Contratación de metodologías ágiles para el desarrollo de software

Elías Mohor Fuster, abogado asociado, Carey

Artículo publicado en Hipervínculos, Blog de Tecnologías y Privacidad de Carey, en septiembre de 2019

Hace aproximadamente un año y en conjunto con nuestro cliente Inria Chile, fundación dedicada a la investigación, desarrollo y transferencia de tecnologías en el área de las ciencias y tecnologías de la información y comunicación, iniciamos el desafío de esquematizar un contrato de desarrollo de software que abordara las particularidades de las "metodologías ágiles".

La creación de software utiliza metodologías que exigen poder reescribir una y otra vez las tareas y objetivos inicialmente acordados, siendo esenciales para este enfoque de trabajo la confianza y la colaboración con el cliente. Regular hitos, objetivos, tareas y plazos –como lo hace un contrato tradicional de desarrollo de software– no resulta útil ni conveniente, porque todos esos ítems se van modificando rápidamente con el avance del proyecto, con lo que pronto el contrato se vuelve un papel obsoleto, que no es más que un dolor de cabeza para los desarrolladores y el cliente.

En este artículo expondremos el proceso que emprendimos con nuestro cliente Inria Chile para estructurar un contrato de desarrollo bajo metodologías ágiles, y en particular, la experiencia y conclusiones a las que llegamos tras utilizar las metodologías ágiles en el ejercicio mismo de diseño del contrato.

Comenzaremos por explicar, de forma muy breve, qué son las metodologías ágiles para el desarrollo de software y por qué han resultado más convenientes y son preferidas por los desarrolladores actualmente.

Breve reseña de las metodologías empleadas para el desarrollo de software

En sus albores, el desarrollo de software se realizaba de una forma más bien improvisada: Los desarrolladores tenían una idea del software que querían crear y comenzaban a escribir código para llegar a esa idea. Consecuentemente, los primeros contratos de desarrollo se limitaban a fijar un conjunto de características y funcionalidades deseadas y le daban un plazo al equipo de desarrollo para alcanzarlas.

Bajo una mirada retrospectiva, se le ha llamado a este antiguo esquema –con un tono irónico quizás– el modelo de desarrollo "big bang", porque el contrato y la metodología (o más bien la ausencia de ella) se limitan a enumerar un conjunto de funcionalidades que se espera aparezcan "de la nada" en una fecha determinada. Este esquema tardó muy poco en dejar al descubierto sus graves falencias: Los desarrolladores se ponían a trabajar con un listado de funcionalidades deseadas, que muchas veces resultaban ambiguas y que por tanto había que interpretar. El proceso tampoco contaba con instancias de retroalimentación del cliente, ni menos de los usuarios finales del producto. Por todos estos factores, el resultado

usualmente era el mismo: el día del estreno del producto los clientes generalmente quedaban insatisfechos con el resultado, dando lugar a conflictos entre las partes involucradas.

La industria del software se dio cuenta rápidamente de que era necesario implementar metodologías de planificación que permitieran hacer frente a estos problemas, y tomó prestadas las de otras ramas de la ingeniería, como la construcción, por ejemplo, articulando lo que se conoció como el modelo de desarrollo de software de "cascadas", bajo el cual se dividen las diferentes etapas del proyecto y se van ejecutando una a una íntegramente. Por ejemplo: (i) identificación de los requerimientos y funcionalidades; (ii) diseño de la solución (iii) implementación (escritura del código); (iv) pruebas del software y (v) mantenimiento. Una vez que se termina una etapa se pasa a la siguiente sin mirar atrás.

Esta metodología predominó en la industria y en la academia durante los años ochenta y noventa, principalmente sostenida por la creencia de que una buena planificación en el inicio del proyecto haría más eficiente y exitoso el desarrollo.

Sin embargo, ya en los años ochenta se comenzaban a oír importantes críticas a este modelo. Algunos desarrolladores observaron que, durante el proceso, el equipo de desarrollo va aprendiendo constantemente, recibe retroalimentación de quien le encargó el desarrollo o de los futuros usuarios del software, corrige lineamientos que no estuvieron bien enfocados en el inicio e incluso descubre la necesidad de incorporar funcionalidades no contempladas al momento de la planificación y eliminar algunas que sí fueron consideradas en el comienzo.

Ninguno de estos elementos es recogido por el modelo de cascadas, que avanza etapa tras etapa de forma inexorable y sin mirar atrás, porque así mandan la planificación y las funcionalidades acordadas por las partes en el momento que se encargó el desarrollo. Volver una y otra vez a la etapa de "identificación de requerimientos" sería un contrasentido para el espíritu del modelo de cascadas pues su finalidad es, precisamente, ordenar cómo se avanza en el proyecto estableciendo etapas bien definidas.

Esta importante falencia del modelo comenzó a notarse en el producto resultante: El software terminado no recogía los requerimientos o funcionalidades que pudieron haber sido "descubiertos" en las últimas etapas del proyecto, y además, tenía usualmente un importante desfase con la tecnología de la época y las necesidades del negocio, pues había sido diseñado a partir de las ideas y la realidad de la época en que se realizaron las primeras etapas de la planificación de cascada, que en algunas ocasiones podía llegar a tener años de distancia con la fecha de lanzamiento del producto.

Con el correr de los años, los desarrolladores se fueron dando cuenta, entre otras cosas, de que: (i) es imposible reunir todos los requerimientos al principio de un proyecto, y que estos deben ser descubiertos en la marcha; (ii) los requisitos deben ser plasmados en un listado flexible, que admita modificaciones futuras, pues con seguridad irán cambiado con el avance del proyecto; y (iii) la metodología debe recoger el hecho de que siempre habrá más cosas por hacer que tiempo y dinero, por lo que debe incorporar mecanismos para ir cambiando las prioridades del desarrollo e ir usando eficientemente los recursos disponibles.

Para hacer frente a esta realidad, durante el año 2001 un grupo de importantes desarrolladores de software se reunió a diseñar un nuevo modelo de trabajo. El grupo, que se autodenominó la "Alianza Ágil" (*The Agile Alliance*) sentó las bases de este nuevo modelo en un breve documento que denominó el "Manifiesto ágil" y en una precisa enunciación de doce principios respecto a cómo debería estructurarse el desarrollo de software. En

Carey y Cía. Ltda.
Isidora Goyenechea 2800, Piso 43
Las Condes / Santiago / Chile
Tel : +56 2 2928 2200 / Fax : +56 2 2928 2228
www.carey.cl

aquellos documentos, la Alianza recalcó la importancia de privilegiar conceptos como la colaboración con el cliente y la respuesta al cambio por sobre otros como la negociación contractual, la documentación técnica o el seguimiento estricto de planes. Desde su publicación a esta fecha, el manifiesto ha sido firmado por un creciente número de desarrolladores y sus principios han ido siendo adoptados con cada vez más fuerza en la industria del software.

Desde la publicación del manifiesto ágil a esta fecha han emergido numerosas metodologías de desarrollo, que pretenden recoger los principios antes enumerados (p. ej. metodología *Scrum*, programación extrema, desarrollo de software *lean*, entre otros). Aunque existen diferencias importantes entre unas y otras, todas comparten el principio de *"responder al cambio por sobre seguir un plan"*. Recoger este importante principio en los contratos que regulen el desarrollo de software bajo estas metodologías representa un desafío significativo.

A continuación abordaremos las dificultades que enfrentamos al elaborar un contrato que recogiera estos principios de agilidad y expondremos algunos de nuestros hallazgos tras este ejercicio.

Como redactar un contrato para las metodologías ágiles

Como se puede prever, un contrato de desarrollo "tradicional" –entendiendo por tal uno que regula hitos, objetivos, tareas y plazos– probablemente no resulte útil ni conveniente para regular el desarrollo de software bajo metodologías ágiles, que exigen poder modificar el proyecto muchas veces durante su transcurso.

La primera deficiencia que reporta la estructura tradicional de un contrato de desarrollo es que los hitos, objetivos y plazos pactados inicialmente van quedando **obsoletos** al poco tiempo bajo las metodologías ágiles, pues de acuerdo con ellas, tanto el cliente como el desarrollador van descubriendo la necesidad de hacerle ajustes o derechamente reemplazarlos por otros.

La obsolescencia de los términos contractuales pone a las partes frente a un **dilema**: o se mantienen apegadas al contrato, y continúan desarrollando un proyecto que ya saben que está mal enfocado o que derechamente ya no sirve; o acuerdan una modificación al contrato, ya sea a través del procedimiento de control de cambios o de una modificación contractual propiamente tal; o derechamente deciden alejarse de las disposiciones contractuales, con el creciente riesgo de responsabilidad contractual que ello conlleva para cada una.

Nos podemos dar cuenta, entonces, de que hacer un contrato que sirva para las metodologías ágiles significa **equilibrar agilidad y flexibilidad, por una parte, con certeza jurídica por la otra**, pero ¿cómo equilibrar estos dos conceptos tan distintos?

Alternativa 1: Modificación del contrato

Una primera alternativa –quizás la más obvia– sea incluir **una cláusula que permita a las partes modificar el proyecto completo si es necesario**. Sin embargo, no es difícil darse cuenta de que esta alternativa en realidad no representa ninguna solución. Las partes

Carey y Cía. Ltda.
Isidora Goyenechea 2800, Piso 43
Las Condes / Santiago / Chile
Tel : +56 2 2928 2200 / Fax : +56 2 2928 2228
www.carey.cl

siempre tienen la facultad de modificar el contrato, aunque éste no lo diga. Por lo tanto, incluir una cláusula de esta naturaleza no es realmente un aporte, ni representa innovación alguna.

Pero, además, toda modificación de contrato depende, lógicamente, de que ambas partes estén de acuerdo en modificarlo; y creemos que hay muy buenas razones para pensar que ese acuerdo no siempre se alcanzará.

Para entender este punto pensemos, por ejemplo, en un proyecto de desarrollo tradicional, regulado por un contrato de desarrollo tradicional, que dure 10 meses. Imaginemos que en el **cuarto mes**, el cliente descubre –gracias a los avances realizados– que **el proyecto completo está muy mal enfocado**, y se debate entre continuar con el proyecto, o pedir las modificaciones necesarias para reencauzarlo a sus necesidades. Las opciones que tendrá en un contrato tradicional son:

- *terminar inmediatamente el contrato, de común acuerdo con el desarrollador,* alternativa que se ve poco viable, pues el desarrollador no tiene incentivos para aceptar la propuesta del cliente y renunciar a los seis meses de trabajo y pago restantes que le garantiza su contrato vigente; o

- *gatillar la cláusula de control de cambios del contrato* –si la hay– lo que implica detener el proyecto para negociar si los cambios solicitados están o no dentro del ámbito inicial, y de no estarlo, negociar su objeto y precio, y cómo impacta este cambio en la planificación final; o

- *modificar un contrato sin cláusulas de control de cambios*, lo que conlleva los mismos inconvenientes anteriores, pero sin un protocolo y tiempos preestablecidos para ello.

Alternativa 2: Flexibilidad total y radical del contrato

En el otro extremo, una solución podría ser pactar contractualmente una **flexibilidad total y radical** en la determinación del objeto del contrato, los plazos y precio asociado; y facultar a los representantes técnicos de cada parte para que modifiquen el proyecto a su discreción, cuando lo estimen necesario. Un contrato bajo este esquema podría, incluso, no tener requerimientos técnicos de ningún tipo, de modo que sean los propios representantes técnicos de cada parte quienes los vayan descubriendo con el correr del proyecto.

Sin embargo, tampoco es difícil darse cuenta de que esta alternativa no es una buena solución. Un contrato de esta naturaleza **genera incertidumbre, pues no define nada**, lo que atenta contra uno de los propósitos esenciales de todo contrato: Generar algún nivel de certeza jurídica.

Tampoco soluciona el problema de los acuerdos que tenía la alternativa anterior, sino que solo lo traspasa a los representantes técnicos, quienes también podrían entramparse en discusiones sin salida respecto al rumbo del proyecto, por tener opiniones divergentes. Finalmente, nos daremos cuenta de que un contrato como éste se asemeja mucho a trabajar **sin un contrato** –al menos en lo que respecta a las características del proyecto y su metodología de avance– lo que nos demuestra que esta alternativa no cumple su propósito, y que el contrato sigue siendo inútil.

Carey y Cía. Ltda.
Isidora Goyenechea 2800, Piso 43
Las Condes / Santiago / Chile
Tel : +56 2 2928 2200 / Fax : +56 2 2928 2228
www.carey.cl

Alternativa 3: Intercalar períodos de trabajo con períodos de reevaluación del proyecto

Es entonces que empieza a emerger una tercera alternativa, intermedia entre la rigidez y la flexibilidad total. La alternativa consiste en ir **intercalando períodos de trabajos con períodos de reevaluación del proyecto**, es decir, alternar momentos de avance en torno a objetivos fijos, con momentos de análisis y planificación de los siguientes tramos de avance.

La idea sería **fragmentar el proyecto**, en varios tramos pequeños, e ir introduciendo entre tramo y tramo un período de reevaluación que permita a las partes rediseñar el proyecto de ser necesario, o ir especificando sus objetivos, en el evento de que estos no hubieran sido acordados desde el inicio.

Al dar con esta alternativa nos dimos cuenta de que, para ser útil, el contrato debía centrarse en **regular la metodología de trabajo** antes que los requerimientos técnicos, pues lo importante era lo primero y no lo segundo. Así fue que comenzamos a trabajar en regular contractualmente una metodología de ciclos de trabajo intercalados con etapas de revisión y planificación de los ciclos posteriores.

La mecánica de trabajo es bastante simple y parece lógica. Cada ciclo constituye un período de trabajo con objetivos bien definidos. Durante su transcurso, el desarrollador puede trabajar tranquilo en las tareas previamente acordadas y no tiene la obligación de irlas reevaluando. Tampoco es el minuto para que el cliente intervenga y corrija el avance, pues esto sólo ocurrirá una vez concluido el ciclo, en una etapa especialmente destinada para tal efecto. Las partes deben tomar registro de las conclusiones a las que lleguen durante esta etapa, pues ellas guiarán el trabajo del desarrollador durante el ciclo siguiente.

¿Y quiénes son las personas facultadas para adoptar estos acuerdos de modificación del proyecto? A nuestro juicio el contrato debería delegar esta potestad en los representantes técnicos de cada parte, pues la naturaleza de estos acuerdos exige que quienes los adopten sean personas con un buen entendimiento de los aspectos técnicos del proyecto. No obstante, también deben ser personas con habilidades de negociación y toma de decisiones; en la jerga de algunas metodologías ágiles (como la metodología Scrum, por ejemplo) al representante del cliente se le suele llamar *"product owner"* (dueño del producto), porque es quien, en último término, lleva las riendas del proyecto.

Como ya hemos anticipado, otro elemento fundamental es que los acuerdos de estas dos personas queden documentados en una bitácora o registro, que nosotros hemos denominado "Registro de Proyecto" en nuestra propuesta contractual, y que será el resguardo de ambas partes ante las decisiones adoptadas durante el avance del proyecto.

Hasta aquí la mecánica de trabajo parece bastante simple, intuitiva y lógica. Sin embargo, al poco andar nos dimos cuenta de que la metodología exigía aún más **ajustes contractuales** para poder funcionar, pues esta idea de los ciclos de trabajo, por sí sola, presenta algunos problemas importantes que el contrato debe atender.

Los siguientes son los ajustes contractuales que, por el momento, hemos podido identificar:

1) ***Las etapas de análisis y planificación deben tener duración indefinida.*** Fijarles una duración determinada resulta finalmente contraproducente para las dos partes. Eso les

Carey y Cía. Ltda.
Isidora Goyenechea 2800, Piso 43
Las Condes / Santiago / Chile
Tel : +56 2 2928 2200 / Fax : +56 2 2928 2228
www.carey.cl

haría adoptar acuerdos bajo la presión de que, si no se apuran, el proyecto continuará de acuerdo con las directrices previamente adoptadas, que bien podrían haber quedado obsoletas a la luz de los últimos avances. Esto no le conviene a nadie, pero en particular no le conviene al cliente, que debe tener la libertad y tranquilidad de corregir y reestructurar su proyecto con calma.

2) ***Los ciclos deben tener duraciones genéricas y no estar asociadas a fechas específicas***. Este segundo ajuste es una consecuencia del primero, y consiste en que la **duración** de los ciclos se debe establecer usando plazos **genéricos** (por ejemplo "dos semanas") y no usando fechas específicas ("hasta el 25 de agosto"). ¿Por qué? Pues porque si ya hemos resuelto que la etapa de revisión y planificación que hay entre ciclo y ciclo tenga una duración indefinida, pues entonces su prolongación iría acortando todos los plazos futuros que se hubieran fijado con fechas específicas. Si el propósito de esta etapa, como ya hemos dicho, es analizar y planificar con total libertad y tranquilidad, los plazos de los ciclos futuros deben **suspenderse** mientras dure esta etapa. Creemos que esto se logra con un ajuste muy sencillo, que es el uso de plazos genéricos, que se inicien o "activen" sólo una vez que las partes hayan resuelto iniciarlo, por haber concluido la revisión del ciclo anterior y la planificación del ciclo siguiente. Mientras eso no ocurra, ningún plazo comienza a correr.

3) ***El proyecto entero debe ser modificable en cualquier etapa de análisis y planificación.*** El tercer ajuste consiste en resguardar que las etapas de análisis y planificación no se vean influenciadas por los acuerdos que anteriormente se hubieran adoptado respecto de ciclos futuro (sus objetivos o duración, por ejemplo). Como ya hemos señalado, la idea fundamental de las etapas de revisión y planificación es que las partes puedan rediseñar libremente el proyecto si es necesario y ello puede requerir disminuir o aumentar la cantidad de ciclos futuros, su duración y objetivos. Por ello, todas las planificaciones de ciclos futuros deben ser inherentemente provisionales; deben ser modificables hasta el último momento antes de su inicio.

4) ***Las partes deben poder terminar unilateralmente el contrato, sin expresión de causa.*** Este cuarto ajuste es una consecuencia de todos los ajustes anteriores. Si durante cada etapa de revisión y planificación las partes tienen total libertad para modificar el proyecto, sin que corra en su contra ningún plazo para ponerse de acuerdo, pues entonces resulta posible que **nunca se pongan de acuerdo**. Al menos es una posibilidad no menor. Las partes podrían discutir y discutir sin alcanzar un punto de encuentro, pues la metodología no contempla un mecanismo para suplir su falta de acuerdo, y tampoco es bueno que lo contemple, como ya hemos explicado. A nadie se le puede obligar a llegar a un acuerdo, si no, no es un verdadero acuerdo. Aquí, las partes han llegado a un punto muerto.

Pero ¿tiene sentido mantener vivo un contrato que ha llegado a un punto muerto? Creemos que no y eso es lo que justifica nuestro cuarto ajuste. Las partes deben poder terminar unilateralmente el contrato, sin expresión de causa, durante cada etapa de revisión y planificación.

Permitirle esto a las partes cumple dos funciones. Una es **equiparar** el poder negociador de cada una durante la etapa de revisión y planificación. Si ambas pueden "salir de la mesa de negociación", es esperable que su discusión sea más libre y sincera. La segunda función es otorgarle a las partes una **garantía**, que les permita **liberarse** del contrato si la otra parte no está debidamente comprometida con la metodología.

Carey y Cía. Ltda.
Isidora Goyenechea 2800, Piso 43
Las Condes / Santiago / Chile
Tel : +56 2 2928 2200 / Fax : +56 2 2928 2228
www.carey.cl

5) ***Los precios deben pactarse por ciclo trabajado y no por el proyecto completo.*** Este ajuste es una consecuencia del cuarto, y es que el precio por los servicios debe pactarse por ciclo trabajado y no por todo el proyecto. Si el proyecto puede terminar en cualquier momento, debe idearse un mecanismo para que el precio no dependa de que el proyecto llegue a término. Creemos que eso se logra estableciendo que los pagos se hagan en función de cada ciclo terminado.

Además, el precio de cada ciclo debe ser **negociable y ajustable** durante la etapa de revisión y planificación anterior a su inicio. Si lo son sus objetivos y duración, pues también lo debe ser su precio, para mantener la equidad de las negociaciones.

6) ***Regular qué ocurre con la terminación anticipada.*** El sexto ajuste consiste en regular qué ocurre si alguna de las partes ejerce su derecho a la terminación anticipada. Esto resulta conveniente para que las partes puedan terminar el contrato en buenos términos, pero también es una exigencia del propio espíritu de la metodología ágil, que siempre busca que el cliente pueda marcharse con algo que le resulte útil y que el esfuerzo invertido no se pierda. En nuestra propuesta contractual establecimos que la terminación anticipada dé lugar a un ciclo de cierre de proyecto –sin costo para el cliente– donde las partes negocien un último entregable. Eso sí, delimitamos en el contrato lo que el cliente puede exigir de ese entregable final. El desarrollador sólo está obligado a incorporar en ese entregable tareas y objetivos que hubieran formado parte de los ciclos efectivamente concluidos. **No se le puede exigir incorporar tareas y objetivos nuevos**. Ahora bien, si el desarrollador voluntariamente quiere incorporarlos, bien puede hacerlo, pero no se lo puede obligar a ello.

7) ***La planificación de un nuevo ciclo implica la aceptación del trabajo realizado en el ciclo anterior.*** Este último ajuste es **esencial** para que toda la metodología sea viable y consiste en establecer contractualmente que **la planificación de un nuevo ciclo implique la aceptación del trabajo realizado en el ciclo anterior, con independencia de las observaciones que le haya formulado el cliente.** Como ya hemos explicado, el principal propósito de las metodologías ágiles es darle al cliente la oportunidad de revisar y corregir periódicamente el trabajo del desarrollador, quien está constantemente abriéndole las puertas, una y otra vez, para que observe y corrija lo que hace el equipo de desarrollo. En este entendido, lo natural y esperable es que cada revisión deje observaciones e instrucciones de mejora al desarrollo logrado. Ese es el principal objetivo de la metodología.

Por tanto, lo natural e inherente a la metodología será encontrarnos con un registro de proyecto lleno de observaciones, comentarios e instrucciones de mejora. Si las hay, es porque justamente la metodología está funcionando.

Pero imaginemos qué pasaría si una vez terminado el contrato, el cliente tomara el registro de proyecto, con todas sus observaciones, comentarios e instrucciones de mejora y lo llevara a un tribunal, y lo usara como prueba de un incumplimiento contractual para exigir la restitución de todo lo que ha pagado durante el proyecto. Ese escenario sería, sin duda, un abuso de la metodología y un uso –de muy mala fe– de las herramientas que esta concede.

Para resguardar que eso no ocurra, el contrato debe incorporar un mecanismo que haga que el avance signifique la validación de los ciclos terminados. Creemos que esto se logra estableciendo que toda planificación de un nuevo ciclo implique la aceptación del trabajo realizado en el ciclo anterior, con independencia de las observaciones que le haya formulado el cliente. Esto, bajo la lógica de que, **si las partes han podido ponerse de acuerdo sobre un nuevo ciclo es porque quieren seguir trabajando juntas**. Más todavía si tenían la

Carey y Cía. Ltda.
Isidora Goyenechea 2800, Piso 43
Las Condes / Santiago / Chile
Tel : +56 2 2928 2200 / Fax : +56 2 2928 2228
www.carey.cl

facultad de terminar unilateralmente el contrato. Por lo tanto, lo razonable es concluir que valoran lo avanzado, aún a pesar de las observaciones y correcciones que pudieran haber surgido durante cada etapa de revisión y planificación.

Si el cliente llegara a rechazar completamente lo realizado en un ciclo y no quiere que opere la validación y aceptación, pues sencillamente no debe seguir embarcado en el proyecto y debe hacer uso de su facultad de terminación unilateral que, como ya hemos explicado, no lo deja a la deriva, sino que le da derecho a un último ciclo de cierre de proyecto, sin costo para él.

Estos son los ajustes que hemos podido identificar por el momento y quizás haya más que por ahora no podemos ver. Pero con los que ya tenemos, podemos sintetizar la metodología contractual de la siguiente manera: El proyecto se divide en ciclos de trabajo en los que el desarrollador debe trabajar para objetivos y tareas previamente acordados. Entre un ciclo y otro hay una etapa de duración indefinida en la que las partes revisan el trabajo del ciclo anterior y planifican el del ciclo siguiente, pudiendo modificar sus objetivos, tareas, duración, precio y entregables esperados; e incluso pudiendo eliminar ciclos futuros y terminar unilateralmente el contrato. Si aun teniendo esta última facultad las partes logran acuerdo sobre cómo seguir y planifican un nuevo ciclo, pues ella ha de implicar la validación y aceptación del trabajo realizado en el ciclo anterior, para resguardar la viabilidad de la metodología. Todos los acuerdos alcanzados deben ser documentados en una bitácora o registro de proyecto firmado por los representantes técnicos de ambas partes.

Cuando ya habíamos redactado el contrato, nos dimos cuenta con agrado y sorpresa de dos cosas: que podía funcionar perfectamente sin requerimientos técnicos iniciales –es decir como un proyecto en blanco– y no sólo eso, sino que también se puede emplear para otros proyectos distintos del desarrollo de software, pues el trabajo por ciclos y la mecánica de negociación periódica son universales.

Los contratos reflejan las esperanzas de las partes, pero en mucho mayor medida, también los miedos. Los proyectos exitosos finalmente no surgen de un contrato bien dibujado, sino de relaciones basadas en la colaboración, transparencia y confianza. El contrato ágil debe, por lo tanto, entregar la flexibilidad necesaria para que esos niveles de confianza puedan prosperar.

***

Modelo de contrato ágil **Cláusula de agilidad**

Los contenidos y materiales de este artículo no constituyen asesoría legal. Este artículo sólo tiene fines informativos, es de carácter general y no pretende ser exacto ni completo.

Carey y Cía. Ltda.
Isidora Goyenechea 2800, Piso 43
Las Condes / Santiago / Chile
Tel : +56 2 2928 2200 / Fax : +56 2 2928 2228
www.carey.cl