágil, en cambio, al menos uno de los elementos debe ser variable. De lo contrario, dejaría de ser ágil. El tiempo y el coste pueden variar en función del alcance, que suele ser imprevisible en este tipo de proyectos.

Son, por tanto, **contratos híbridos:** comparten características con un contrato tradicional y, a su vez, tratan de desmarcarse de él apostando por la **adaptabilidad.** No obstante, no existe el contrato ágil definitivo, como tampoco existe el contrato tradicional definitivo. No hay un modelo que copiar y pegar, porque cada redacción depende de tantos factores como los que afecten a cada proyecto. Independientemente del tipo de contrato, este debe adecuarse a las circunstancias. Con la diferencia de que los contratos

[2] El triángulo (o triple restricción) es un concepto clásico de la gestión de proyectos, anterior a la agilidad; Pries-Heje y Pries-Heje (2014) se citan aquí por sus definiciones, no por el origen del concepto.

ágiles hacen más por tener en cuenta y anticiparse a futuros cambios, por ejemplo mediante cláusulas.

En definitiva, un contrato ágil trata de cumplir con los objetivos de un contrato legal, pero busca otras formas de conseguirlo. Y, aunque no existe una fórmula mágica para su elaboración, es posible identificar principios comunes en su estructura.

Principios aplicados

> Para que un contrato tenga éxito, debe ser el resultado de relaciones basadas en la confianza, la colaboración y la transparencia.
>
> *Gerster y Dremel (2019, p. 4)*

Para que un contrato ágil tenga éxito, debería permitir al equipo cumplir con los doce principios del *Manifiesto Ágil*. Como ya se ha dicho, estos contratos tienen características híbridas y comparten elementos con los tradicionales. Una forma de entender los principios de un contrato ágil es como puntos clave de cualquier acuerdo empresarial a los que se les aplican principios ágiles.

A continuación se elaboran estos puntos, viendo cómo se gestionan en contratos ágiles y qué relación tiene cada adaptación con los principios del *Manifiesto*.

Incentivos por valor

En los contratos tradicionales, la remuneración por el trabajo realizado puede generar un conflicto de intereses entre cliente y proveedor. Por ejemplo: si se paga por tiempo invertido, el proveedor carece de incentivos para terminar antes de la fecha de entrega, mientras que el cliente buscará precisamente una entrega anticipada para ahorrar costes. Se genera así una tensión que afecta negativamente a la eficiencia del proyecto.

Ante estas situaciones, autores como Jeff Sutherland recomiendan ajustar los incentivos para alinear los intereses de cliente y proveedor (Sutherland, s. f.).

En la Figura 3 puede verse una tabla con los incentivos utilizados en el modelo tradicional, los resultados que pueden dar y alternativas posibles en contratos ágiles.

Modo tradicional	Resultados no deseados	Mejora de incentivos
Pagar por hora	Recompensa consumir más tiempo. Reduce la productividad de las mejoras.	Pagar por punto de historia entregado.
Pagar por líneas de código	No recompensa la calidad. Produce código menos limpio y conciso.	Pagar por valor comercial entregado.
Funcionalidad parcialmente completada	Demasiados incentivos para el trabajo en progreso.	Contar solo con funcionalidades publicables dentro del valor comercial entregado.
Fecha de entrega y alcance fijos	Anima a los equipos a trabajar menos para entregar a tiempo.	Elegir uno de los dos: fecha de entrega fija o alcance fijo.

Modo tradicional	Resultados no deseados	Mejora de incentivos
Proceso de aprobación complejo	Entregas lentas. Se limita la experimentación.	Tener un único punto de aprobación y un proceso suficiente para garantizar la calidad.

Fig. 3. *Traducción al español de la tabla de incentivos de Jeff Sutherland.*

Ofrecer incentivos o remuneración por puntos de historia o valor comercial, en lugar de por horas o líneas de código, son ejemplos claros de cómo estos cambios se alinean mejor con los principios del *Manifiesto Ágil*. Sobre todo con los siguientes:

> Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
>
> El software funcionando es la medida principal de progreso.

Los contratos ágiles, mediante incentivos como los mencionados, reflejan la filosofía ágil de tener como objetivo entregar valor al cliente.

Flexibilidad

La flexibilidad es fundamental en los proyectos ágiles, tanto que las metodologías de este tipo siempre incluyen mecanismos para un desarrollo más adaptable. Institucionalizar en el contrato estos aspectos, que pueden variar en función de la metodología empleada, ayuda a flexibilizar y conecta con varios principios del *Manifiesto*.

- **Establecer ciclos de feedback regulares.** Por ejemplo, especificar que el equipo trabajará en sprints, con las reuniones que esto suele llevar asociadas, y cada cuánto se realizarán reuniones de refinamiento del backlog. Si el proveedor se reúne a intervalos breves y regulares con el cliente para revisar nuevas funcionalidades, este puede dar feedback y sugerir cambios. En un modelo tradicional, los cambios se suelen solicitar al final, cuando llega la fecha de entrega, incrementando costes, tiempo y esfuerzo.
- **Identificar roles y procesos** para recibir feedback del cliente y refinar el backlog. Se definen las normas iniciales del acuerdo, donde se registra cómo van a manejarse los cambios y quién va a gestionarlos (Sutherland, s. f.).

Aunque estas estrategias conectan con varios principios del *Manifiesto Ágil*, quizá los más evidentes sean estos:

> Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
>
> Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Estos principios promueven un enfoque iterativo e incremental, que permite a los equipos entregar valor temprana y continuamente y hacer frente a los cambios sin desperdiciar recursos.

Colaboración directa

> Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

En los proyectos ágiles el cliente debe estar completamente implicado en el desarrollo, y esto debería quedar reflejado también en el contrato.

- **Compromiso total del cliente:** se debe involucrar al cliente, a ser posible especificando a un representante que colaborará de manera directa con el equipo durante el desarrollo (el rol product owner en scrum). Para lograrlo hay que dejar claros los objetivos, la participación de ambas partes y el riesgo que, aunque alto, debe ser compartido. Lo más importante es evitar las exigencias al cliente: hay que dar alternativas de cómo alcanzar los objetivos. Hay que sacar lo que más valor dé y, si hay trabas, tratar de llegar a una resolución. Es decir, se necesita una ayuda mutua entre cliente y proveedor.
- **Gestión de responsabilidades:** hay que saber manejar el modo de trabajo y la velocidad de cada uno. Quizás el cliente va demasiado lento y el proveedor necesita darle soluciones para que se adecue al ritmo de trabajo, ya que a veces el cliente puede tener más de un proyecto. Lo ideal es buscar alternativas de forma conjunta para que esto no afecte al ritmo de trabajo (Tortorella, 2012).

Comunicación frecuente

> El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

Una mala comunicación o falta de transparencia entorpece el avance del proyecto y daña la relación con el cliente. Para que la colaboración sea fluida, es necesario que

ambas partes se mantengan informadas en todo momento, y para ello es recomendable establecer cuanto antes unas rutinas de comunicación.

En el modelo tradicional, la comunicación con el cliente a la que estamos acostumbrados no es continua y no siempre está planificada. Suele dar lugar a un feedback tardío y a una pérdida del foco de atención que repercute en el alcance y los costes.

Con una metodología ágil lo que se busca es:

- **Visibilidad de los avances:** el cliente tiene que ver en qué se está trabajando para poder sugerir cambios durante el proceso. Si sabe lo que se hace, gana sensación de control y puede opinar. Para ello, en muchas ocasiones se incluyen en el contrato reuniones semanales con el cliente, o se comparten los avances a diario a través de alguna aplicación.
- **Pruebas de usuario conjuntas:** está relacionado con el punto anterior. En el modelo tradicional es habitual que el cliente no pruebe su propio producto. Pero, de nuevo, si el cliente no prueba, no puede saber si quiere cambios o mejoras. Se dará cuenta más adelante, lo que supone un aumento en los costes, el esfuerzo y el tiempo. Lo habitual es establecer sesiones dedicadas exclusivamente a probar el producto junto con el cliente e incluirlas en el contrato.
- **Manejo de las expectativas:** es muy importante tener claros los objetivos de cada uno y hasta dónde se puede llegar. Por ejemplo, el proveedor solo debería prometer lo que puede cumplir, y el cliente debe saberlo cuanto antes. Es buena idea aclarar los objetivos del proyecto lo antes posible y colaborar en la creación de un backlog detallado, siendo conscientes de que este cambiará.

Confianza

Los contratos ágiles tienen una desventaja respecto al modelo tradicional: es habitual que, al empezar a trabajar con un proveedor nuevo, exista cierta desconfianza inicial; y si además el cliente no ha trabajado antes con metodologías ágiles, esa desconfianza aumenta.

Una solución, similar a lo que haríamos en el modelo tradicional, puede ser dar referencias. Cuanta más información tenga el cliente, más seguro se sentirá. Se puede ofrecer información de otros proyectos realizados de forma ágil. Además, los departamentos de marketing y ventas deberían esforzarse por entender las necesidades y dudas de los clientes, para ofrecerles información adecuada sobre este tipo de contratos.

Sin embargo, a veces esto no es suficiente. Muchas empresas recurren a incluir ciertas cláusulas que aporten seguridad adicional a clientes reticentes. Por ejemplo, una cláusula sobre un posible acuerdo para un proyecto piloto de bajo riesgo o un sprint 0 (Tortorella, 2012).

Para mejorar la confianza entre las partes también se pueden incluir otras cláusulas que especifiquen las consecuencias de determinados comportamientos, tales como una falta de sinceridad sobre el estado de avance durante las iteraciones. De esta forma se previenen daños a la relación cliente-proveedor (Sutherland, s. f.). Esto beneficia a ambas partes, ya que la falta de confianza o una mala relación puede generar estrés al equipo, desmotivar y bajar la eficiencia (Tortorella, 2012).

En realidad, todos los aspectos del contrato, desde el foco en la entrega de valor al cliente hasta la comunicación frecuente y directa, están relacionados de alguna manera con reforzar la confianza entre las partes. Se trata de un intangible sin el cual es difícil que el proyecto tenga éxito, pues la agilidad se basa en equipos autoorganizados, capaces y motivados. Como dice el *Manifiesto*:

> Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

Ventajas y desventajas

Ventajas y desventajas de los contratos ágiles en general (Sutherland, s. f.):

Ventajas

Permiten un acuerdo legal consciente del cambio. Asumen que los cambios forman parte del desarrollo y son capaces de afrontarlos con solvencia, evitando sorpresas desagradables.

Aumentan la eficiencia sin perjudicar al proveedor. La comunicación frecuente y directa, acordada mediante un contrato adecuado, permite resolver problemas con más rapidez y generar valor real antes.

Los cambios acarrean costes menores, como consecuencia directa de las dos ventajas anteriores. Pueden realizarse sobre la marcha, antes de que los errores crezcan tanto que dañen el presupuesto del proyecto. Esto abarata costes y consigue un producto realmente valioso para el cliente.

Potencian la creatividad del proveedor: estos contratos brindan la flexibilidad necesaria para experimentar y proponer soluciones innovadoras, que pueden surgir durante el desarrollo y no solo durante la fase de planificación.

Requieren invertir menos tiempo en la documentación formal del proyecto. Los documentos formales de requisitos, propios de la gestión predictiva, son mucho más pesados que los backlogs de la gestión ágil.

Suelen producir productos de alta calidad. Al no existir un documento de alcance cerrado, sino un product backlog vivo, el proveedor puede sugerir cambios sin temor a desviar todo el alcance, y generar más valor en menos tiempo.

Desventajas

Desconfianza inicial. Si el cliente desconoce qué es la agilidad, la poca sensación de control ante la posibilidad de cambios puede generar desconfianza al principio.

Fuga hacia proveedores más afines. Esa desconfianza puede llevar al cliente a buscar otros proveedores más afines a su forma de trabajar. Un proveedor ágil puede caer entonces en la tentación de competir con ellos utilizando estrategias erróneas.

Problemas de compatibilidad. Si se comparte el desarrollo con otros proveedores no ágiles, pueden surgir problemas por la forma de trabajar.

Reparto desigual del riesgo. Si el cliente considera que el riesgo es alto, aunque compartido, puede intentar que el proveedor asuma una mayor parte.

Dependencia del compromiso del cliente. Si existe mala comunicación o el cliente no asume su rol en el desarrollo, aumentan los riesgos y se perjudica a ambas partes. El cliente debe estar plenamente comprometido y colaborar de manera directa con el proveedor.

CAPÍTULO III

ELABORAR UN CONTRATO ÁGIL

Pasos para definir las condiciones del marco contractual y elegir un modelo de contrato.

En el marco tradicional, todo aquello que no aparezca en un contrato se realiza mediante una solicitud de cambio formal, definida en el propio contrato (Chabes Floreano, 2020). Aunque el mecanismo exista, los cambios se perciben como algo a evitar. En el marco ágil, sin embargo, el cambio es siempre una posibilidad, por lo que nos encontraremos con contratos muy variados, cuyos elementos básicos pueden modificarse y adaptarse durante la negociación. Las negociaciones de los contratos ágiles pueden incluso prolongarse, en algunos casos y con ciertos modelos de contrato, y continuar durante el desarrollo.

En este capítulo se elaboran consideraciones generales a tener en cuenta para la selección de modelos y cláusulas concretos.

Definir el marco contractual

Antes de elaborar un contrato ágil hace falta definir las condiciones del marco del contrato, en lo que se conoce como *master service agreement*: un acuerdo base para el contrato (Fritz y Fischer, 2021, p. 4). Para ello se tienen en cuenta:

- **El tema principal del contrato:** la visión del proyecto.
- **La gobernanza del proyecto:** cómo se dirigirá y documentará el proyecto, incluyendo roles.
- **Precio/remuneración:** condiciones de cobro de los servicios.
- **Cooperación con el cliente:** diferenciación de tareas del cliente y del proveedor.
- **Niveles de servicio (*Service Level Matrix*):** detalles sobre los servicios que se compromete a prestar el proveedor, tales como estándares de calidad, tiempo de respuesta y penalizaciones en caso de que no se cumplan estas condiciones.

Este acuerdo base se elabora antes de la redacción del contrato para plantear riesgos y compartir responsabilidades.

Cuando el desarrollo se apoya en herramientas de inteligencia artificial, conviene añadir al marco una consideración más: el uso permitido de esas herramientas, la titularidad del código que generan y el tratamiento de los datos del cliente. Se aborda en detalle en «Contratos ágiles en la era de la IA» (→ capítulo 5).

Elegir un modelo de contrato

Hay empresas que, en vez de buscar principios específicos y cláusulas que encajen con sus proyectos, copian y pegan cláusulas que quizás no responden a sus necesidades. Pero no existe un contrato modelo que pueda aplicarse a cualquier situación. Cada proyecto tendrá un modelo de contrato que se adecúe a sus objetivos y necesidades, y que debería determinarse de antemano. Pero ¿qué modelo elegir?

Para decidir entre los distintos tipos de contratos es necesario tener en cuenta una serie de factores:

- Tamaño del proyecto.
- Dominio de las tecnologías necesarias para el desarrollo.
- Grado de madurez y experiencia con metodologías ágiles.
- Capacidad de autogestión y autosuficiencia del equipo de desarrollo del proveedor.
- Distribución de riesgos y recompensas entre las partes.
- Confianza entre las partes.
- Tipo de relación cliente-proveedor: competitiva, cooperativa, indiferente o dependiente (Hernández, 2019).

Hay una serie de elementos que deben abordarse en todos los contratos ágiles, pero que pueden presentar variaciones en función del modelo de contrato y de los factores anteriores. Según Arbogast et al. (2010), estos elementos son los que se definen en el resto de este capítulo.

Elementos comunes

El ciclo de entrega

Los modelos ágiles que gestionan el avance continuo mediante *timeboxing* entregan una parte terminada del producto al final de cada iteración. En los contratos tradicionales esto se suele conocer como hitos, que se fijan por fecha o por objetivos. No obstante, los ciclos de entrega tienen unas características distintivas:

- **El resultado de cada ciclo es implementable.** Cada entrega de iteración es una parte del producto terminada, probada y lista para usarse.
- **La duración es más pequeña**, y oscila normalmente entre una y cuatro semanas.
- **El alcance del proyecto es variable**, y antes de cada nuevo ciclo suele decidirse y estimarse lo que se desarrollará en él.

El contrato deberá definir aspectos como la duración media de los ciclos y las condiciones que deben darse para aprobar el resultado de la iteración (lo que en scrum se conoce como «incremento»), aunque esto es un elemento aparte a definir (los «criterios de aceptación»).

Alcance del proyecto

Los contratos ágiles no definen con exactitud su alcance (scope en inglés), aunque los hay más y menos específicos. Existen dos extremos: por un lado, contratos como los de coste fijo, en los que el alcance se identifica al principio, pero con mecanismos de cambio; por otro, contratos como los progresivos, en los que no es necesario definir ningún alcance más allá de la primera iteración.

Se desaconseja elaborar un contrato con un alcance y una visión demasiado vagos. Si este aspecto del contrato se vuelve tan vago que resulta inescrutable, puede llevar a conflictos en el futuro y a una falta de compromiso. Si el proyecto corre el riesgo de caer en este error, lo ideal es permitir que los asesores legales definan el alcance desde su propio entendimiento y perfilar esa definición con su asistencia. Se les puede hacer partícipes de la visión del proyecto mediante talleres y otras actividades.

Control del cambio

Los cambios en el alcance del proyecto son el área que **requiere mayor cuidado en la elaboración del contrato**, para hacer el cambio fácil y frecuente. Es aconsejable no exigir juntas de administración de cambios, solicitudes ni procesos de cambio.

Los conceptos de cambio en los contratos de proyectos ágiles se dividen en dos categorías: el cambio en las relaciones entre las partes y los cambios en el alcance del proyecto.

Cambios de relación entre las partes

Por ejemplo, si se produce una adquisición durante el desarrollo. Para abordar este tipo de cambios, el formato típico de los contratos tradicionales puede ser aplicable.

Cambios en el alcance

Para flexibilizar el alcance y evitar el rozamiento burocrático, se aconseja no exigir reuniones o procesos formales para solicitar, documentar y aprobar cambios. En un proyecto ágil existen otros sistemas para ello, y normalmente forman parte de la metodología que se emplea. El product owner y el equipo suelen estar empoderados para tomar estas decisiones, en la medida en que el cliente lo considere oportuno, y los cambios se deciden durante las reuniones típicas de cada modelo, colaborando de forma directa.

Cabe observar que el nivel de flexibilidad puede ir desde muy alto y sin penalizaciones, en contratos progresivos, hasta medio, con ganancias y dolores compartidos, en contratos de precio fijo (Arbogast et al., 2010, p. 520).

Rescisión

Este elemento está relacionado con la filosofía ágil que dice que los cambios son bienvenidos. Aunque la rescisión anticipada tiene connotaciones negativas, en un proyecto ágil puede ser un evento deseable: puede significar que el éxito se ha logrado antes de lo esperado. Y en caso de que la colaboración no resulte, es preferible para ambas partes una separación fácil.

El modelo de rescisión ideal en un contrato ágil es aquel que permite que el cliente se detenga al final de cualquier iteración, sin penalización. No obstante, sin

penalizaciones, el proveedor puede salir perjudicado. Para esto existen variaciones de las cláusulas de rescisión.

Criterios de aceptación

La aceptación del incremento generado al final de cada ciclo se basa en un acuerdo, un conjunto de pruebas previamente acordadas. En el caso de scrum es normal referirse a estas como «definición de "hecho"».

A veces hay cambios de opinión o rechazos de entregas tras varias iteraciones, y esto puede generar conflictos. Por eso es importante definir unos términos claros de finalización, aceptación y corrección.

Por otra parte, las entregas iterativas y la comunicación directa con el cliente simplifican el proceso de aceptación: con el tiempo, la aprobación y los acuerdos de cada entregable se acumulan. No se desarrolla a ciegas, sino colaborando y confirmando con regularidad que se avanza en la dirección correcta.

Entregables

Los contratos tradicionales a menudo incluyen una lista detallada de lo que se debe entregar. Estos detalles a veces se incorporan en un apéndice del «plan de calidad». Este formato puede resultar demasiado específico y rígido para proyectos ágiles, y traer consecuencias indeseadas, tales como:

- Centrarse en negociar y cumplir con el plan de calidad en lugar de cooperar para crear funcionalidades útiles.
- Reforzar la planificación predictiva de mando y control, en lugar de la capacidad colectiva de aprendizaje y respuesta al cambio.
- Reforzar la creencia de que se puede definir un proyecto en detalle y generar expectativas poco realistas de entregas predecibles.

Por ello, en los contratos ágiles existen otras maneras de definir los entregables, para que respondan a la filosofía de entrega de valor frecuente e incremental.

En este punto hay que considerar también la documentación técnica que puede ir asociada a cada entregable. Esta documentación, que suele especificarse de forma detallada en los contratos tradicionales, se solicita como respaldo para el futuro mantenimiento del proyecto y como ayuda para nuevas incorporaciones. Sin embargo, el mantenimiento suele subcontratarse a las mismas personas que desarrollan el proyecto, por lo que no es necesaria una documentación exhaustiva. Exigirla como entrega anticipada puede suponer una pérdida de tiempo. Una opción, en caso de que sea una necesidad para el cliente, es realizar un taller conjunto de documentación ágil.

Periodicidad de pagos

El sistema más popular y extendido es realizar un pago por iteración, una vez que el entregable es aceptado. Lo más sencillo es acordar el pago del 100 % del precio pactado por iteración al final de cada ciclo.

Garantías

Las preocupaciones sobre las garantías se atenúan de forma incremental conforme el proyecto avanza. El perfil de riesgo asociado a la garantía final es más bajo por la confianza y la aceptación de los entregables.

Al igual que con las limitaciones de responsabilidad, la garantía debe vincularse a cada entrega de trabajo incremental (al final de cada iteración), aunque exista una garantía general para el producto final.

Se suelen utilizar técnicas como la ya referida «definición de "hecho"» y la medición continua del rendimiento para establecer y medir garantías. La primera se emplea para asignar criterios de aceptación generales que permitan dar por terminadas las historias de usuario. En cuanto a la medición del rendimiento, se pueden establecer indicadores clave (KPI) con los que medir el desempeño del proveedor y el progreso del proyecto.

Limitaciones de responsabilidad

Las limitaciones de responsabilidad son cláusulas que establecen un límite a la cantidad de responsabilidad que una parte asumirá en caso de incumplimiento del contrato. En contratos ágiles pueden ser menos específicas que en los tradicionales.

La negociación de estas cláusulas suele ser difícil, pero establecerlas ayuda a superar dificultades.

En un contrato tradicional, estas cláusulas suelen proteger al proveedor: limitan la cantidad de daños y perjuicios que es responsable de pagar en caso de incumplimiento, excluyen ciertos tipos de daños o requieren que el cliente tome ciertas medidas para mitigar los daños antes de presentar una reclamación. Por ejemplo, se puede establecer que el cliente no pueda recuperar más de lo que pagó por los bienes o servicios.

En un contrato ágil, los riesgos por imprevistos se atenúan, ya que el desarrollo es progresivo y se entrega un incremento útil al final de cada ciclo. Estos incrementos, al usarse, permiten identificar los problemas y hacer cambios antes de que su impacto resulte desbordante. Esto reduce costes, así como los daños a la imagen del cliente y a la relación entre las partes, y debería tenerse en cuenta al redactar las cláusulas de limitaciones de responsabilidad. Cabe reflejar, además, que el cliente debe estar implicado en la identificación y corrección de los fallos conforme surjan.

Fijación de precios

Varía en función del tipo de contrato. El precio puede fijarse por unidad de tiempo (por iteración o ciclo), por unidad de trabajo (normalmente conocido como «punto de historia» o «punto de función»), por uso de la solución implementada...

Los modelos de pago por uso (pay-per-use) pueden tener sentido en proyectos de desarrollo de software, en los que cada «uso» de un sistema personalizado para el cliente se rastrea automáticamente por el proveedor. Se factura al cliente de forma regular en función de la frecuencia de uso. Este enfoque alinea los intereses del cliente y del proveedor, ya que ambas partes ganan si el sistema se usa cada vez más.

CAPÍTULO IV

MODELOS Y CLÁUSULAS

Inventario de modelos y cláusulas para contratos ágiles, en orden alfabético.

Modelos de contrato

Acuerdo de sprint

Se suele usar para proyectos pequeños, que se completan con un número reducido de sprints. No es realmente un contrato comercial, sino un acuerdo entre el product owner y el equipo. Puede incluirse en el contrato comercial, que puede ser de tipo tiempo y materiales.

Características

- **El alcance puede variar, aunque se mide tras cada sprint.** El equipo se compromete a entregar los requerimientos acordados con un estándar de calidad definido al final del sprint. El product owner acepta no cambiar sus instrucciones antes del final del sprint.
- **No hay posibilidad de introducir cambios** (ya que serían dentro de un sprint en curso).
- **Es adecuado para trabajar en scrum** en microdesarrollos o pequeños evolutivos, aunque no es un uso muy habitual.

Para este tipo de contratos es habitual que tanto proveedor como cliente sepan de agilidad y lleven tiempo trabajando de este modo. ¿En qué consiste?

1. Se asigna un equipo a un proyecto.
2. Si no tiene velocidad conocida, se evalúan los tres primeros sprints para obtenerla.
3. Se establece un precio por equipo y sprint, o por punto de historia implantado, asumiendo la cantidad de puntos que pueden completar.
4. El contrato finaliza cuando se alcanza el valor deseado.

Bob Martin

Este modelo de contrato es obra de Robert Cecil Martin, ingeniero y desarrollador de varios principios de diseño de software. Es también uno de los coautores del *Manifiesto Ágil*.

Su propuesta combina el contrato de tiempo y materiales, el precio por hora y el precio por punto de historia. Según Martin (s. f.):

- Si el esfuerzo real es igual a la estimación original, el pago del cliente equivale a un esquema simple de contrato por tiempo y materiales.
- Si el esfuerzo real varía, el pago del cliente varía de forma menos severa que en un contrato tradicional por tiempo y materiales.
- El cliente y el proveedor comparten el pain/gain si el proyecto requiere más o menos esfuerzo que la estimación original.

Costes reembolsables

Este tipo de contratos se utiliza cuando el alcance es incierto y los costes no se pueden estimar lo suficiente para hacer un contrato de precio fijo.

En el contrato se estima por anticipado una determinada suma, un coste que ambas partes consideran probable. Durante el desarrollo del proyecto, el proveedor debe responsabilizarse de llevar un registro de los costes reales. Si son menores que la suma acordada, el cliente sale beneficiado. Si se incurre en costes adicionales, el cliente debe asumirlos. Es un tipo de acuerdo con más riesgo para el cliente.

Otras versiones

Costes más honorarios con incentivos (*Cost plus incentive fee*)

El proveedor recibe un reembolso por los costes acordados para realizar el proyecto y un pago adicional por sus honorarios. Estos honorarios se basan en incentivos preestablecidos: si el proveedor cumple con los objetivos, recibe una recompensa adicional en forma de honorarios. En caso de que el coste final del proyecto sea diferente al estimado al principio, cliente y proveedor comparten las desviaciones según una fórmula previamente negociada. Por ejemplo, se puede acordar que el 80 % de las desviaciones corresponda al cliente y el 20 % al proveedor.

Costes más honorarios fijos (*Cost plus fixed fee*)

El proveedor recibe un reembolso por los costes acordados para realizar el trabajo del contrato y un pago adicional por sus honorarios, que se calculan como un porcentaje de los costes del proyecto estimados al principio.

Costes más honorarios por cumplimiento de objetivos (*Cost plus award fee*)

Al proveedor se le reembolsan todos los costes, pero la mayor parte de los honorarios se obtiene en función de la satisfacción de ciertos criterios subjetivos generales de desempeño, definidos e incorporados dentro del contrato (Chabes Floreano, 2020).

Híbridos

Los contratos híbridos en este contexto parecen redundantes, pero es una categoría que cabe aplicar en ciertos casos y que se suele mencionar. Según Chabes Floreano (2020), son contratos a medio camino entre lo tradicional y la metodología ágil. Combinan, por ejemplo, varios modelos de contrato y varios tipos de cláusulas. Suelen darse en proyectos grandes y pueden conseguirse aplicando cláusulas ágiles a contratos que, por lo demás, tienen un formato más o menos tradicional. Por ejemplo, un contrato por tiempo y materiales con cláusulas.

Por capacidad (por iteraciones)

Este modelo se ajusta a proyectos complejos, donde los requisitos cambian constantemente entre sprints y el proveedor tiene que trabajar de forma independiente: proyectos con un alto nivel de desarrollo.

Características

Según Hernández (2019):

- El alcance es abierto.
- El product owner puede introducir todos los cambios necesarios.
- El coste total del proyecto = velocidad del equipo × número de sprints previstos hasta la fecha de release × valor del punto de historia.

Hay que tener en cuenta:

- Las historias de usuario de referencia.
- La definición de «hecho» acordada.
- El product owner debe ser efectivo al redactar los criterios de aceptación y priorizar las historias que realmente aporten valor.

Intereses:

- El proveedor debe entregar los puntos previstos para poder continuar con más iteraciones.
- El cliente tiene previsibilidad de costes y de la cantidad de funcionalidades que obtendrá a lo largo del tiempo.

Ventajas y desventajas

Entre las **ventajas** destacan el alto nivel de agilidad de este modelo y el amplio margen de maniobra y libertad creativa que brinda al proveedor.

Como **desventaja**, un coste total incierto y unas negociaciones que pueden requerir más tiempo que otros modelos.

Precio fijo

Son los más habituales en proyectos de media y larga duración con gran complejidad. Estos contratos definen claramente los requisitos del trabajo y se recomiendan solo cuando las partes tienen suficiente competencia y experiencia en el mercado para acordar un precio provisional realista. Aunque en este sentido son similares a los contratos tradicionales, hay diferencias importantes: el precio se estima con técnicas ágiles y no es definitivo hasta pasado un periodo de tiempo.

Estimación

Para poder acordar el precio antes del desarrollo del proyecto, se divide el backlog del proyecto —al menos el primer ciclo— en historias de usuario. El proveedor utiliza proyectos de referencia para estimar el tiempo y el esfuerzo necesarios para completar cada historia de usuario en puntos (story points). Finalmente, las partes deben acordar un precio fijo por un determinado número de story points. El contrato debería dejar reflejado el compromiso de realizar reuniones de revisión conjuntas después de cada sprint, en las que discutir si el alcance requiere ajustes.

Punto de salida

El contrato debe reflejar un periodo de tiempo, desde el inicio de la colaboración hasta lo que se suele llamar el «punto de salida» o «no retorno», durante el cual el cliente puede echarse atrás sin penalizaciones y el precio del proyecto se mantiene alterable. Es un tiempo para comprobar si las estimaciones realizadas parecen acertadas o no. A partir del punto de «no retorno», el precio se vuelve fijo y la colaboración se consolida.

Ventajas y desventajas

Además de las **ventajas** generales de un contrato ágil, este modelo facilita la planificación y la asignación de recursos. Puede dar más confianza a un cliente que no esté familiarizado con la agilidad, sin dejar de ser un contrato flexible y adaptado a metodologías ágiles.

Como **desventaja**, sus negociaciones también pueden llevar tiempo, aunque permite empezar a desarrollar mientras se afina la estimación del precio.

Otras versiones

Según Arbogast (2010) estas son algunas versiones:

Precio fijo más honorarios con incentivos (*Fixed Price Incentive Fee*)

Permite que haya desviaciones en el desempeño, con incentivos financieros ligados al cumplimiento de las métricas acordadas. Por ejemplo, en un contrato de servicios de consultoría, el proveedor puede establecer un precio fijo más un honorario adicional si alcanza una tasa de retorno específica para el cliente. El precio fijo garantiza que ambas partes tengan una idea clara del coste total del proyecto, mientras que los incentivos aportan una motivación adicional para lograr el resultado deseado. Es importante que las metas y objetivos se establezcan claramente en el contrato para evitar malentendidos.

Precio fijo por iteración (por unidad de tiempo)

Es diferente porque la estimación del precio es más simple y previsible. Es habitual entre los subcontratistas ágiles. Hay dos formas de plantearlo:

- **Los requisitos están definidos y acordados antes de la iteración.** El proveedor tiene que aclarar el trabajo y confiar en las estimaciones. Un problema para los clientes puede ser que el proveedor añada una tarifa de contingencia, debido al riesgo asociado con la variabilidad en el trabajo de investigación y desarrollo.
- **Los requisitos son altamente flexibles o no hay requisitos predefinidos.** La cuestión clave es la confianza del cliente en el proveedor. Tiene que haber transparencia, la entrega debe ser frecuente y debe haber facilidad para poner fin al proyecto.

Precio fijo por unidad de trabajo (UoW)

A diferencia del desarrollo tradicional, una UoW ágil refleja el séptimo principio ágil: el software funcionando es la medida principal del progreso. Estos modelos tienen varios nombres: «precio por punto de historia», «precio por punto de función», «precio por punto de característica», etc. El «punto» siempre está relacionado con una estimación de tamaño o esfuerzo y, por lo tanto, con el coste. No obstante, hay quien considera que un story point no es una medida exacta de valor. Por lo general, pueden usarse algunos esquemas para determinar el precio fijo por punto: por ejemplo, calcular el promedio basado en varios proyectos anteriores o en una cantidad personalizada. En este último caso, el cliente paga el valor en puntos promedio del proveedor durante algunas iteraciones; luego, proveedor y cliente acuerdan un precio fijo personalizado por punto, basado en ese coste más algún margen de ganancia. Lo importante en este tipo de modelos es que cliente y proveedor tengan un marco claro y común para definir un punto[3].

Precio y alcance fijos

Este modelo suele aparecer en las conversaciones sobre contratos ágiles, pero no se recomienda.

Se define por un alcance fijo que se divide en entregas, trabajando en ciclos. El pago se suele realizar por iteración, con una suma global al finalizar (si no se ha agotado el presupuesto antes). El pago por iteración es un porcentaje fijo del precio total, basado en una estimación del número total de iteraciones o en su duración.

Contempla la posibilidad de cambios, pero con fricciones añadidas. El alcance fijo significa que estos no son deseables y, por tanto, suelen existir procesos formales para solicitarlos y aprobarlos. Son contratos que tratan de solaparse con las idiosincrasias de un entorno ágil —como las entregas en ciclos de corta duración y los requisitos en historias de usuario (Stevens, 2019)—, pero que se contradicen y llevan a problemas similares a los de los contratos tradicionales.

[3] La irrupción de la IA generativa tensiona el supuesto de estabilidad entre esfuerzo y valor sobre el que se apoyan los modelos «por punto de historia»; se analiza en «Contratos ágiles en la era de la IA» (→ capítulo 5).

Ventajas y desventajas

Apenas se aprecian **ventajas**. A veces se usan por dar mayor seguridad y confianza al cliente, pero se trata de una falsa seguridad. Dado que el alcance es fijo, la gestión de cambios está diseñada para prevenir, limitar y dificultar cualquier modificación del backlog. Pero los cambios en proyectos ágiles no son predecibles ni evitables, y las fricciones que generen afectarán al avance y al presupuesto del proyecto.

En estos modelos aparece el triángulo de hierro: alcance, precio y duración son fijos. El cliente fija el alcance, y el proveedor fija el coste y el tiempo. La mayor parte del riesgo recae sobre el proveedor: debe analizar los requerimientos, el esfuerzo necesario para cumplir con las expectativas del cliente y hacer una oferta mejor que la de la competencia. Pero si sus estimaciones fallan, perderá dinero. Si se han infravalorado los riesgos y esfuerzos, o se ha accedido a un coste demasiado bajo, el proveedor pierde, y en realidad el cliente también, pues el proyecto se resentirá.

Muchas de estas limitaciones giran en torno a fechas de entrega restrictivas, que en este tipo de contratos suelen tener mucha importancia. Cuando es imposible cumplir con los plazos prometidos, se suelen usar estrategias erróneas que perjudican tanto al cliente como al proveedor (Hernández, 2019). Por ejemplo:

- **Sobreesfuerzos:** las horas extra no tienen retribución, por lo que hay menos motivación y peor rendimiento.
- **Aumentar la capacidad:** se realizan nuevas contrataciones, lo que implica invertir tiempo y esfuerzo en explicaciones. Esto aumenta la complejidad en comunicación y coordinación, y la posibilidad de conflicto. Se generan concesiones mínimas por parte del cliente, en un círculo infinito de prórrogas por parte del proveedor.
- **Prescindir de eventos o modificarlos.** La reunión diaria puede convertirse en una discusión constante (especialmente cuando los retrasos son evidentes) centrada en buscar culpables. En casos extremos, se abandona completamente scrum para intentar cumplir la fecha de cualquier manera.

La relación proveedor-cliente puede resentirse por las falsas expectativas y los imprevistos, y puede ir desde competitiva hasta indiferente. Cuando el cliente pide más de lo acordado, el proveedor suele ceder o prometer de más para mantenerlo. A todo ello hay que sumar los intereses contradictorios que surgen de todas las partes implicadas (Hernández, 2019):

- **Intereses cruzados.** El cliente suele ejercer demasiado control. Sus distintos sectores interesados pueden querer realizar cambios según su conveniencia, por lo que el proveedor deberá ceder a la presión para mantenerlo.
- **Los objetivos interdepartamentales son obstáculos.** Por ejemplo, si el área comercial que gestiona la oferta tiene objetivos de facturación poco realistas: cerrar la operación de cualquier forma, aunque luego no sea factible por tiempo o presupuesto.

- **Contra el usuario.** El proveedor intentará entregar un producto que cumpla los requisitos en la fecha pactada, aunque muchas funcionalidades sean defectuosas o no estén implantadas, ya que luego tendrá un contrato de mantenimiento durante el cual podrá, sin presión, solventar estas cuestiones. Es el usuario final quien sufre por la baja calidad.

Venture capital financing model

Este modelo financiero se suele emplear en inversiones de capital de riesgo para evaluar, en una etapa temprana, la viabilidad de una inversión en una empresa. Utiliza proyecciones financieras a largo plazo (ingresos, gastos, flujos de efectivo, valoración...), así como riesgos e incertidumbres, para determinar el potencial de retorno. Después de cada periodo de trabajo (entre cada financiación) se revisan los términos del contrato.

Cláusulas

En un contrato ágil, las cláusulas están diseñadas para ser flexibles y permitir a las partes trabajar juntas para lograr el objetivo del contrato. Aunque se puede argumentar que usarlas no garantiza ajustarse al ideal de agilidad, un uso consciente puede dar como resultado contratos híbridos adecuados según el proyecto.

Las cláusulas pueden definir ciertas obligaciones y responsabilidades, como:

- El objetivo del contrato.
- Los bienes o servicios que se proporcionarán, y sus límites.
- Cómo se llevarán a cabo las comunicaciones entre las partes, incluyendo puntos de contacto y protocolos.
- Cómo se gestionarán los cambios en el alcance a lo largo del proyecto.
- Los términos del pago, incluyendo precio, plazos y condiciones.
- Las responsabilidades y penalizaciones de cada parte en caso de incumplimiento.

Bonus/penalty clauses

Las cláusulas de bonificación y penalización consisten en recompensar al proveedor si finaliza antes de la fecha acordada y penalizarlo si hay retrasos.

Características

Según Stevens (2019):

- La cantidad de dinero de la bonificación y la penalización suele depender de la desviación respecto a la fecha de entrega.
- Esta cláusula hace los cambios más difíciles de aceptar, porque siempre afectan al tiempo de entrega.
- El riesgo recae por completo sobre el proveedor. Cuanto menos tiempo se tarde, más dinero pierde el cliente, que puede preferir que haya retrasos; es necesario que exista confianza y buena comunicación entre las partes.
- La relación cliente-proveedor puede ser cooperativa, pero puede degenerar en indiferente si la fecha acordada no es muy importante para el cliente.

Changes for Free

En español, «cambios gratis». Es una cláusula creada por Jeff Sutherland que se combina con la siguiente (*Money for Nothing*). Establece un acuerdo mutuo a la hora de estimar requisitos.

Reglas

- El product owner prioriza la pila de producto antes de cada sprint.
- Los cambios de prioridad son gratuitos siempre que no varíe la cantidad total de trabajo.

- Pueden añadirse requisitos nuevos sin coste en los límites del sprint, si su esfuerzo es equiparable al de otros ya estimados.
- A cambio, el cliente se implica: sigue el proyecto de cerca y colabora con el product owner para mantener un product backlog de calidad.

Money for Nothing

En español, «dinero por nada». Es otra cláusula creada por Jeff Sutherland. Establece un acuerdo mutuo a la hora de estimar riesgos.

Características

- El cliente determina en qué momento desarrollar un requisito (o varios) tiene más coste que el valor que aportará después.
- El cliente puede rescindir el contrato en cualquier momento; si lo hace, abona al proveedor un porcentaje (habitualmente en torno al 20 %) del valor de las historias que quedaban pendientes de desarrollar.
- El proveedor asume el riesgo de entregar tarde los trabajos estimados y acordados mutuamente.

Money for Nothing, Changes for Free (combinación)

Son dos cláusulas que incentivan al cliente y al proveedor a enfocar la relación desde el punto de vista del valor. El modo de trabajar es quizá como en un contrato de tiempo y materiales con un objetivo de coste, normalmente con la idea de que el proyecto no debería agotar el presupuesto. La diferencia es que hay un incentivo para el cliente (paga un porcentaje) y para el proveedor (ahorra costes y obtiene dinero gratis) por acabar antes.

- Una vez entregada cierta cantidad de proyecto, si se ha alcanzado un valor suficiente y no es necesario más desarrollo, **se cancela el proyecto.**
- **Si el proyecto se cancela antes del plazo estimado, se aplica *Money for Nothing*:** el cliente paga al proveedor un porcentaje del compromiso remanente, que suele coincidir con el beneficio neto del proveedor por el tiempo de servicio no prestado.
- ***Changes for Free:*** el product owner puede introducir cuantos cambios considere, siempre dentro de lo que prescribe scrum. Si, por ejemplo, introduce cambios durante un sprint o influye en las estimaciones, no es posible continuar y se convierte en un contrato por tiempo y materiales.
- **El alcance puede cambiar.** Está planificado, pero puede tener cambios dentro de requerimientos o historias de usuario del mismo tamaño. Hacer cambios más grandes o añadir nuevos requerimientos conlleva dinero extra.
- **El riesgo es compartido.** A ambas partes les interesa completar el proyecto en fecha o antes: el cliente tiene menos pérdidas y el proveedor, más margen.

- **Si se supera el presupuesto,** se pueden aplicar las reglas del contrato de precio fijo o de un contrato de costes máximos. Para conseguir una relación cooperativa, parece más recomendable aplicar el precio fijo.

Not to exceed with fixed fee

Se trata de establecer un presupuesto mínimo (*fixed fee*) y uno máximo (*not to exceed*). En estos casos se suele recomendar llegar a acuerdos con el cliente y, en caso necesario, incluir en cláusulas qué sucede si se supera el presupuesto máximo.

Sprint de confianza

Se suele añadir en contratos con clientes que necesitan sentirse más seguros antes de entrar en una relación contractual. El sprint de confianza da facilidades para rescindir el contrato sin penalizaciones tras un número determinado de sprints. Tiene similitudes con el acuerdo de sprint, con la diferencia de que puede formar parte de cualquier otro tipo de contrato ágil. Es común para empresas que nunca han trabajado con agilidad y para proveedores que quieren mostrar al cliente las ventajas del enfoque ágil.

CAPÍTULO V

CONSEJOS

Recomendaciones extraídas de fuentes bibliográficas para la elaboración y negociación de contratos ágiles.

Elaboración de contratos

La sección se divide en cuatro puntos:

- **Encontrar al mejor socio/proveedor:** consejos dedicados al cliente sobre la elección de socios o proveedores.
- **Sobre asesoría legal:** consejos sobre la importancia de recurrir a asesores legales y abogados, no solo para la redacción del contrato.
- **Sobre contratos y cláusulas:** consejos sobre las cláusulas y tipos de contratos adecuados para cada tipo de proyecto.
- **Elaborar, presentar y negociar un contrato:** consejos sobre la elaboración, la venta y la negociación de un contrato ágil.

Encontrar al mejor socio / proveedor

Para el cliente, uno de los puntos clave antes de realizar el contrato es identificar un buen socio. El éxito del proyecto depende del compromiso de ambas partes y de que la relación sea de cooperación. Los requisitos contractuales deberían seleccionarse para garantizar un equilibrio óptimo de costes, tiempo y calidad para ambas partes.

En ocasiones se puede encargar a otro proveedor externo, o a un recurso interno con conocimientos suficientes, que desempeñe un papel operativo en la implementación del proyecto. Según Fritz y Fischer (2021), este servicio externo puede ejercer como scrum master o product owner.

Esta decisión no debe tomarse a la ligera. El product owner es un puesto estratégico clave para el éxito del proyecto, y en general se recomienda que sea alguien de la empresa del cliente, que conozca al milímetro el negocio y la visión. No obstante, si se encuentra a la persona correcta, un buen servicio de consultoría puede aportar valor (Léonard, 2019). Puede ser buena idea si la organización no tiene suficiente experiencia o no dispone de una persona que pueda dedicarse al 100 % a la función de product owner.

Sobre asesoría legal

Para Arbogast et al. (2010) es importante que el asesor o abogado que vaya a redactar un contrato ágil tenga información sobre agilidad o haya investigado sobre ello. Un abogado con formación tradicional tenderá a desconfiar de modelos y cláusulas que desconozca o no considere lo bastante probados, con el fin de proteger al cliente.

Los asesores legales sin experiencia previa en proyectos ágiles pueden partir de suposiciones equivocadas. Por ejemplo, al no estar familiarizados con las incertidumbres y variables propias de un proyecto de software, pueden enfocarlo como un proyecto escalonado, como uno de construcción. Estas suposiciones no casan con el desarrollo ágil (Arbogast et al., 2010, p. 502) y es muy probable que se expresen en el lenguaje del contrato, perjudicando a las partes.

Además, la visión tradicional de la asesoría legal también influye en la definición de «éxito» del proyecto. No se suelen tener en cuenta factores como que los usuarios finales estén satisfechos con el producto, o que este pueda ser un éxito aunque sus requisitos iniciales hayan cambiado (Arbogast et al., 2010).

Por otro lado, entre abogados suele decirse que, si ambas partes están algo descontentas, el contrato es un éxito: ambas han tenido que ceder para llegar a acuerdos. Pero esta mentalidad puede contradecir lo que se busca en agilidad, que son situaciones *win-win*.

Sobre contratos y cláusulas

Fritz y Fischer (2021) recomiendan distintos contratos en función de la complejidad y la duración del proyecto:

- **Proyectos sencillos y de corta duración**, centrados en costes y con una visión clara. Los autores recomiendan lo que denominan «contrato de trabajo para el proyecto integral» (*work contract for the holistic project*), aludiendo a costes transparentes y predecibles. Sin embargo, este modelo tiene un bajo grado de flexibilidad. En su lugar, podría ser más recomendable el **acuerdo de sprint**; **o por iteraciones**, si no es un proyecto tan reducido.
- **Proyectos complejos y de larga duración.** Recomiendan un **contrato por iteraciones**, que permite realizar ajustes sobre la marcha en lugar de intentar planificar lo impredecible desde el comienzo. Hay que tener en cuenta que en este caso los costes son inciertos, ya que habrá ajustes de una iteración a otra. También pueden utilizarse los contratos ágiles de precio fijo, con la desventaja de que requieren más tiempo de planificación y un periodo de prueba.
- **Proyectos ejecutados y controlados por el cliente.** Aquí los autores hablan de los contratos de servicio. Dado que el proveedor no está obligado por el éxito del proyecto, los recursos internos deben tener suficiente experiencia en la gestión de proyectos ágiles para garantizar un cierto estándar de calidad y rentabilidad.

En cuanto a cláusulas, Hernández (2019) considera que utilizar las cláusulas *Money for Nothing y Changes for Free* en proyectos o microdesarrollos de 200 horas es poco productivo.

Elaborar, presentar y negociar

Antes siquiera de plantearnos elaborar un contrato, debemos instrumentar un marco de contratos ágiles, es decir, preparar el terreno. Hernández (2019) recomienda lo siguiente:

- **Indagar en el mercado** para saber qué proveedores ofrecen enfoques ágiles.
- **Formar al product owner.**
- **Hacer pruebas:** hay que aprender a ser ágil, y para ello es necesario practicar. Por ejemplo:

    - **Aprovechar pequeños evolutivos o microdesarrollos** de menos de un mes para implantar un sprint único que sirva de prueba para aprender sobre agilidad.
    - **Aprovechar pequeños proyectos (2 o 3 meses de estimación)** usando contratos de tiempo y materiales para implantar un equipo scrum y conocer su velocidad.

Supongamos que ya tenemos cierta experiencia y hemos hecho proyectos con una metodología ágil, y que ahora queremos aplicarla a un proyecto de mayor magnitud. Para Hernández (2019) debemos tener claros varios puntos:

- **Cada contrato responde a una necesidad:** usar la matriz de Stacey puede ser útil, porque propone diferentes formas de desarrollo según el grado de certeza en los requerimientos y el dominio de la tecnología implicada.
- **El alcance no debería ser fijo** si la dimensión y la complejidad del proyecto son determinantes.
- **Implantación de scrum disfuncional:** si la implantación de scrum es disfuncional, o hemos comenzado a aplicarlo hace apenas una semana, un contrato ágil no va a solucionar todos los problemas.

Una vez claros estos puntos, hay que escoger el tipo de contrato y las cláusulas. Existen distintas opiniones sobre la mejor forma de redactar un contrato o qué elementos evitar e incluir. No obstante, una opinión común es la que comparte Tortorella (2012) en su vídeo Kleer — *Contratos ágiles en la vida diaria*: **el primer contrato ágil debería ser lo más sencillo posible** (precio por sprint, por persona, por hora, etc.), aunque debe incluir situaciones que podrían darse. No va a ser perfecto y tendrá que modificarse. Se pueden añadir tantas cláusulas como sean necesarias, adecuadas para cada proyecto y cliente.

En caso de conocer el enfoque ágil y trabajar con un cliente que también lo conoce, Hernández (2019) propone hacer acuerdos marco con un precio por punto de historia. Si, además, ya tenemos un equipo scrum y su velocidad es conocida, puede asignarse a un proyecto más grande utilizando un contrato por capacidad o de tiempo y materiales con techo de coste y alcance variable.

Para Arbogast et al. (2010), lo recomendable al elaborar un contrato es:

- **Evitar incentivos y penalizaciones.** Se asume que los incentivos y bonus son beneficiosos, pero hay quien considera que reducen la transparencia y la calidad y aumentan la competitividad entre cliente y proveedor. Con las penalizaciones ocurre lo mismo. Una alternativa pueden ser los modelos de *share the pain/gain*, que fomentan la colaboración y la mejora conjunta.
- **Evitar el *Quality Management Plan* y la *Deliverables List*.** Los contratos tradicionales suelen implicar niveles bajos de confianza y transparencia. Una de las soluciones tradicionales es exigir un «plan de gestión de calidad» convencional o una «lista de entregables» que defina una larga lista de verificación de documentación, en lugar de centrarse en ofrecer valor real. En scrum existe una

definición de «hecho», realizada por el equipo del proveedor y el cliente, que se va definiendo con cada iteración: se basa en prácticas y artefactos, se redefine al principio de cada iteración y evoluciona conforme a lo que cada parte considere que aporta valor.

- **Colaborar antes y a menudo con abogados.** Es importante recurrir a asesores legales antes de redactar cualquier contrato y durante el desarrollo del proyecto.

Para elaborar un contrato también es relevante saber cómo proponérselo al cliente y negociar. Para Tortorella (2012), la propuesta del contrato y del proyecto depende del tipo de cliente. Además, considera fundamentales varios de los principios de los que hablábamos en el capítulo 3: transparencia, confianza y comunicación.

Para dar confianza al cliente se puede ofrecer un proyecto piloto o un primer sprint que no supongan un gasto si son rechazados. De hecho, puede incluirse como cláusula, y el proyecto puede no tener mucho riesgo.

La transparencia y la comunicación también juegan un papel importante en la «venta» y negociación del contrato. Por un lado, el cliente debe tener claro que las estimaciones son eso, estimaciones. Decir que una tarea puede tardar dos semanas no es una sentencia, es una posibilidad. Esto también ayuda a manejar las expectativas, un punto importante que debe dejarse claro en todos los departamentos, especialmente en ventas: se debe prometer lo que se puede cumplir. Los comerciales tienen que aprender a decir que no (Tortorella, 2012).

Por otro lado, el cliente debe saber qué se está haciendo. Para ello, Tortorella (2012) propone usar lo que se conoce como *agile engagement road map*: mostrar cómo se manejan las iteraciones y explicar al cliente en cuántas puede hacerse el proyecto. No obstante, antes de llegar a este punto y después de firmar el contrato, hay que tener en cuenta tres etapas previas:

- ***Check the fit:*** establecer relaciones de negocio.
- ***Project inception:*** explorar y evaluar.
- ***Iteration 0:*** planificar la infraestructura del proyecto.

Herramientas y estrategias de negociación

Las estrategias y herramientas que se explican en este apartado se han extraído del libro *Agile Contracts. Creating and Managing Successful Projects with Scrum* (2013), de Opelt, Gloger, Pfarl y Mittermayr, que desarrolla en detalle cómo negociar y licitar contratos ágiles de precio fijo. Es posible extraer de él algunos consejos generales para la negociación de contratos ágiles; en su mayoría, sugerencias para convencer a clientes no familiarizados con metodologías ágiles.

Estrategias para la negociación

El objetivo de las negociaciones es el mismo para ambas partes: negociar e implementar un proyecto con éxito. No obstante, cada implicado tiene intereses distintos que influyen en la negociación:

- **Objetivos del cliente:** comerciales y otros objetivos significativos, que dependerán del tipo de negocio y de contrato. Lo ideal es que estén definidos desde el principio y que coincidan con las personas encargadas de tomar decisiones, para poder seguir una línea clara de negociación. Una vez concretados, se debe decidir el método de implementación.
- **Objetivos del proveedor:** también comerciales y significativos. Un objetivo significativo puede ser incluir en el contrato ciertos principios básicos de implementación; uno comercial, cómo negociar el sistema de bonificaciones. Ambos varían en función del tipo de contrato, pero también de la situación del mercado: puede que un proveedor aún no tenga reconocimiento en la industria y decida ofrecer precios muy agresivos para ganar cuota de mercado.
- **Objetivos del equipo y de las personas implicadas:** relacionados con los pagos de bonificación de las personas involucradas en ambas partes. Se denominan incentivos financieros o no financieros, tangibles o intangibles. De nuevo, varían en función del tipo de contrato y de proyecto.

Por tanto, un primer paso es **definir los principios para el proyecto y la negociación desde el principio.** En caso de escoger un enfoque ágil, tanto cliente como proveedor deben acordar cómo van a conducir de forma cooperativa la negociación siguiendo dicho enfoque.

Como proveedores, hay que tener en cuenta que para el cliente es importante escoger al proveedor adecuado. El cliente se preguntará qué servicio y qué éxito espera de un proveedor; es decir, usará estrategias para saber qué proveedores pueden garantizar sus objetivos. Por eso es importante atender a dos aspectos: **el equipo de negociación y la agenda y documentación.**

Equipo de negociación

El equipo de negociación y todas las partes interesadas deben ser discretos y mantener la confidencialidad de los objetivos del proyecto, las estrategias de negociación y las

tácticas. Es habitual que los socios negociadores traten de obtener información, por ejemplo manteniendo conversaciones con empleados.

Opelt et al. (2013) proponen utilizar el modelo organizativo del FBI: la negociación se lleva a cabo en un equipo con tres roles estrictamente asignados.

1. Líder de la negociación

Es una sola persona. Se enfoca en la negociación, trata de identificar posibles malentendidos y busca optimizar todos los temas relevantes del proyecto. En una fase inicial, su función es probar las afirmaciones del socio negociador y asegurar la información. En principio, nunca decide en solitario sobre cuestiones esenciales: debe coordinar con su equipo todas las decisiones fuera de su ámbito de actuación, lo que permite disponer de más tiempo y evita decisiones precipitadas. Uno de sus mayores retos es la **tarea emocional:**

- Destacar las fortalezas del propio equipo técnico y poner en perspectiva las del proveedor, para conseguir que la otra parte acepte un modelo de contrato cooperativo.
- Provocar errores emocionales en los socios negociadores, sin dejarse provocar.
- Ser firme con sus demandas, pero hacer al proveedor concesiones importantes reales.
- No ofrecer compromisos, pero obtener propuestas de compromiso sobre cualquier tema.
- Ejercer una posición de poder convincente y no dejarse intimidar.

2. *Commander*

Sirve de apoyo al líder de la negociación. No interviene directamente, sino que sigue la negociación en silencio y le brinda orientación durante las pausas. No obstante, si esto no es suficiente, el commander puede llegar a intervenir. También sirve de enlace entre el *decision maker*, a quien mantiene informado, y el líder, que debe centrarse en gestionar la negociación.

3. Decision maker

Es quien toma las decisiones. Suele ser un cargo ejecutivo de la organización, que se mantiene al margen para no dejarse influir emocionalmente. Debe conocer los requisitos del proyecto y no tener preferencias personales por un proveedor u otro.

Agenda

Hay que asegurarse de que, tras finalizar el contrato, queden los menos temas abiertos posibles. El borrador inicial del contrato debe contener algunos elementos que puedan reconsiderarse en la negociación. En general, se debe aspirar a determinar la agenda —y, por lo tanto, el contenido de la negociación— uno mismo.

El objetivo es un contrato equilibrado que fomente la cooperación y, en consecuencia, asegure el éxito del proyecto. Con modelos ágiles, las negociaciones no deberían considerarse completas hasta que el proyecto se haya implementado con éxito: debido a la posibilidad de cambios, lo normal es que haya que revisar el contrato o los acuerdos durante el desarrollo. Por eso lo ideal es dejar claro un conjunto de reglas básico para cada proyecto, antes y después de las negociaciones de precios y contratos.

Herramientas de negociación

Las herramientas o estrategias para negociar un contrato ágil dependen de la empresa con la que se trabaje, el esfuerzo y el enfoque. Estas son algunas:

Sentando las bases

Estimular el interés antes de la negociación

La prioridad es que todas las partes comprendan las ventajas generales de los métodos ágiles y cómo un contrato ágil las respalda. Si una empresa ya trabaja con métodos ágiles, o al menos ha experimentado con ellos, puede aprovechar esa experiencia. No obstante, a menudo hay reticencias por parte de mandos intermedios (gestión de proyectos), que deben cambiar procedimientos y formas de trabajo practicadas durante años. Por ello, el proceso de transformación debe incluir reflexiones sobre lo que cada miembro de cada parte tiene que perder, y demostrar los beneficios individuales. Para convencer de la efectividad de este tipo de contratos pueden usarse los siguientes argumentos:

Se implementa lo que se necesita realmente, no solo lo especificado. Lo lógico es que el cliente exprese lo que quiere lograr y el proveedor lo haga posible. El cliente no debe centrarse en el «cómo», sino en el «qué»; así el proveedor tiene libertad para decidir cómo, y la oportunidad de implementar el «qué».

- **Estadísticas llamativas.** El alcance detallado de un proyecto de dos años cambia, por lo general, en más de la mitad. Lo lógico es ser flexible y probar una forma de contrato que minimice drásticamente el riesgo de inversiones estancadas.
- **Planes de bonificación.** Discutir los esquemas de incentivos actuales y su impacto real en el proyecto o el presupuesto es fundamental para tener un equipo motivado y eficiente.
- **Transparencia.** Con el modelo ágil se ve lo que se completa y lo que no, cada dos semanas. En el modelo tradicional los avances no se ven hasta el final; lo mismo ocurre con las pruebas, que no suelen realizarse hasta el último tercio del proyecto, cuando encontrar errores implica que la información llega demasiado tarde.

Intercambio de experiencias

Se trata de identificar problemas a partir de experiencias para probar algo nuevo. Un punto de partida ideal es cuando el cliente ya está familiarizado con los problemas de los contratos tradicionales. Según Opelt et al. (2013), la mayoría de los clientes traen consigo una «base de problemas».

Conviene aprovecharlo para mantener una conversación abierta, de la que surja el intercambio de experiencias; a partir de esos problemas y de los beneficios presentados, el propio cliente se dará cuenta de que el cambio a un enfoque ágil es ventajoso. Una opción en este punto es ofrecer un proyecto de prueba con la cooperación del cliente, para darle confianza y empezar a establecer expectativas correctas.

Establecer lenguaje y experiencias comunes

Una vez existe una conciencia común del problema, hay que trabajar en las bases de algo nuevo. El entendimiento mutuo requiere un lenguaje común y una visión compartida. Lo recomendable es organizar talleres de dos horas, por ejemplo. En los primeros 45 minutos se describe la idea básica de los métodos ágiles y se establecen expectativas realistas. Después, se trabaja en grupo con ejemplos de situaciones cotidianas para «practicar» el método ágil: por ejemplo, estimar los gastos de un viaje de tres semanas a las Bahamas mediante *Planning Poker*, y luego «implementarlo» con alguna otra actividad, como el *Ball-Point Game*.

Discusión de la forma

Feature shoot-out

Consiste en invitar a los clientes a un «concurso de características» donde se comparen los beneficios de los diversos tipos de contratos. Pueden usarse apoyos visuales (*flip charts* y colores) e imágenes para ilustrar los resultados. Sirve para convencer a un círculo más amplio de empleados del cliente y suavizar los rechazos más resistentes.

The black swan scenario

Este concepto apareció en 2011, en un artículo de Flyvbjerg y Budzier, en referencia a proyectos que, por su fracaso, devoraban grandes sumas de presupuesto y podían acabar con carreras y hasta destruir empresas (Flyvbjerg y Budzier, 2011). Se ha utilizado como argumento central para crear conciencia sobre la necesidad de cambios. Hay que presentarlo al cliente y acordar el análisis de riesgos de proyectos concretos; esto puede ayudarle a considerar los contratos ágiles, sobre todo si los errores encontrados tienen un gran impacto. Con el enfoque ágil se maximizan la transparencia y el valor adicional y, a la vez, se minimizan los riesgos.

Workshop on contract setup

Para mostrar la eficiencia de los contratos ágiles, puede realizarse un taller de un día con el cliente. Lo ideal es elaborar junto a él un ejemplo basado en un contrato ágil

(contrastado, si se quiere, con uno tradicional) y con un contexto adecuado. Se trata de poner en práctica la teoría explicada y de que el proveedor demuestre cómo trabaja y cómo se maneja ante los cambios.

Informes y métricas

Uno de los aspectos clave del modelo ágil es que el cliente siempre está informado sobre el estado del proyecto. No se trata de informes de estado, sino de una metodología de desarrollo que permite medir el progreso en pequeñas unidades binarias (terminadas o sin terminar). Sobre esta base se acuerdan en el contrato los informes que se presentan periódicamente.

KISS Backlog View

Se sigue el lema KISS (*keep it simple, stupid*: «hazlo simple»). Los elementos del backlog se asignan al sprint planificado y se estipula en el contrato una «tasa de conversión» de puntos de historia a gastos de equipo. Construyendo una tabla, el proveedor puede ver en cualquier momento qué funciones se han entregado y cuáles ha aceptado el cliente; gracias a los puntos de historia, se pueden prever los gastos. Si se quiere, pueden agregarse dos columnas que muestren qué partes se han descrito finalmente (en historias de usuario detalladas) y qué especificaciones han sido aprobadas por ambas partes. También debe mostrarse la velocidad del sprint y los días invertidos, las personas implicadas o los costes del equipo.

Team metric

Es una forma de medir cómo ha estado trabajando el equipo: muestra al cliente hasta qué punto ha trabajado en tareas relacionadas con el proyecto y en qué participó, por ejemplo, en mejoras del producto.

Focus: There Is a Single Goal!

En algunos proyectos se persigue toda una serie de objetivos, lo que provoca inevitablemente una disminución de la eficiencia y conflictos de intereses. Una forma de resolverlo es **fijar un objetivo común:** debe tener el mayor valor comercial y ser crucial para el éxito del proyecto. Esto no significa que ese único objetivo defina el contrato en su totalidad o en detalle; con un objetivo tan simple formulado, los distintos niveles en la dirección del proyecto pueden tomar decisiones.

Contratos ágiles en la era de la IA

La IA generativa hoy es una herramienta cotidiana: según la encuesta de Stack Overflow de 2025, el 84 % de las personas desarrolladoras usa o planea usar asistentes de IA, frente al 76 % del año anterior (Stack Overflow, 2025). Esta irrupción afecta a varios de los supuestos sobre los que se apoya un contrato ágil, y conviene abordarla con la misma actitud crítica que el resto del libro: sin exageraciones y con la evidencia disponible.

Lo que la evidencia dice (y lo que no)

Es tentador asumir que la IA multiplica la productividad y que, por tanto, habría que rehacer estimaciones y precios. Sin embargo, hay matices. Los experimentos controlados sobre tareas acotadas muestran aceleraciones notables —hasta un 56 % más rápido al completar una tarea concreta con GitHub Copilot (Peng et al., 2023), y en torno a un 20 % en un ensayo de Google (Paradkar et al., 2024)—. Pero esas ganancias individuales no se trasladan sin más a la velocidad de entrega del equipo: el tiempo que se ahorra escribiendo código reaparece en la revisión, las pruebas y la integración. El ensayo controlado más riguroso hasta la fecha encontró que personas desarrolladoras experimentadas, trabajando en bases de código maduras fueron de media un 19 % más lentas con IA, pese a estar convencidas de haber ido más rápido (Becker et al., 2025).

De aquí se extrae una lección directa para la contratación: **la velocidad del equipo sigue siendo algo que se mide, no que se presupone.** Un contrato ágil bien planteado ya resuelve esto (la velocidad se observa durante los primeros sprints y el precio se ajusta en consecuencia), y esa lógica no cambia con la IA. Lo que cambia es que las estimaciones iniciales son aún menos fiables al principio, porque el impacto real de las herramientas depende del tipo de tarea, la madurez del código y la experiencia del equipo. La recomendación de fijar un «punto de no retorno» tras el que el precio se consolida (→ «Precio fijo», capítulo 4) gana, si acaso, más sentido.

Estimación, puntos de historia y precio

Los modelos que fijan el precio «por punto de historia» o «por unidad de trabajo» parten de una relación estable entre esfuerzo y valor. Si la IA altera la productividad de forma desigual entre tareas, esa relación se vuelve más ruidosa. No invalida el modelo, pero aconseja:

- **Revisar la «tasa de conversión»** de puntos a coste con más frecuencia, sobre todo al inicio de la colaboración.
- **No trasladar al cliente un supuesto «dividendo de IA» no verificado.** Prometer entregas más rápidas «porque usamos IA» reproduce el error de sobreventa que este libro critica en los contratos de precio y alcance fijos.
- **Vigilar el desplazamiento del cuello de botella** hacia la revisión y la calidad, que puede encarecer iteraciones aparentemente más rápidas.

Propiedad intelectual del código generado por IA

Este es, probablemente, el punto que más afecta a la letra del contrato, y donde más importa distinguir jurisdicciones. La buena noticia es que, pese a la ausencia de una norma armonizada a escala mundial, los ordenamientos **convergen en un principio:** los derechos de autor exigen **autoría humana.** Lo generado *enteramente* por una IA, sin intervención creativa humana significativa, **no goza de protección**; lo asistido por IA, con aportación humana suficiente, sí puede protegerse. El Convenio de Berna es el trasfondo común de esta postura.

- **España.** El Texto Refundido de la Ley de Propiedad Intelectual (Real Decreto Legislativo 1/1996) reserva la condición de autor a la persona física (art. 5). El Registro ha denegado inscripciones de obras sin autoría humana identificable, como la novela *Iris,* presentada en Madrid en 2023 (DiG Abogados, 2025).
- **Unión Europea.** La Directiva 2001/29/CE y la jurisprudencia del TJUE exigen que la obra sea «creación intelectual propia del autor» (asunto *Infopaq*), lo que presupone un autor humano. El Parlamento Europeo reafirmó este principio en su informe sobre derechos de autor e IA generativa (Parlamento Europeo, 2026).
- **Estados Unidos.** La Oficina de Derechos de Autor (*Copyright and Artificial Intelligence*, Part 2, 2025) mantiene que los prompts, por sí solos, no bastan para atribuir autoría. Los tribunales lo confirmaron: en *Thaler v. Perlmutter* (D. C. Cir., 2025) se declaró que una obra generada de forma autónoma por IA no es registrable, y en marzo de 2026 el Tribunal Supremo declinó revisar el caso, con lo que la exigencia de autoría humana queda asentada.
- **Hispanoamérica.** El patrón se repite. En México, la Suprema Corte de Justicia de la Nación resolvió en 2025 que las obras generadas exclusivamente por IA no son registrables, por ser la autoría un derecho de las personas físicas (arts. 12 y 18 de la Ley Federal del Derecho de Autor), en línea con el Convenio de Berna y el T-MEC (SCJN, 2025). Argentina (Ley 11.723) y Colombia sostienen el mismo requisito de autoría humana, todavía sin normas específicas para la IA (Ciencia Latina, 2025).

Esta convergencia no es universal: algunos tribunales asiáticos (en China, por ejemplo) han protegido creaciones generadas con IA cuando mediaba intervención humana mediante *prompt engineering*. Y el marco evoluciona con rapidez, por lo que cualquier contrato con proyección internacional debería revisarse con asesoría legal local.

La consecuencia práctica es incómoda pero clara: **una cláusula de cesión clásica del tipo «todo el producto del trabajo pertenece al cliente» puede estar cediendo derechos sobre material que carece de protección legal.** A esto se suma el riesgo por el lado del entrenamiento: un asistente puede reproducir fragmentos de código con licencias incompatibles procedentes de sus datos de entrenamiento. Es un terreno ya litigado[4] y con consecuencias económicas reales: en 2025, Anthropic acordó pagar 1500

[4] En *Doe v. GitHub* (2022) se desestimó la mayoría de las reclamaciones, y solo siguen en pie, en apelación, las relativas a licencias de código abierto y a incumplimiento contractual.

millones de dólares por haber entrenado con libros pirateados, en un fallo que distinguió el entrenamiento con copias adquiridas legalmente (admisible) del realizado con material pirateado (no admisible). Conviene, además, leer la letra pequeña de los proveedores de IA: buena parte de sus condiciones de uso trasladan la responsabilidad al usuario, aunque algunos ofrecen ya compromisos de indemnización por reclamaciones de copyright.

Recomendaciones para el contrato:

- **Distinguir en las cláusulas entre entregables «generados por IA» y «asistidos por IA»**, y asignar la titularidad en consecuencia.
- **Exigir una cadena de custodia de la autoría humana** (revisión, modificación sustancial, documentación del proceso) que sostenga la protección de los entregables.
- **Cláusulas de datos y confidencialidad:** qué información del cliente puede introducirse en herramientas de terceros y bajo qué garantías.
- **Cláusula de saneamiento por licencias de terceros:** repartir la responsabilidad si el código incorpora fragmentos con licencias incompatibles.

Esta sección tiene carácter informativo y no constituye asesoramiento legal; dado que el marco evoluciona con rapidez, conviene contrastar cada contrato con asesoría jurídica en la jurisdicción aplicable.

Responsabilidad y calidad

La evidencia también advierte sobre la calidad: el código asistido por IA puede introducir defectos sutiles y problemas de seguridad que trasladan trabajo aguas abajo (Butler et al., 2025). Esto conecta con las cláusulas de **garantías y limitaciones de responsabilidad** ya tratadas: la definición de «hecho» debería incorporar de forma explícita la revisión de las contribuciones de IA, y las garantías, contemplar quién responde de un defecto originado en una sugerencia automática.

Cumplimiento normativo: el Reglamento de IA de la UE

Para clientes y proveedores europeos, la contratación no puede ignorar el **Reglamento de IA de la UE (Reglamento 2024/1689)**, en vigor desde el 1 de agosto de 2024 y de aplicación escalonada: las prácticas prohibidas y la alfabetización en IA se aplican desde el 2 de febrero de 2025; las obligaciones para modelos de IA de propósito general (GPAI) y la gobernanza, desde el 2 de agosto de 2025; y la mayoría de obligaciones para sistemas de **alto riesgo**, junto con el régimen sancionador, desde el 2 de agosto de 2026 (Comisión Europea, 2024). Las sanciones llegan hasta 35 M€ o el 7 % del volumen de negocio mundial en los casos más graves.

La implicación práctica es clara: si el producto contratado emplea IA en un ámbito de alto riesgo (empleo, crédito, educación, infraestructuras…) o integra un modelo GPAI,

las obligaciones de transparencia, documentación y cumplimiento en materia de copyright deben repartirse en el contrato.

Encaje con los principios ágiles

Nada de lo anterior contradice el *Manifiesto*; más bien lo tensa. La IA refuerza la entrega frecuente, pero pone a prueba el principio de que «el software funcionando es la medida principal de progreso» cuando la velocidad aparente esconde deuda técnica. Y devuelve al centro dos intangibles que este libro no deja de repetir: **la transparencia** (declarar cómo y dónde se usa la IA) y **la confianza** (asumir de forma compartida una incertidumbre que las herramientas no eliminan, solo desplazan). El *18th State of Agile Report* lo resume bien: la adopción de IA corre muy por delante de su gobernanza (Digital.ai, 2025). El contrato ágil es, precisamente, uno de los lugares donde esa gobernanza puede empezar a escribirse.

CONCLUSIÓN

En nuestro recorrido por el terreno del marco legal ágil hemos podido acercarnos un poco más al concepto de «contrato ágil». Tal y como ocurre con la gestión de proyectos, los contratos ágiles tratan de encajar en una realidad marcada por modelos tradicionales, transformando sus bases en principios ágiles y modificando los elementos del contrato, siempre en pos del cambio y no de su limitación. Podría decirse, por tanto, que son contratos híbridos y que responden a una necesidad de cambio.

Su elaboración depende del proyecto y de las circunstancias y el contexto en que dicho proyecto se haya concebido. Es decir, no existe un contrato ágil modelo que sea útil y eficaz para cualquier proyecto, porque está sujeto al tipo de cliente, la situación de la empresa, la relación entre cliente y proveedor... No obstante, puede encontrarse información sobre los diversos tipos de contratos que sirva de guía para diseñar uno propio.

En cuanto a la negociación de un contrato ágil, hemos aprendido que las estrategias y herramientas son parecidas a las de uno tradicional; la única diferencia es la naturaleza del contrato. No todo cliente va a sentirse cómodo trabajando bajo un enfoque ágil, por lo que, en la mayoría de las ocasiones, las negociaciones pasan por establecer unos objetivos claros entre las partes y convencer al cliente de los beneficios de trabajar con una metodología ágil.

En definitiva, los contratos ágiles pueden suponer un desafío, tanto en su planteamiento como en su elaboración y negociación. Su mayor desventaja es que todavía no cuentan con la confianza que puede suscitar un contrato tradicional.

A todo ello se añade, desde 2023, un factor que ninguna de las fuentes originales tuvo que considerar: la irrupción de la IA generativa. No cambia la naturaleza del contrato ágil, pero sí tensiona algunos de sus supuestos —la relación entre esfuerzo y valor, la titularidad de lo que se entrega, el reparto de responsabilidades— y añade obligaciones normativas nuevas. Lejos de restar vigencia a la filosofía ágil, refuerza sus dos pilares intangibles: la transparencia y la confianza. Un buen contrato ágil sigue siendo, ante todo, un acuerdo para cambiar bien.

APÉNDICES

Información, recursos y comunidad profesional

Club Agile →

Skill Arena →

Comunidad →

BoK/Wiki →

Observatorio →

Tiras de cómic →

Acerca de la certificación Scrum Manager® →

Preguntas frecuentes sobre cursos y exámenes Scrum Manager® →

Redes sociales y profesionales

LinkedIn

Youtube

Spotify

Ivoox

Mejora continua y control de calidad

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

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

Referencias bibliográficas

Libros y artículos académicos

Arbogast, T., Larman, C. y Vodde, B. (2010). *Practices for Scaling Lean & Agile Development: Large, Multisite, & Offshore Product Development with Large-Scale Scrum*. Addison-Wesley.

Becker, J. et al. (2025). *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*. METR.

Butler, S. et al. (2025). *Echoes of AI: Investigating the Downstream Effects of AI Assistants on Software Maintainability*. arXiv:2507.00788.

Eveleens, J. L. y Verhoef, C. (2010). «The Rise and Fall of the Chaos Report Figures». *IEEE Software*, 27(1), 30–36.

Gerster, D. y Dremel, C. (2019). «Agile Contracts: Learning from an autonomous driving sourcing project». *Proceedings of the 27th European Conference on Information Systems (ECIS)*, Estocolmo y Uppsala.

Lewicki, R. J., Saunders, D. M. y Barry, B. (2009). *Essentials of Negotiation*. McGraw-Hill Higher Education.

Opelt, A., Gloger, B., Pfarl, W. y Mittermayr, R. (2013). *Agile Contracts. Creating and Managing Successful Projects with Scrum*. Wiley.

Peng, S., Kalliamvakou, E., Cihon, P. y Demirer, M. (2023). *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot*. arXiv:2302.06590.

Pichler, R. (2012). *Agile Product Management with Scrum*. Addison-Wesley.

Pries-Heje, L. y Pries-Heje, J. (2014). «Agile Contracts: Designing an Agile Team Selection Guideline». *IRIS*, n.º 5.

Takeuchi, H. y Nonaka, I. (1986). «The New New Product Development Game». *Harvard Business Review*, 64(1), 137–146.

Informes y estudios

BearingPoint (2020). *Agile Pulse 2020*.

BearingPoint (2022). *Agile Pulse 2022. Doing Agile vs. Being Agile*.

Digital.ai (2025). *18th Annual State of Agile Report*.

Fritz, W. y Fischer, T. (2021). *The Art of Agile Contracting*. BearingPoint.

Stack Overflow (2025). *Developer Survey 2025*.

U.S. Copyright Office (2025). *Copyright and Artificial Intelligence, Part 2: Copyrightability*.

Artículos y recursos web

Alfonso, A. (2018). «Make Agile Contracts Work For Your Teams & Clients». *The Digital Project Manager*.

Bañeres, J. P. (2012). «Ejemplo de plantilla para documentar el uso de scrum en un proyecto». *Navegapolis*.

Barato, J. (2017). «Contratos ágiles». *PMPeople*.

Hernández, J. (2019). «Contratos ágiles — Primera, segunda y tercera parte». *Agile Experience*.

Kelly, A. (2011). «Agile Contracts». *InfoQ*.

Léonard, N. (2019). «¿Podría un Product Owner ser un consultor externo?». *LinkedIn*.

Martin, R. C. (s. f.). *Fixed Price / Fixed Scope Contracts*. (Modelo «Bob Martin».)

Stevens, P. (2019). «10 Agile Contracts». *Agile Software Development*.

Sutherland, J. (s. f.). *Agile Contracts*. Scrum Inc.

VersionOne / Digital.ai (2012). «Must-Haves for Agile Contracts».

Marco normativo y jurisprudencia

Comisión Europea (2024). *Reglamento (UE) 2024/1689, por el que se establecen normas armonizadas en materia de inteligencia artificial (Reglamento de Inteligencia Artificial)*. Diario Oficial de la Unión Europea.

Directiva 2001/29/CE del Parlamento Europeo y del Consejo, relativa a la armonización de determinados aspectos de los derechos de autor y derechos afines en la sociedad de la información.

España. *Real Decreto Legislativo 1/1996, Texto Refundido de la Ley de Propiedad Intelectual (TRLPI)*.

México. *Ley Federal del Derecho de Autor* (arts. 12 y 18); Suprema Corte de Justicia de la Nación (SCJN, 2025), resolución sobre obras generadas por IA.

Argentina. *Ley 11.723 de Propiedad Intelectual*.

Parlamento Europeo (2026). *Informe sobre los derechos de autor y la inteligencia artificial generativa: oportunidades y desafíos* (A10-0019/2026).

Thaler v. Perlmutter, 130 F.4th 1039 (D. C. Cir. 2025); certiorari denegado por el Tribunal Supremo de EE. UU. (marzo de 2026).

Doe v. GitHub, Inc., n.º 3:22-cv-06823 (N. D. Cal. 2022) —mayoría de reclamaciones desestimadas; resto en apelación ante el Noveno Circuito—.

Fuentes doctrinales de apoyo (§ propiedad intelectual): DiG Abogados (2025), «Derechos de autor en obras generadas por inteligencia artificial»; Ciencia Latina (2025), «Inteligencia artificial, propiedad intelectual y derechos de autor en América Latina».

Vídeos

Abad Londoño, J. H. (2017). *Hablemos de contratos ágiles* [vídeo]. YouTube.

Ágiles – Liderazgo y cultura (2022). *Contratos ágiles* [vídeo]. YouTube.

Chabes Floreano, J. (2020). *Los contratos en el mundo AGILE* [vídeo]. YouTube.

Schwartz, F. (2017). *Contratos ágiles* [lista de reproducción]. YouTube.

Tortorella, P. (2012). *Kleer — Contratos ágiles en la vida diaria* [vídeo]. YouTube.